Epiphany: Adaptable RDFa Generation Linking the Web of Documents to the Web of Data

Benjamin Adrian[1], Jörn Hees[2], Ivan Herman[3],
Michael Sintek[1], and Andreas Dengel[1,2]

[1] Knowledge Management Department, DFKI GmbH, Kaiserslautern, Germany
[2] CS Department, University of Kaiserslautern, Kaiserslautern, Germany
[3] Centre for Mathematics and Computer Sciences (CWI), Amsterdam, The Netherlands
benjamin.adrian@dfki.de, j_hees@cs.uni-kl.de, ivan.herman@cwi.nl
michael.sintek@dfki.de, andreas.dengel@dfki.de

Abstract. The appearance of Linked Open Data (LOD) was an important milestone for reaching a Web of Data. More and more RDF data sets get published to be consumed and integrated into a variety of applications. Pointing out one application, Linked Data can be used to enrich web pages with semantic annotations. This gives readers the chance to recall Semantic Web's knowledge about text passages. RDFa provides a well-defined base, as it extends HTML tags in web pages to a form that contains RDF data. Nevertheless, asking web authors to manually annotate their web pages with semantic annotations is illusive. We present Epiphany, a service that annotates Linked Data to web pages automatically by creating RDFa enhanced versions of the input HTML pages. In Epiphany, Linked Data can be any RDF dataset or mashup (e.g., DBpedia, BBC programs, etc.). Based on ontology-based information extraction and the dataset, Epiphany generates an RDF graph about a web page's content. Based on this RDF graph, RDFa annotations are generated and integrated in an RDFa enhanced version of the web page. Authors can use Epiphany to get RDFa enhanced versions of their articles that link to Linked Data models. Readers may use Epiphany to receive RDFa enhanced versions of web pages while surfing. We prove the quality of Epiphany in an evaluation based on Linked Data from BBC about music biographies and compare results of Epiphany and Open Calais.

1 Introduction

Motivated by the Linked Open Data (LOD) Initiative [1] more and more domain-specific Linked Data gets published in RDF format into the growing LOD cloud,[4] which is the emerging Web of Data. Following the Semantic Web idea, it is necessary not only to create links across different data sets, but also to link text

