Metadata Extraction from Conference Proceedings Using Template-Based Approach

Liubov Kovriguina[1(✉)], Alexander Shipilo[3], Fedor Kozlov[1], Maxim Kolchin[1], and Eugene Cherny[1,2]

[1] ITMO University, Saint-petersburg, Russia
{lkovriguina,alexandershipilo,kozlovfedor}@gmail.com,
{kolchinmax,eugene.cherny}@niuitmo.ru
[2] Åbo Akademi University, Turku, Finland
[3] Saint-Petersburg-State University, Saint-Petersburg, Russia

Abstract. The paper describes a number of metadata extraction procedures based on rule-based approach and pattern matching from CEUR Workshop proceedings Cf. http://ceur-ws.org and its converting to a Linked Open Data (LOD) dataset in the framework of ESWC 2015 Semantic Publishing Challenge Cf. http://github.com/ceurws/lod/wiki/SemPub2015.

Keywords: Metadata extraction · Semantic publishing · Linked open data · Semantic web · PDF parsing · Natural language processing

1 Introduction

The work that is presented in this paper aims to provide a solution for Task 2 of ESWC 2015 Semantic Publishing Challenge (see footnote 1). The task is to crawl and parse PDF papers from CEUR Workhop proceedings web site[3] and create a LOD dataset containing detailed information about the papers, citations, authors and their organizations and etc.

The source code and instructions to run the crawler are available at our GitHub repository[1].

The main goal of the paper is to provide an approach for information extraction from the textual content of the papers in PDF format and translating it to LOD format. This information should provide a deeper understanding of the context in which the paper was written. In particular, extracted information is expected to answer queries about authors' affiliations and research institutions, research grants, funding bodies, and related works. Previous work includes results presented in the paper [2].

Tasks of the paper include

– analysis of workshop paper elements and ontology development using published and frequently used vocabularies;

[1] Cf. http://github.com/ailabitmo/ceur-ws-lod.

F. Gandon et al. (Eds.): SemWebEval 2015, CCIS 548, pp. 153–164, 2015.
DOI: 10.1007/978-3-319-25518-7_13

– development of paper metadata extraction procedures from PDF files;
– development of the tool crawling PDF papers, applying metadata extraction procedures and publishing results as Linked Open Data;
– testing the developed tool using testing module and the set of SPARQL queries.

The output dataset should allow to perform the following queries.

– Q2.1 (Affiliations in a paper): Identify the affiliations of the authors of the paper X.
– Q2.2 (Papers from a country): Identify the papers presented at the workshop X and written by researchers affiliated to an organization located in the country Y.
– Q2.3 (Cited works): Identify all works cited by the paper X
– Q2.4 (Recent cited works): Identify all works cited by the paper X and published after the year Y.
– Q2.5 (Cited journal papers): Identify all journal papers cited by the paper X
– Q2.6 (Research grants): Identify the grant(s) that supported the research presented in the paper X (or part of it).
– Q2.7 (Funding agencies): Identify the funding agencies that funded the research presented in the paper X (or part of it).
– Q2.8 (EU projects): Identify the EU project(s) that supported the research presented in the paper X (or part of it).
– Q2.9 (Related ontologies): Identify the ontologies mentioned in the abstract of the paper X.
– Q2.10 (New ontologies): Identify the ontologies introduced in the paper X (according to the abstract).

2 Data Model

The output of PDF parser is written to the dataset. SPARQL queries are sent to the data of this dataset. These queries aim to provide information about paper structure, references, paper heading metadata (authors, affiliation), related projects and mentioned ontological resources. To be able to perform SPARQL queries an ontology of paper content and metadata has to be developed (see overall architecture at Fig. 2).

