Doc2RDFa: Semantic Annotation for Web Documents

Martin Beno, Erwin Filtz, Sabrina Kirrane, and Axel Polleres

Vienna University of Economics and Business,
Vienna, Austria

Abstract. Ever since its conception, the amount of data published on the world-wide web has been rapidly growing to the point where it has become an important source of both general and domain specific information. However, the majority of documents published online are not machine readable by default. Many researchers believe that the answer to this problem is to semantically annotate these documents, and thereby contribute to the linked "Web of Data". Yet, the process of annotating web documents remains an open challenge. While some efforts towards simplifying this process have been made in the recent years, there is still a lack of semantic content creation tools that integrate well with information worker toolsets. Towards this end, we introduce Doc2RDFa, an HTML rich text processor with the ability to automatically and manually annotate domain-specific content.

Keywords: Information retrieval, Semantic web, RDFa

1 Introduction

Advancements in the field of web technologies over the past two and a half decades have led to an exponential rise in the amount of information available online, such as Open Government Data Portals or publicly available legal databases, often only published as PDFs or HTML [1], both of which are not designed to be machine-readable by default. One possible means to unlock the knowledge stored in textual documents is to add semantic annotations to the documents [9], thus making them machine-readable. Although the benefits of semantically annotated documents having been widely recognized [11], there is still a vast number of web documents without any semantic annotations being published every day, likely because adding semantic annotations to documents is a laborious, error-prone, challenging task [9]. The legal domain for instance, holds a large amount of unannotated text documents, possibly because the adoption of automated annotation systems has yet to happen in that domain. That is why we propose a domain-specific system for the legal domain to increase the searchability and interlinking of legal information by using semantic annotations. As such, it would be highly beneficial to have a tool which could be used to create, modify and publish documents in a format that allows the document to be enhanced with additional semantic information. In order to address this gap, we introduce our Doc2RDFa tool, which can be used to automatically annotate web documents, in a manner that enables them to be automatically added to the linked "Web of Data". The contributions described in this paper can be summarized as follows: (i) we extend an existing open source web-based rich text processor in

a manner that caters for the embedding of metadata into web documents using RDFa; (ii) we provide a user-friendly interface for the creation and automatic annotation of web documents; and (iii) we further enhance the tool with Natural Language Processing (NLP) such that metadata can be automatically extracted from domain-specific web based documents. Previous work in this area shows that there is a need for semantic content authoring tools [7], but annotating documents is a time consuming process which often requires domain-specific knowledge of experts [4, 10]. Existing general purpose tools, like RDFaCE [8] or Loomp [6], lack accuracy when applied in a domain-specific context. In the remainder of this paper we provide a short description of the use case in Section 2. In Section 3 we describe the front-end and back-end as well as the workflow of our system. The paper concludes with a summary and future work in Section 4.

2 Use Case

Legal documents residing in legal databases can often be viewed and downloaded in HTML format. However, this serialization does not contain semantic information which is necessary to make web documents machine-readable. For instance, semantic annotations in the form of the keywords describing the case, information pertaining to the deciding court, relevant dates or referenced laws if available could be used to improve search over legal data. As such, there is a need for an all encompassing system that can be used to create, download, display, annotate, and store web documents. Web documents can be represented using the W3C Resource Description Framework in attributes (RDFa) serialization, which allows for the embedding of semantic information inside HTML documents. When it comes to automatic text annotation the focus is put on the text analytics as opposed to the user interface. However in the case of manual annotations a rich text processor is highly desirable as it is the tool of choice used by many domain specialists. With this in mind, and by drawing upon existing literature, we have defined a set of requirements for our annotation system: (i) User friendliness: An essential aspect of the editor is the user-friendliness of the interface. The tool should be usable by users not familiar with the concepts of Semantic Web. A view also held by Heese et al. [6], who claim that the complexities of the annotator need to be hidden away from the users. (ii) Familiarity: It should be possible to both write and annotate documents using the same front-end interface. Khalili and Auer [7] argue that by integrating the annotation system in the same environment, in which the documents are created, minimizes user actions, thereby increasing the efficiency, user satisfaction, learnability and utility of the system. (iii) Automation: The automatic annotation system should be able to accurately annotate the document, thus requiring little to no manual effort from the user.

