OntosFeeder – A Versatile Semantic Context Provider for Web Content Authoring

Alex Klebeck[1], Sebastian Hellmann[2], Christian Ehrlich[1], and Sören Auer[2]

[1] Ontos GmbH, Poetenweg 49, Leipzig, Germany
{alex.klebeck,christian.ehrlich}@ontos.com
http://ontos.com

[2] Universität Leipzig, Institut für Informatik, AKSW,
Postfach 100920, D-04009 Leipzig, Germany
{hellmann,auer}@informatik.uni-leipzig.de
http://aksw.org

Abstract. As the amount of structured information available on the Web as Linked Data has reached a respectable size. However, the question arises, how this information can be operationalised in order to boost productivity. A clear improvement over the keyword-based document retrieval as well as the manual aggregation and compilation of facts is the provision of contextual information in an integrated fashion. In this demo, we present the *Ontos Feeder* – a system serving as context information provider, that can be integrated into Content Management Systems in order to support authors by supplying additional information on the fly. During the creation of text, relevant entities are highlighted and contextually disambiguated; facts from trusted sources such as *DBpedia* or *Freebase* are shown to the author. Productivity is increased, because the author does not have to leave her working environment to research facts, thus media breaks are avoided. Additionally, the author can choose to annotate the created content with *RDFa* or *Microformats*, thus making it "semantic-ready" for indexing by the new generation of search engines. The presented system is available as Open Source and was adapted for *WordPress* and *Drupal*.

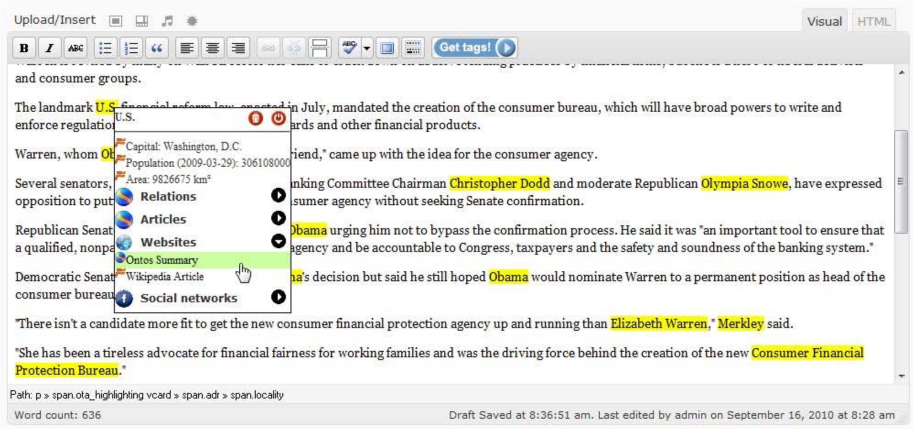
1 Introduction

One of the routine tasks of a content author (e.g. a journalist) during the time of writing is researching for context information required for the intended article. Without proper tool support, the author has to resort to manual searching (e.g. Google) and skimming through available information sources. The availability of structured data on the Semantic Data Web allows to automate these routine activities by identifying topics within the article with the aid of Natural Language Processing (NLP) and subsequently presenting relevant context information by retrieving descriptions from the Linked Open Data Web (LOD).

We present the *Ontos Feeder*[1] – a system serving as context information provider, that can be integrated into Content Management Systems in order to

[1] http://www.ontos.com

G. Antoniou et al. (Eds.): ESWC 2011, Part II, LNCS 6644, pp. 456–460, 2011.



and consumer groups.

The landmark U.S. in July, mandated the creation of the consumer bureau, which will have broad powers to write and enforce regulatio ards and other financial products.

Capital: Washington, D.C.
Population (2009-03-29): 306108000
Area: 9826675 km²

Warren, whom Ob iend," came up with the idea for the consumer agency.

Several senators, nking Committee Chairman Christopher Dodd and moderate Republican Olympia Snowe, have expressed opposition to put sumer agency without seeking Senate confirmation.

Republican Senat Obama urging him not to bypass the confirmation process. He said it was "an important tool to ensure that a qualified, nonpa agency and be accountable to Congress, taxpayers and the safety and soundness of the banking system."

Democratic Senat a's decision but said he still hoped Obama would nominate Warren to a permanent position as head of the consumer bureau

"There isn't a candidate more fit to get the new consumer financial protection agency up and running than Elizabeth Warren," Merkley said.

"She has been a tireless advocate for financial fairness for working families and was the driving force behind the creation of the new Consumer Financial Protection Bureau."