To develop the ontology analysis of paper content and metadata relations has to be done. CEUR website stores workshops' papers. This implies we need to introduce "Workshop" and "Paper" classes to the ontology and link them. Queries Q2.1 and Q2.2 require information about authors, their affiliations and countries so we included classes "Author", "Organization", "Country" and their relations. Queries Q2.3, Q2.4, Q2.5 concern the type of the document where the paper was published which results in adding "Document" and "Journal" classes and properties describing citation, date of publication and DOI. Queries Q2.6, Q2.7, Q2.8 concern grants, funding agencies and EU projects so corresponding classes were added to the ontology and properties describing paper funding by

the funding agency and grant attributes. Class "Ontology" and corresponding properties were added for the last two queries. As a result, the developed ontology includes the following classes: "Workshop", "Paper", "Author", "Organization", "Country", "Document", "Journal", "Grant", "Funding Agency", "EU Project" and "Ontology".

Based on the elaborated ontology we chose actual vocabularies to create the ontology model. Vocabularies were selected by their relevance and popularity. Their classes and properties have to describe relations between the objects. In contradictory situations the most frequently used vocabulary was selected.

The developed ontology for workshop papers is shown in Fig. 1. It is based on the BIBO[2] (The Bibliographic Ontology Specification). The Bibliographic Ontology Specification provides main concepts and properties for describing citations and bibliographic references (i.e. quotes, books, articles, etc.) on the Semantic Web. Classes from this ontology are used to describe papers, cited documents, authors and their organizations. The properties from this ontology are used to describe citing, reviewing in the text, publications in journals, document titles, dates and DOI. To describe relations between authors and papers the FOAF[3] (Friend of a Friend) and the DC[4] (The Dublin Core) ontologies are used. To describe author's affiliation with certain organization the SWRC[5] ontology is used [4]. The DBpedia Ontology[6] is used to describe the class of organization's country. The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ARPFO[7] (Academic Research Project Funding Ontology) ontology is used to describe classes of grants, funding agencies and EU projects. ARPFO provides classes and properties to describe the project funding structure of academic research, and also provides classes and properties to encode the relations.

3 Our Approach

Metadata extraction procedures are based on regular expressions, natural language processing methods, heuristics concerning html document style (font family, size, etc.), style of the elements of standard bibliographic description [1,3]. We combined all these methods while developing the current approach. Proposed rules were elaborated on the training dataset including LNCS and ACM templates but are not limited to them. Rules are applied to the HTML representation of the text.

In the next subsections we describe specific solutions which we applied for a particular query.

[2] Cf. http://purl.org/ontology/bibo/.
[3] Cf. http://xmlns.com/foaf/0.1/.
[4] Cf. http://purl.org/dc/elements/1.1/.
[5] Cf. http://swrc.ontoware.org/ontology.
[6] Cf. http://dbpedia.org/resource/.
[7] Cf. http://vocab.ox.ac.uk/projectfunding.

