Unstable markup:
A template-based information extraction from web sites with unstable markup

Maxim Kolchin and Fedor Kozlov

ITMO University
kolchinmax@niuitmo.ru, kozlovfedor@gmail.com

Abstract. This paper presents results of a work on crawling CEUR Workshop proceedings[1] web site to a Linked Open Data (LOD) dataset in the framework of Semantic Publishing Challenge 2014[2]. Our approach is based on so-called "templates of web site' blocks" and DBpedia for crawling and linking extracted entities.

1 Introduction

The work that is presented in this paper aims to provide a solution for Task 1 of Semantic Publishing Challenge 2014[2]. The task is to crawl CEUR Workhop proceedings web site[1] and create a LOD dataset containing detailed information about workshops, proceedings volumes, papers and their authors and etc.

The source code and instructions to run the crawler are located at our Github repository[3].

1.1 Challenges

At first glance, the task looks a pretty straightforward, but there are several challenges that need to be solved:

- the web site has a quite unstable and in some cases invalid HTML markup because of absence of a standardised and strict template for creation of pages for proceedings volumes, so it makes it harder to crawl such pages, because usually crawlers are written for web sites with fixed markup;
- only a small percentage of proceedings volumes uses RDFa markup and microformats are used only for volumes starting from 559th one, so at the time of writing around 49% of volume pages don't have any metadata that could help in crawling;

[1] CEUR Workshop proceedings, URL: http://ceur-ws.org
[2] Semantic Publishing Challenge 2014, URL: http://2014.eswc-conferences.org/semantic-publishing-challenge
[3] The source code and instructions, URL: https://github.com/ailabitmo/sempubchallenge2014-task1

– according to the rules of the web site, proceedings should comply with some requirements regarding numbers of invited and regular papers, therefore there are joint proceedings of several workshops. Such workshop and proceedings should be represented in the dataset accordingly;
– the web site includes proceeding not only in English, but also in German. In addition it's quite common practise for authors of papers written in English to use names of their universities or companies in a native language;

2 Our approach

We developed a crawler based on "templates of web site' blocks" approach. It uses sets of special predefined templates for each type of entity. The main aim of this templates is to cover entire variety of entity representations in HTML format. Some of templates used for extracting papers from workshops pages are:

– a template based on RDFa metadata,
– a template based on Microformats,
– and two templates specific to some similar HTML markups .

When HTML page parsing begins, the crawler consecutively runs predefined templates till one of the templates returns the valid data. Validation based on template's structure. Template's parsing process extracts data from HTML page using XPath Language and regular expressions. XPath Language is used for searching text data by elements and properties in HTML markup. Regular expressions is used for extracting entity tokens from plain text. When data is extracted template parsing process converts data into ontology instances and properties.

The main advantage of our approach is a flexibility of different data representations in HTML markup with usage of the same code of the crawler and support of invalid HTML.

2.1 Architecture

The parser is implemented in Python and based on Grab Spider framework[4]. This framework allows to build asynchronous site crawlers. Crawler downloads all workshop's pages and papers and then runs the parsing tasks. There is a collection of specific parsers for each entity. Each parser in collection process a part of some HTML page to build properties and entity relations.

The overall system architecture is shown in fig. 1.

[4] Grab framework, URL: `http://grablib.org/`

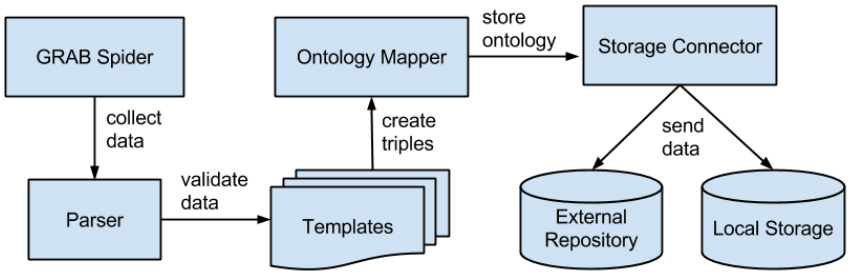

Fig. 1. The overall system architecture

2.2 Data representation

To represent crawled data we use several different ontologies such as Semantic Web Conference Ontology (SWC)[5], Semantic Web for Research Communities ontology (SWRC)[6], The Bibliographic Ontology (BIBO)[7], The Timeline Ontology (TIMELINE)[8], Friend of a Friend (FOAF)[9], Dublin Core (DC and DCTERMS) [10] and DBpedia Ontology (DBPEDIA-OWL)[11] and RDF Schema (RDFS)[12].

A part of the data representation schema is shown on Fig. 2. Representation of time and time intervals doesn't use The Event Ontology (EVENT) because it assumes inclusion of blank nodes. Since RDFLib doesn't work well with them we decided to use TIMELINE ontology instead. TIMELINE ontology provides timeline:atDate property for setting a date to an instance and *timeline:beginsAtDateTime* and *timeline:endsAtDateTime* properties for a time interval.

On CEUR Workshop proceedings web site some proceedings volumes has links to each other. These links usually relate a proceedings of a workshop to the previous its editions and we uses *skos:related* property to represent this relationships.

2.3 Specific solutions

In most cases all problems are solved by an appropriate template of a block, but there are some problems requiring specific solutions.