Path: p » span.ota_highlighting vcard » span.adr » span.locality

Word count: 636

Draft Saved at 8:36:51 am. Last edited by admin on September 16, 2010 at 8:28 am

Fig. 1. Entities are highlighted in the WYSIWYG editor of the CMS, Pop-ups allow to select further information

support authors by supplying additional information on the fly. Ontos Feeder uses the Ontos Web Service (OWS, see Section 3) to analyse the article text and retrieve *Ontos Entity Identifier* (OEI) URIs for relevant topics. These OEIs are interlinked with several data sources on the web and enriched with internal facts from the *Ontos Knowledge Base*. The Feeder is open-source and currently available for the CMS (Drupal[2] and Wordpress[3]. Additionally, the Feeder can automatically annotate the article with Microformats and RDFa annotations. These are increasingly utilized by search engines such as Google or Bing [4].

2 Feature Description and User Interface Walkthrough

The content creation process begins with the writing of an article in a supported CMS system. Having written the content, the author clicks on the `get tags` button to send the text to the OWS. The OWS analyses the text and returns disambiguated URIs for the found entities. Then the Ontos Feeder annotates the returned entities in the original text within the CMS and highlights them in the WYSIWYG editor (see Figure 1). In a **context information area** of the CMS an overview of the found entities is given in the form of thumbnails (see Figure 2). Now the author has several choices:

- View additional information about the entities and navigate recursively.
- Adapt the filter (e.g. by type) in the config options and remove some of the entities.
- Revise the text and resend the text to the OWS.
- Accept all annotated entities and publish them along with the text.

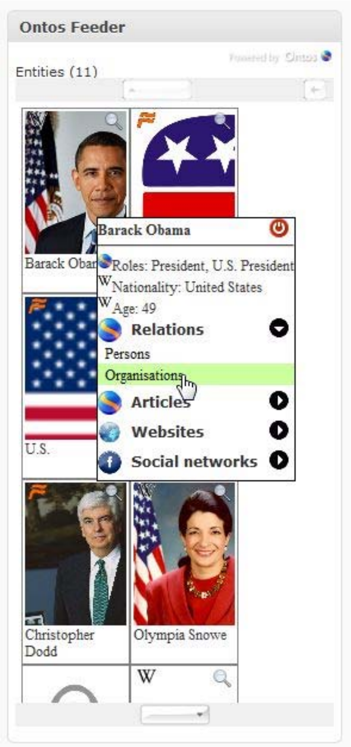
[2] `http://sourceforge.net/projects/ontosfeeder`

[3] `http://wordpress.org/extend/plugins/ontos-feeder/`

[4] `http://ivan-herman.name/2009/12/12/rdfa-usage-spreading.../` and `http://www.mahalo.com/rdfa`

If an author requires additional information about a particular entity, pointing at each annotation or thumbnail results in showing an appropriate pop-up menu with further contextual information. Each entity type provides different context information depending on their source; some of them are gathered from LOD providers such as DBpedia or Freebase, and some are coming directly from the OWS itself. While the LOD providers are used to retrieve entity attributes like age, nationality or area, the OWS provides information from the Ontos Knowledge Base comprised of information about the relationships to other entities (persons or organisations), related news articles as well as a summarizing entity report. Clicking on the related persons or organisations link in the popup menu refreshes the context information area with the thumbnails of that entities, so that the author can navigate recursively through all the relationships.

The *Ontos Knowledge Base* contains aggregated information from various sources. The URIs assigned to the extracted entities by the Web Service are *Ontos Entity Identifiers* (OEI). OEIs are de-referencable identifiers and are connected via `owl:sameAs` links to other Linked Data entities. Therefore, additional information from other Linked Data providers such as *DBpedia* and *Freebase* is presented in the entity context as well.



Roles: President, U.S. President
Nationality: United States
Age: 49

Fig. 2. The context information area is displayed next to the WYSIWYG editor and allows to navigate recursively to relevant contextual information from the Data Web

3 Architecture

While the server side consists of the OWS, the client side consists of the Core system and the CMS - adapters (see Figure 3). The core is CMS independent and can be embedded into a specific CMS by an appropriate adapter. Currently adapters for Drupal and WordPress are available.