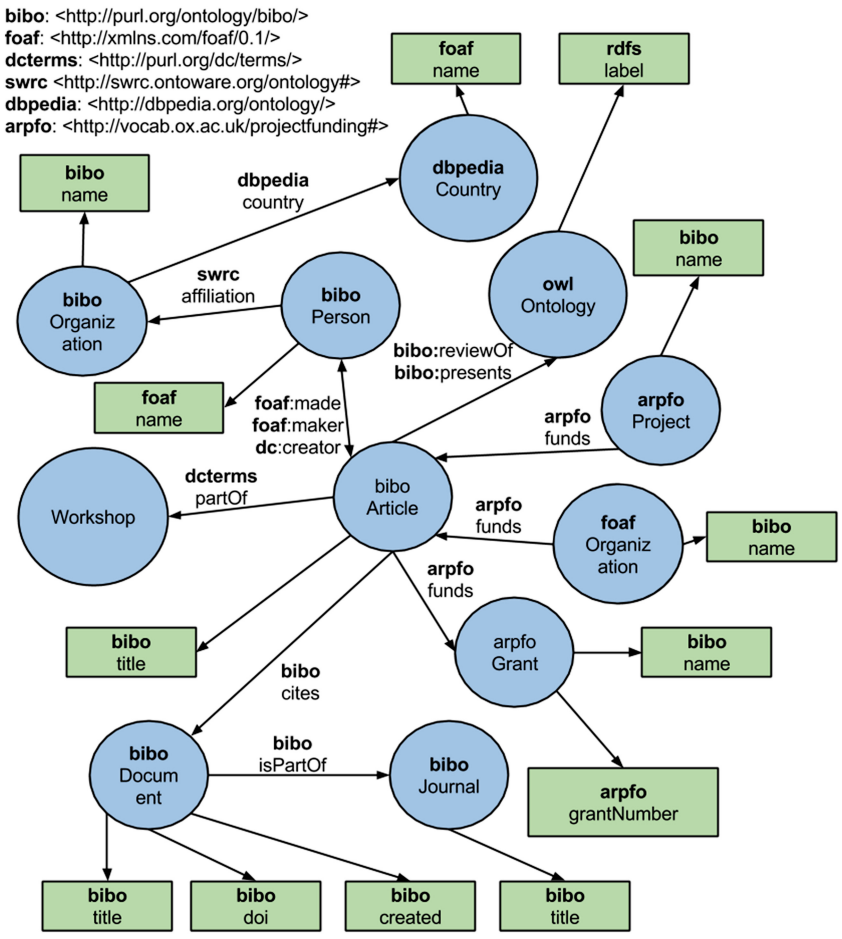

Fig. 1. The ontology for papers of workshops

Query 2.1. Queries 2.1-2.2 require heading parsing. Parsing procedure starts with splitting HTML file into pages. The heading is assumed to be the block beginning from the $<$div$>$ containing string 'Page 1' to the $<$div$>$ which has more than 30 words in it (excluding tags), because some papers do not have the 'Abstract' section. To extract the title of the paper we used font characteristics and text position on the page. HTML elements inside the headings are sorted according to the value of the 'top' property. Then the text, encapsulated in the blocks, having the same value of the 'font-family' property, is extracted as title. An example is provided below, title of the paper is 'Keynote: Listening to the pulse of our cities during City Scale Events'.

1. Example of the improper PDF -> HTML title parsing.

```
<div style=‘‘position:absolute; border: textbox 1px solid;
writing-mode:lr-tb; left:142px; top:182px; width:330px;
height:14px;’’>
 <span style=‘‘font-family: YTEDIA+CMBX12; font-size:14px’’>
Listening to the pulse of our cities during City
 <br>
 </span>
</div>
<div style=‘‘position:absolute; border: textbox 1px solid;
writing-mode:lr-tb; left:275px; top:164px; width:63px;
height:14px;’’>
 <span style=‘‘font-family: YTEDIA+CMBX12; font-size:14px’’>
 Keynote:
 <br>
 </span>
</div>
<div style=‘‘position:absolute; border: textbox 1px solid;
writing-mode:lr-tb; left:263px; top:200px; width:88px;
height:14px;’’>
 <span style=‘‘font-family: YTEDIA+CMBX12; font-size:14px’’>
 Scale Events
 <br>
 </span>
</div>
```

2. The same HTML block sorted by the ‘top’ property values.

```
<div style=‘‘position:absolute; border: textbox 1px solid;
writing-mode:lr-tb; left:275px; top:164px; width:63px;
height:14px;’’>
 <span style=‘‘font-family: YTEDIA+CMBX12; font-size:14px’’>
 Keynote:
 <br>
 </span>
</div>
<div style=‘‘position\:absolute; border\: textbox 1px solid;
writing-mode\:lr-tb; left\:142px; top\:182px; width\:330px;
height\:14px;’’>
 <span style=‘‘font-family\: YTEDIA+CMBX12; font-size\:14px’’>
 Listening to the pulse of our cities during City
 <br>
 </span>
</div>
<div style=‘‘position\:absolute; border\: textbox 1px solid;
writing-mode\:lr-tb; left\:263px; top\:200px; width\:88px;
height\:14px;’’>
```

```
<span style=‘‘font-family\: YTEDIA+CMBX12; font-size\:14px’’>
 Scale Events
 <br>
 </span>
</div>
```