3 Doc2RDFa

In this section we provide an overview of the core components of Doc2RDFa: (i) the web-based rich text word processor; and (ii) the NLP and information extraction tool. Following on from this we describe how users typically interact with the system.

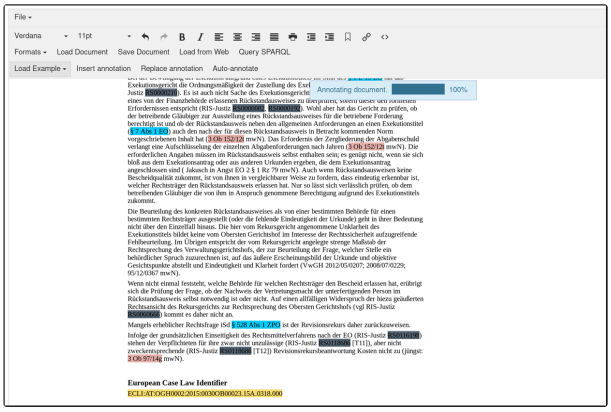

Fig. 1. Doc2RDFa Editor

3.1 Annotation Aware Text Editor Front-End

With the advent of Web 2.0 and HTML5, document creation has been increasingly moving towards browser based solutions [5]. A major benefit of web applications is the fact that they are operating system agnostic. The only requirement is a modern browser supporting HTML5 and JavaScript features. Furthermore, many web-based rich text editors support the creation of plain HTML text files, the presentation of which can then further be adapted with Cascading Style Sheets (CSS). Both HTML and CSS are open standards, maintained by the World Wide Web Consortium (W3C). We opted for building upon an existing rich text editor (TinyMCE)[1], which is still in development, can be easily customized, has an active community contributing plugins, and is licensed under the LGPL license. We extend the editor shown in Fig. 1 by implementing a number of features and style formats to facilitate the creation of annotated web documents. Documents written in TinyMCE are saved as plain text HTML files. In the recent years, there have been various approaches used to annotate HTML documents, such as using Microdata, JSON-LD, and RDFa. We annotate the documents using RDFa, a W3C recommendation. The Resource Description Framework (RDF), which underpins the Linked Data Web, can be used to represent and link information, in a manner which can be interpreted by both humans and machines. The main benefit of using RDFa over Microdata is that the metadata are self-contained in the HTML source code, eliminating the need to separate the semantic annotations from the web document markup.

3.2 Natural Language Processing Back-End

The back-end is responsible for not only loading and saving annotations, but is also enriched with NLP capabilities such that it is possible to automatically propose document annotations (e.g., named entities, temporal information). Here we use the General Architecture for Text Engineering (GATE) [3], an NLP tool, which is capable of reading