Ontos Web Service (OWS) The Core system sends queries to the OWS. The Ontos Knowledge Base contains aggregated and refined information from around 1 million documents mainly from English online news. The Ontos Semantic Engine (NLP) extracts entities and their relationships from the text and disambiguates entities based on significance[1]. The significance is a complex measure based on the position of the occurrence in the text, the overall number of occurrences and

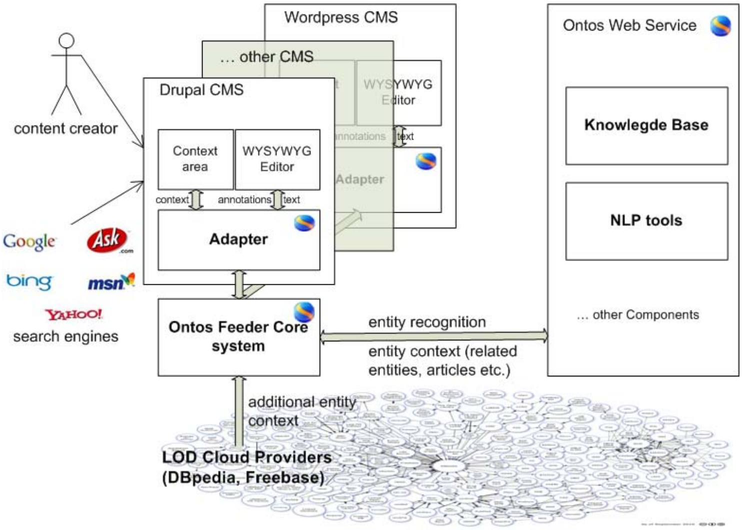


Fig. 3. Ontos Feeder overall architecture

the number of connected events and facts extracted from the text. The resulting information is returned to the Ontos Feeder.

Ontos Feeder. The Ontos Feeder currently supports requesting information for persons, organisations, locations and products, but can generally be extended to handle any of the entity types supported by the OWS. The user can configure, which types of entities the OWS should try to recognize in the provided text. The retrieval of each single piece of contextual information is encapsulated as a separate task by Ontos Feeder to increase the flexibility. The task engine supports task chaining, so if information could not be retrieved from a particular Linked Data source, it is requested from another one. The type of presented contextual information depends on the type of the recognized entity. The contextual information of a *Person* for example can consist of the age, nationality, status roles, connections to other persons and organisations, latest articles about this person, a Wikipedia article, a New York Times article, the personal homepage and a collection of different public social network profiles from Twitter or Facebook. Information about connections to other people and organisations, the status roles and the relevant articles are collected from the OWS. As every single information piece is requested by its own task, the variety of the presented contextual information can easily be adapted to personal needs.

4 Embedding Metadata

The OWS is able to annotate plain text as well as markup data such as HTML documents. The result is returned as a `stand-off annotation`, either in the form of start and end positions for text or an XPath expression for XML markup. A specialized annotation algorithm is used to: 1. highlight the annotations in the source HTML document in the editors. and 2. insert the annotations *inline* (as e.g. RDFa) into the HTML source of the article. Because all of the supported CMS WYSIWYG editors (currently FCKEditor and TinyMCE[5]) are capable of returning the current article as plain text, Ontos Feeder utilizes the Web Service in plain-text mode. As each of the editors have a different API, a special abstraction layer is put in front of the annotation algorithm to make it editor-independent. Furthermore, to make the annotation algorithm work faster for a plain-text document, all annotations are sorted in descended order and inserted bottom-up into the text. This avoids the recalculation of the annotation positions as compared to the top-down insertion. The annotation algorithm is capable of dealing with the entire supported semantic markup languages (RDFa and Microformats) and allows for annotation highlighting and on-the-fly binding of the contextual pop-up menu (see Figure 1).

5 Related Work and Summary

In recent years, several services have been published for suggesting annotations of tags to users. Among those services, OpenCalais and Zemanta are highly related to the Ontos Feeder as they also provide CMS integrations[6]. While Zemanta focuses on provide tag and link suggestions only, OpenCalais additionally extracts facts from the written text of the article. In contrast, the focus of the OWS is to provide disambiguated additional information, which is useful for the author. The data comes from the Ontos Knowledge Base and has been aggregated and fused from several sources. Also, the contribution of the Ontos Feeder, go well beyond the provision of a mere wrapper of a data service as it has a flexible, extensible architecture, is open-source and provides a context information area with recursive Linked Data navigation that aids the author. It transform stand-off annotations into inline RDFa and thus allows for a more fine-grained annotation method. Future work will be devoted to the area of co-referencing[2] for example by using OKKAM. Furthermore, it is planned, that users are able to define own vocabularies for named entity recognition, thus personalizing the annotation process.

References

1. Efimenko, I., Minor, S., Starostin, A., Drobyazko, G., Khoroshevsky, V.: Providing Semantic Content for the Next Generation Web. In: Semantic Web, pp. 39–62. InTech (2010)
2. Glaser, H., Jaffri, A., Millard, I.: Managing co-reference on the semantic web. In: WWW 2009 Workshop: Linked Data on the Web, LDOW 2009 (April 2009)

[5] `http://ckeditor.com/` and `http://tinymce.moxiecode.com/`

[6] `http://drupal.org/project/[opencalais|zemanta]`