Here parts of the title are mixed up: ‘Keynote:’ is inserted between ‘City’ and ‘Scale’. Proper order of title parts can be restored using ‘top’ values: in the discussed heading ‘Keynote:’ has top value 164px; ‘Listening to the pulse of our cities during City’ has top value 182px; ‘Scale Events’ has top value 200px, all title parts have font-family ‘YTEDIA+CMBX12’. Therefore, ascending sorting by the ‘top’ value settles the right sequence of title elements. After title extraction authors names and surnames are identified. We used the assumption that personal information provided in the heading is redundant. Major part of the authors choose various combinations of their name and surname as an e-mail nickname. Local part of the e-mail address frequently matches the following patterns:

– ‘name.surname’ (‘john.smith’),
– ‘first_symbol_of_the_name.surname’ (‘j.smith’),
– ‘surname’ (‘smith’)

Therefore, information in the local part of the e-mail address can be used to find the string containing author name.

The block after the title is split into tokens by spaces, dots, commas and colons. Then all emails are extracted and split into local and domain parts. Local parts are split by dot into tokens, the latest are searched above the e-mail in the heading. If a token matches a substring, this substring is considered as a candidate for the person’s name or surname. So a block of text between the authors name and surname is extracted. This block contains affiliation. This approach is useful for the case when there are no digits pointing to the affiliation (like in LNCS template). The procedure covers 2 types of e-mail parsing: (1) authors with the same affiliation have separate e-mails (like in ACM template), (2) authors with the same affiliation have local names listed in figure brackets and common domain name. For the second case affiliation is duplicated for each author. Multiple affiliation is parsed using digits in the heading.

Query 2.2. To identify the papers presented at the workshop X and written by researchers affiliated to an organization located in the country Y, affiliation is parsed from the last symbol, comma is used as a delimiter between the tokens (thus a token may consist of more than one space-separated item, e.g. Czech Republic, United Arab Emirates). Candidate token is checked via the countries list whether it is a country. For some cases like “NY USA” only a substring of the token refers to the country name and if validating procedure returned no country, the token is split by spaces and validation is iterated for the last item.

Block of queries 2.3-2.5 requires reference parsing. The file is scanned for the first occurrence of ‘references’ or ‘bibliography’ keyword (case is ignored).

References' block is extracted starting from the found keyword to the end of HTML file. Every bibliographic reference is split into its elements: authors, title, title of periodical or conference, imprint details (publisher, publishing place, year). Beginning from the word "References" to the end of file document space is split into separate references by the paper's number.

Query 2.3. Firstly, it is necessary to set the boundaries of each bibliography item. It may start with a digit (or single token - a surname - followed by a digit) enclosed in square brackets or, if a digit is not enclosed with squares, a dot follows it (it also obligatory appears at the beginning of a line). The end of each bibliographical item is the beginning of the next one. From each reference the following data are extracted: year, title, and the name of journal it is published in (0 - if it is not a journal paper). Year matches a plain regular expression: any four digits match the year.

The title begins from the first capital letter after the end of the authors block. There were found several templates for cited paper authors extractions. They are:

1. J. Conesa "[A-Z]\. [A-Z][a-z]+". The first capital letter after the last matched regular expression is the beginning of a title
2. J. Conesa "[A-Z][a-z]+ [A-Z]\". The first capital letter after the last matched regular expression is the beginning of a title.

If the beginning of the bibliographical item doesn't match these regular expressions, we found the first dot in the string. The title ends with dot or double quote.

To identify the name of the journal we use the regular expression ", \d+(\d+)". The part of the bib item between the beginning of this regular expression and end of the title identified at the previous step.

Query 2.4. To identify all works cited by the paper X and published after the year Y, a procedure addresses the dictionary where reference attributes are stored and checks the year of the paper.