[1] `https://www.tinymce.com/`

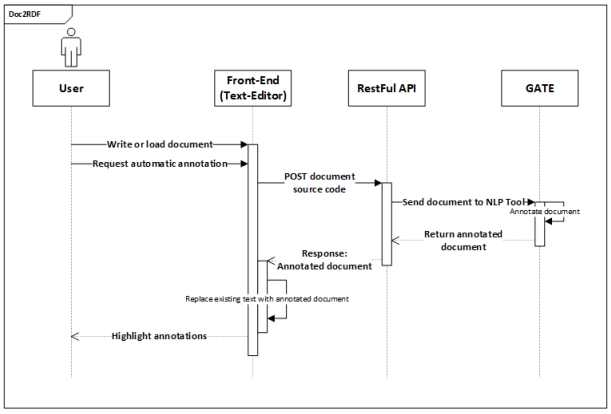

Fig. 2. Doc2RDFa workflow

documents in various formats and subsequently annotating them in one or more corpora. In particular, we leverage the Java Annotations Pattern Engine (JAPE) [2] in order to extract information contained in the legal documents. JAPE rules are based on the tokens produced by a tokenizer at first, splitting the texts in individual tokens of different types, for instance word, number or punctuation. A JAPE rule consists of a left-hand side (LHS) specifying the extraction rules and a right-hand side (RHS) to add the annotation to the document and specify annotation features, containing the property we use for this information according to the applied ontology as well as a background color to highlight the annotation in the front-end. Given that the editor itself is domain independent the extraction rules need to be configured for each specific domain. The extension to other domains therefore requires adding new extraction rules for common patterns. Furthermore, we are not restricted to patterns, for which JAPE rules are the first choice. When it comes to a limited amount of words or concepts that should be detected, for instance the courts in a jurisdiction, the use of gazetteers is recommended which contain a list of words that are looked up. The use of gazetteers is only recommended for small sets of lookup words or concepts that do not change regularly and are of a maximum length such that the lists could be managed and updated manually. The third extension possibility our system provides is to use machine learning techniques for annotation.

3.3 Workflow

The workflow of Doc2RDFa is shown in Figure 2. The process starts with the user interacting with the front-end text editor. HTML documents can be loaded from either local storage or retrieved directly from a remote web server. Once a document has been either written or loaded, the user starts the automatic annotation process by pressing the "Auto Annotate" button. The source code of the document is then sent to the RESTFul API and loaded by the GATE NLP, which annotates the document based on a set of pre-defined grammar rules. The RESTful API then sends the annotated HTML code as a response back to the client. Once the client retrieves the annotated document it

loads its contents and replaces the original unannotated HTML code. The annotated text is highlighted in the editor, enabling the user to check whether everything has been correctly annotated. Finally, missing, or incorrect annotations can be manually fixed using the "Insert Annotation" button.

4 Summary and Future Work

In the present paper, we introduced Doc2RDFa, an HTML rich-text processor with the ability to automatically and manually annotate domain-specific content. We subsequently discussed how Doc2RDFa can be used to automatically annotate legal documents that can be integrated in a pipeline of legal corpus creation process, or used to modify existing web documents. For future work we aim to extend the editor with additional features, such as support for triple store databases, and enhancing the search facility to include faceted browsing.

References

1. Beno, M., Figl, K., Umbrich, J., Polleres, A.: Perception of key barriers in using and publishing open data. JeDEM-eJournal of eDemocracy and Open Government 9(2), 134–165 (2017)
2. Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine (1999)
3. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with gate's full lifecycle open source text analytics. PLOS Computational Biology 9(2), 1–16 (02 2013), `https://doi.org/10.1371/journal.pcbi.1002854`
4. Gangemi, A.: A comparison of knowledge extraction tools for the semantic web. In: Extended Semantic Web Conference. pp. 351–366. Springer (2013)
5. Godwin-Jones, R.: Emerging technologies: Web-writing 2.0: Enabling, documenting, and assessing writing online. Language Learning & Technology 12(2), 7–13 (2008)
6. Heese, R., Luczak-Rösch, M., Paschke, A., Oldakowski, R., Streibel, O.: One click annotation. In: SFSW (2010)
7. Khalili, A., Auer, S.: User interfaces for semantic authoring of textual content: A systematic literature review. Web Semantics: Science, Services and Agents on the World Wide Web 22, 1–18 (2013)
8. Khalili, A., Auer, S., Hladky, D.: The rdfa content editor-from wysiwyg to wysiwym. In: Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual. pp. 531–540. IEEE (2012)
9. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J.R., Mylopoulos, J.: Text mining through semi automatic semantic annotation. In: International Conference on Practical Aspects of Knowledge Management. pp. 143–154. Springer (2006)
10. Neves, M., Leser, U.: A survey on annotation tools for the biomedical literature. Briefings in bioinformatics 15(2), 327–340 (2012)
11. Shadbolt, N., Berners-Lee, T., Hall, W.: The semantic web revisited. IEEE intelligent systems 21(3), 96–101 (2006)