[5] Semantic Web Conference Ontology, URL: `http://data.semanticweb.org/ns/swc/ontology`
[6] Semantic Web for Research Communities, URL: `http://ontoware.org/swrc/`
[7] The Bibliographic Ontology, URL: `http://purl.org/ontology/bibo/`
[8] The Timeline Ontology, URL: `http://purl.org/NET/c4dm/timeline.owl#`
[9] The Friend of a Friend (FOAF), URL: `http://www.foaf-project.org/`
[10] Dublin Core, URL: `http://purl.org/dc/elements/1.1/`
[11] DBpedia Ontology, URL: `http://dbpedia.org/ontology/`
[12] RDF Schema, URL: `http://www.w3.org/2000/01/rdf-schema#`

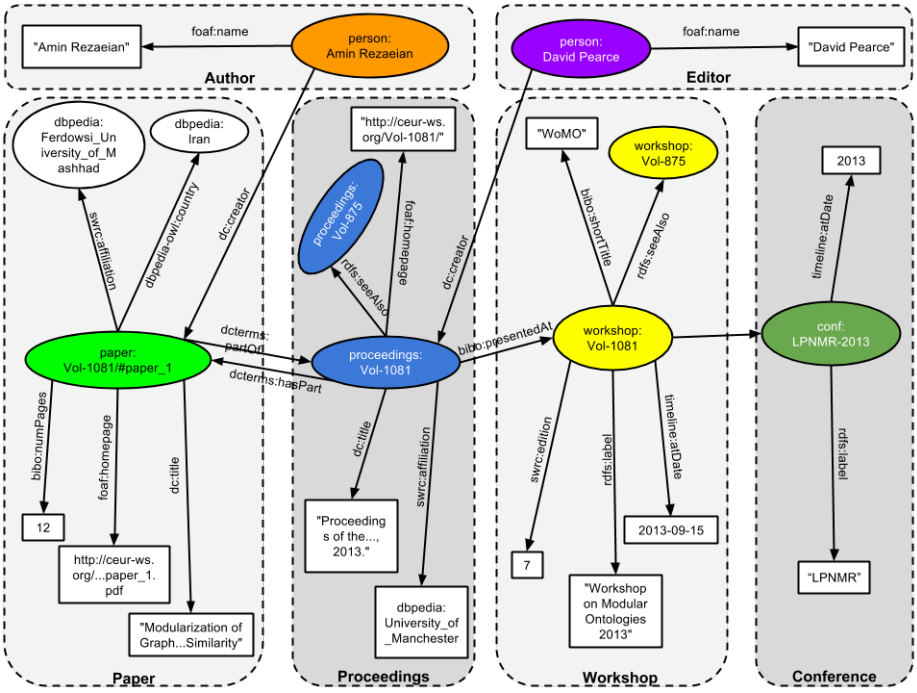

Fig. 2. Schema representing the crawled data

Extraction of countries and affiliations Identification of countries and affiliations in papers was done with external datasets. In case of extracting countries parser extracts the first page from the PDF document. Country-candidates are extracted using regular expressions with predefined templates. Parser sends request with a list of country-candidates to SPARQL-endpoint of DBpedia[1] resource to get list of unique country's IRIs.

The country extraction query must support different naming conventions of country-candidates. Hence the following SPARQL-query is suggested.

```
SELECT DISTINCT ?country {
    VALUES ?search { "The Netherlands" }
    ?country a dbpedia-owl:Country .
    { ?name_uri dbpedia-owl:wikiPageRedirects ?country ;
                rdfs:label ?label .
    }
    UNION
    { ?country rdfs:label ?label }
    FILTER( STR(?label) = ?search )
}
```

Creation of properties and relations for current paper entity is based on received list.

Identification of related workshops As mentioned above, in most cases *skos:related* property is used to relate to a previous edition of the corresponding workshops. But sometimes it's not correct. Especially in case of joint proceedings. To identify correct links we implemented the algorithm measuring similarity of two workshops based on its full name and acronym. In case of absence of an acronym we generate one from the full name's upper case characters. For example for "Concept Extraction Challenge at Making Sense of Microposts 2013" workshop the "CECMSM" acronym is generated. String similarity measurement uses the basic Ratcliff-Obershelp algorithm[2]. This algorithm was selected because it is being provided by the Python Standard Library.

3 Conclusion

Task 1 of Semantic Publishing Challenge 2014 is solved with developed parser based on Grab Spider framework. This parser uses SWC, SWRC, BIBO, TIMELINE ontologies, DBpedia datasets and the basic Ratcliff-Obershelp algorithm for string similarity measurement. Our approach based on templates of web site blocks, the schema representing extracted information and solutions for some specific problems. The main advantages of our approach are flexible representation of different data templates in HTML markup and support of invalid HTML.

3.1 Unsolved issues

In most cases our solution works well, but there are several "places" where it doesn't work well and therefore may not pass some tests completely:

- extraction of country and university candidates from papers works only for texts consisting only of US-ASCII characters because PDFMiner[13] which we use to extract text from PDF files doesn't work well with Unicode symbols;
- papers written in PostScript or HTML are completely ignored;

References

1. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal (2014)
2. Ratcliff, J.W., Metzener, D.E.: Pattern-matching-the gestalt approach. DR DOBBS JOURNAL 13(7), 46 (1988)

[13] PDFMiiner, URL: `http://www.unixuser.org/~euske/python/pdfminer/`