Query 2.5. To identify all journal papers cited by the paper X, principles of bibliographic reference composing are used. When a journal paper is cited, volume number and issue number are given. The last is given in round brackets, so to identify that a paper is published in the journal, part of the reference should match the regular expression "[0-9]+([0-9]+)". The journal title is a sequence between paper title and the sequence that has matched this regular expression. This rule allows to extract journals when no lexemes point to it, e.g. 'Cognitive Linguistics 4(2)'. If this regular expression returned no matches, the string is scanned for keywords 'J. —Journal—Annals—Letters'. If the latest are found, journal title is extracted as the sequence including the keyword, from the end of the title to the first space+digit combination or space + uppercase 'V' after the keyword (e.g. '*J. Data Semantics* V: 64-90', '*Annals of Pure and Applied Logics* 123', '*Information Processing Letters*, 74').

Group of queries 2.6-2.8 is performed over the 'Acknowledgements' section. The section is split into sentences. Firstly, grant numbers are identified and removed from the sentence, then a group of context-free patterns to extract funding agencies is applied, EU-funded projects are extracted after it and, finally, the rest of the funding agencies is extracted.

Query 2.6. Grant number may combine several identification elements, such as type code, activity code, institute code, serial number, support year, etc. We relied that serial number contains at least 3 digits. Therefore, grant number may contain only digits, digits and literals, hyphens and slashes. The regular expression extracting grant number, matches a token containing at least 3 digits and obligatory having a digit as its last symbol.

Query 2.7. To identify the funding agencies that funded the research presented in the paper X we started with testing Stanford Named Entity Tagger[8] but found out it does not extract all funding agencies we need. In the following examples from the training dataset paper[9] "Christian Doppler Forschungsgesellschaft" was recognized as person, "Österreichischer Austauschdienst" was not recognized at all, and "Federal Ministry of Economy, Family and Youth" was partially extracted. There are some other examples where this tool is not precise.

Example. This work was supported by the Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy, Family and Youth, Österreichischer Austauschdienst (ÖAD) and the National Foundation for Research, Technology and Development - Austria.

Considering that acknowledgements section is written in a highly standardized manner, each sentence in the 'Acknowledgements' section was scanned for the stems "support|fund|sponsor". If the sentence contained this stems two group of patterns were applied to extract funding agencies. The group of context-free patterns are applied first. These patterns are:

1. 'by ORGANIZATION under',
2. 'funding from ORGANIZATION under',
3. 'by ORGANIZATION in',
4. 'by ORGANIZATION within',
5. 'by ORGANIZATION-funded| funded'.

If nothing was extracted with these patterns, the procedure switches to project extraction, then returns to scan the sentence for the remained funding agencies. On this stage the sentence is scanned for the keywords 'by|funding| funding from', their indices are returned. Starting from the found keyword to the end of the sentence word sequences having at least one symbol in uppercase (excluding prepositions *of, and, for* are extracted. Candidates for funding agencies are split by comma. For long organization titles special regular expressions are reserved:

[8] Cf. http://nlp.stanford.edu/software/CRF-NER.shtml.
[9] Cf. http://ceur-ws.org/Vol-1155/paper-06.pdf.

– $[A-Z][a-z]+[A-Z][a-z]+$ of $[A-Z][a-z]+, [A-Z][a-z]+$ and $[A-Z][a-z]+$ ('Federal Ministry of Economy, Family and Youth'),
– $[A-Z][a-z]+[A-Z][a-z]+$ for $[A-Z][a-z]+, [A-Z][a-z]+$ and $[A-Z][a-z]+$ ('National Foundation for Research, Technology and Development').

However, an alternative rule can be formulated to extract long titles. An extra condition has to be specified, that a funding agency whole title should have no less than 2 symbols in uppercase. In case it has 2 or less symbols they should not contain "and". Otherwise, this candidate is merged to the previous one.

Query 2.8. To select EU-funded projects keywords and keyphrases pointing to the European Union and its programmes are searched in each sentence of the 'Acknowledgement' block. The following elements were used to write the regular expressions :

– 'EU-funded',
– 'EU FP\d'
– 'FP\d European'
– 'European Union'
– 'FP\d'
– 'EU \dth Framework Program'
– 'European Union \dth Framework Program'

When any of these elements is found, its index is used. On the distance of -4; +4 tokens from this element the sentence part is scanned for a sequence(s) of tokens (or a single token) having at least one symbol in uppercase (e.g., 'NewsReader', 'LOD2', 'DM2E', 'Dr Inventor'). Token is defined as a sequence between the spaces.

Query 2.9 and Q2.10. For procedures in queries 2.9-2.10 we used a stop-list of acronyms and abbreviations related to semantic web (including ontology languages, Semantic Web standards, etc.) to avoid their extraction as candidates for ontology name and Stanford Parser[10] to do syntactic analysis of the sentence in order to remove false candidates in Q2.10. A list of existing ontologies was also used. These two queries are performed on the 'Abstract' section. Firstly, ontologies in the predefined list are searched in each sentence of the abstract and written as the output for query 2.9. Then, if a sentence includes stem 'ontolog', this sentence is sent to the Stanford Parser for syntactic analysis. The Parser returns a list of dependencies between the words in the sentence. Then it is checked, whether there is a dependency between 'introduce|present|propose|describe' and the word having 'ontolog' as a substring. If such dependency exists, part of the sentence at -5;+5 distance from the stem 'ontolog' is scanned for a word sequence where each word has at least one symbol in uppercase. It is extracted and written as a new ontology, mentioned in the abstract. If Stanford Parser gives no dependency between the words, mentioned above, part of the sentence at -5;+5

[10] Cf. http://nlp.stanford.edu/software/lex-parser.shtml.

distance from the stem 'ontolog' is scanned for a word sequence where each word has at least one symbol in uppercase. Such sequence(s) is written as the output of query 2.9.

To find new ontologies, hyperlinks in the 'Abstract' were also parsed. We supposed, that a link given in the abstract may identify a new project. So hyperlink's body was scanned for having 'onto' as a substring and (if true) was split by dot (similarly to the splitting of e-mails in Q2.1). Sentences in the abstract were lowercased and parts of hyperlink body were searched there. If any part was found, we returned the corresponding token from the original sentence. An example is given below. Original sentence is 'BioPortal, a web-based library of biomedical ontologies.available online at http://bioportal.bioontology.org.' includes a hyperlink. This is the list with the elements to be searched as ontology name: ['bioportal', 'bioontology']. Having obtained the index, mentioned ontology name is returned: 'BioPortal'.

4 Implementation

4.1 Overall Architecture

The tool is implemented in Python 2.7. Developed tool uses Grab Spider framework[11]. This framework allows to build asynchronous site crawlers. Crawler downloads all workshop's papers and then runs the parsing tasks. The Paper Parser uses the Metadata Extraction Library to gather information about the paper. The tool uses the Ontology Mapper module to build properties and entity relations. The Ontology Mapper module uses RDFLib[12] library to create and store triples. The overall architecture of the developed tool is shown in Fig. 2.

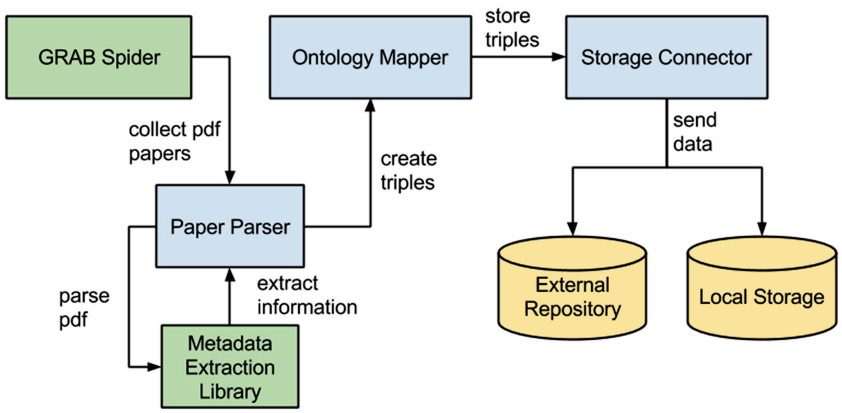

Fig. 2. The overall architecture of the developed tool

[11] Cf. http://grablib.org/.
[12] Cf. https://github.com/RDFLib.

4.2 Library for Context Information Extraction from the PDF Full Text of the Papers

The metadata extraction has several steps. At the first step the input PDF is converted into TXT and HTML format. This conversion is made with pdf2txt utility (a part of Python PDFminer library[13]). Then the metadata extraction library deals with obtained HTML and TXT only. We use BeautifulSoup[14] as a HTML parser library. The module is implemented using Python 2.7.

5 Results and Discussions

The developed tool produces a LOD dataset in the output, which stores information about cited papers, affiliations, agencies, mentioned ontologies, some other objects and relations for each paper. The tool was tested on the training dataset of 12 workshop proceedings having total 101 papers. Testing was accomplished by running original automated tests. Automated tests use SPARQL queries to collect information from dataset and check for equality with manually predefined results stored in CVS format. For example to identify all journal papers cited by the paper http://ceur-ws.org/Vol-1302/paper7.pdf (Q2.5) the following query should be send.

```
SELECT ?resource_iri ?doi ?paper_title ?journal_title {
 VALUES ?paper_iri {
    <http://ceur-ws.org/Vol-1302#paper7>
 }
 ?paper_iri bibo:cites ?resource_iri .
 ?resource_iri  bibo:isPartOf ?journal_iri .
 ?resource_iri  bibo:title ?paper_title .
 ?journal_iri  bibo:title ?journal_title
 OPTIONAL {?resource_iri  bibo:doi ?doi}
}
```

6 Conclusion

Analysis of workshop paper elements resulted in accomplishing the following tasks:

- development of ontology describing paper metadata and mentioned resources and named entities;
- development of metadata extraction procedures;
- development of the tool crawling PDF papers, applying metadata extraction procedures and publishing results as Linked Open Data.

[13] Cf. https://pypi.python.org/pypi/pdfminer/.
[14] Cf. https://pypi.python.org/pypi/BeautifulSoup/3.2.1.

Task 2 of Semantic Publishing Challenge 2015 is solved with the developed tool based on Grab Spider framework, RDFLib library and the developed library for metadata and context information extraction from the PDF full text of the papers. This tool uses BIBO, FOAF, SWRC, ARPFO and DBpedia ontologies. Metadata Extraction Library uses regular expressions based on html page style attributes, natural language processing methods, heuristics about acronym resolving and named entities extraction. Further work implies improvement of named entities extraction procedures, performing deeper syntactic analysis, adding external data sources, using validation via external sources, e.g. heading elements except e-mail can be validated via DBLP[15], candidates for mentioned ontologies can be also checked via external source[16].

Acknowledgments. This work has been partially financially supported by the Government of Russian Federation, Grant #074-U01.

References

1. Guo, Z., Jin, H.: Reference Metadata Extraction from Scientific Papers. In: 2011 12th International Conference on Applications and Technologies Parallel and Distributed Computing (PDCAT), pp. 45–49, October 2011
2. Kolchin, M., Kozlov, F.: A template-based information extraction from web sites with unstable markup. In: Presutti, V., Stankovic, M., Cambria, E., Cantador, I., Di Iorio, A., Di Noia, T., Lange, C., Reforgiato Recupero, D., Tordai, A. (eds.) SemWebEval 2014. CCIS, vol. 475, pp. 89–94. Springer, Heidelberg (2014). http://dx.doi.org/10.1007/978-3-319-12024-9_11
3. Marinai, S.: Metadata extraction from pdf papers for digital library ingest. In: 2009 10th International Conference on Document Analysis and Recognition, ICDAR 2009, pp. 251–255, July 2009
4. Sure, Y., Bloehdorn, S., Haase, P., Hartmann, J., Oberle, D.: The SWRC ontology semantic web for research communities. In: Bento, C., Cardoso, A., Dias, G. (eds.) EPIA 2005. LNCS (LNAI), vol. 3808, pp. 218–231. Springer, Heidelberg (2005). http://dx.doi.org/10.1007/11595014_22

[15] Cf. http://dblp.uni-trier.de.
[16] Cf. http://prefix.cc.