[4] http://richard.cyganiak.de/2007/10/lod/

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:dbpedia="http://dbpedia.org/resource/" ...>
<head>...
<link rel="meta" type="application/rdf+xml"
              href="epiphany/rdf?url=http://www.dfki.de"
              title="EPIPHANY's RDF">
</head><body>...
<span about="dbpedia:DFKI" property="rdfs:label">DFKI</span>
```

Listing 1.1. Excerpt of a web page, enriched by Epiphany. It contains a link to relevant RDF resources and RDFa annotations about their occurrences in the text. For example, it annotates the term 'DFKI' as `rdfs:label` and links it to the DBpedia HTTP URI `dbpedia:DFKI`.

sequences of web pages to existing LOD resources. Technically, the HTML extension RDFa [2] provides functionalities to allow web authors annotating their content with semantic markup and thus link their unstructured text into the world of machine understandable data. In addition to Microformats [3], which is another semantic markup language, RDFa is not constrained to tag text with properties such as names or phone numbers, but also allows linking these properties to existing real world instances of LOD data sets via HTTP URIs. Both, RDFa and Microformats, gain tool support from browser extensions such as Operator,[5] Semantic Radar,[6] or Ozone Browser [4]. Web authors[7] and web developers (see Drupal plug-in [5]) get more and more excited about the possibility to enrich their static or dynamic web sites with semantic markup. Even Google's [6] and Yahoo's [7] web crawlers start analyzing semantic markup in web sites. However, creating these annotations with RDFa (in style of Listing 1.1) or Microformats manually is cumbersome. Furthermore, manually created RDFa annotations are static. Thus they might not represent those properties and instance references the reader is currently interested in.

We present Epiphany,[8] a service that automatically generates RDFa annotations. Epiphany uses Linked Data as input to annotate HTML content with those properties and reference to those LOD resources [8] the user or group is currently interested in. Epiphany generates RDFa as shown in Listing 1.1. The service provides the following functionalities:

- Epiphany is adaptable and can be configured with any existing Linked Data model. Currently it is configured with data from DBpedia and BBC.
- Authors can generate RDFa annotations for their dynamic web pages.

[5] `http://www.kaply.com/weblog/operator`
[6] `http://www.sioc-project.org/firefox`
[7] e.g., Ivan Herman's homepage `http://www.ivan-herman.net`
[8] please lookup Epiphany at `http://projects.dfki.uni-kl.de/epiphany/`

Fig. 1. Screenshot displaying Epiphany generated RDFa annotations.

- Readers can generate RDFa annotations on demand for existing web pages (see screenshot in Fig. 1). These annotations are visualized with lighting boxes that provide additional background information about the resource (e.g., in case of dbpedia:DFKI, listing the abstract, the company logo, web page, etc.) they refer to. Readers also obtain links to common Linked Data Browsers, i.e., Tabulator, Marbles, Zitgist (see screenshot in Fig. 2).
- Web crawlers can be extended to generate Epiphany's RDFa annotations for crawled web pages.

In the following, we start with discussing related work. Afterwards, Epiphany's functionalities, visualizations, user interactions and provenance aspects are explained. The ontology-based information extraction facilities for generating RDF are outlined. An evaluation based on data from BBC music artist biographies confirms the quality of Epiphany. We show that Epiphany's results are comparable to those of Open Calais on the same data set. In addition to Open Calais that is specialized on the news domain, Epiphany may be configured with any domain that is published as Linked Data. After discussing evaluation results, and summarizing Epiphany's functionalities, we present future activities.

2 Related Work

Even before Linked Open Data, annotation systems like S-Cream [9] annotated web pages with instances or datatype properties from domain ontologies, semi-automatically. S-Cream did not provide its annotations in machine-readable format, but highlighted annotations to users or stored annotations back into a do-

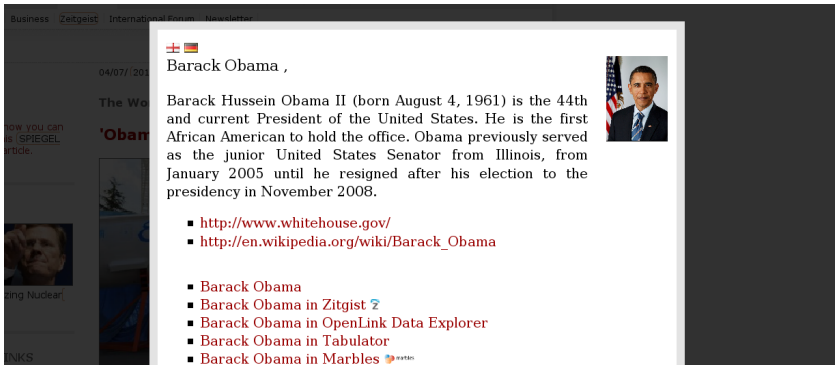


Fig. 2. Screenshot of Epiphany's lighting box for a single RDFa annotation.

main ontology. S-Cream and Epiphany use different kinds of information extraction (IE) techniques. Epiphany uses the ontology-based information extraction facilities that can be trained on any RDF domain model. S-Cream uses Amilcare, a traditional IE system without any ontology support. In consequence, S-Cream had to map non-ontological results (e.g., entities) from Amilcare to properties, classes, and instances of the domain ontology. Epiphany's incorporation of RDF domain knowledge into the IE process provides advantages, i.e., disambiguating possible instance candidates with similar labels, using SPARQL for specifying which entities to extract, or extracting new facts as RDF triples [10].

The Firefox plug-in Piggy Bank allows IE from web sites by screen scrapers. Results are stored in a local or global RDF store [11]. A screen scraper is a piece of Javascript code that extracts RDF information from within a web page's content. Similar approaches are GRDDL [12] and Monkeyformats.[9] GRDDL allows users to add references to XSLT scripts to web page headers that transform XML data on that page into RDF. Monkeyformats are userscripts for the Firefox plugin Greasemonkey [13].[10] These Javascripts search for patterns of DOM elements inside certain websites for adding Microformats into the DOM Tree.

Open Calais[11] services provide named entity recognition (NER, e.g., *Angela Merkel* as a person's name), instance recognition (e.g., *Angela Merkel* as a *person* with an HTTP URI) and facts with a couple of predefined properties (e.g., *Angela Merkel* is *chancellor*) with focus on News content. Open Calais is ontology-based, returns extraction results in RDF, and maintains Linked Data covering common sense instances (cities, countries, persons, companies, etc.). The coverage of instances that possess links to other Linked Data sets is very small. We could not find any cross links for recognized persons or music groups.[12]

[9] `http://monkeyformats.org`
[10] `http://www.greasespot.net`
[11] `http://www.opencalais.com`
[12] An online discussion about Calais' linking coverage: `http://www.opencalais.com/forums/known-issues/linked-data-how-much-linking`

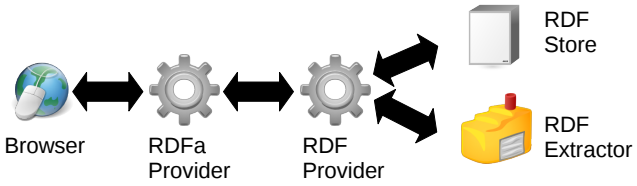


Fig. 3. Epiphany's RDFa generation process

The Gnosis Firefox plugin[13] performs NER about web pages, highlights results in text, and also lists entities grouped by types (e.g., person, city) in a sidebar. Gnosis renders tooltips while hovering over highlighted text passages with the mouse cursor that contain links to search the highlighted text passages in Wikipedia, Google, or the Reuters database. Gnosis does not perform instance recognition nor does it return data in RDF or Microformats.

Zemanta [14] is a web service for building web mashups. It finds relevant web links or images about blog entries. Zemanta also spots for labels of DBpedia[14] or Freebase[15] resources in web pages. The API can return results in RDF format.

Compared to these systems, Epiphany's characteristic features are *adaptivity* by changing Linked Data models used for annotating, *machine-readability*, as Epiphany annotates web pages with RDFa, and finally *usability* as Epiphany renders visualizations that link text with RDF resources from Linked Data.

3 The Epiphany Approach

Epiphany is a web service[16] that recognizes relevant instances and properties of a Linked Data model in web pages. It returns a version of the web page that contains RDFa annotations about these properties and instances, and a link to an RDF graph that summarizes these. We provide an overview about Epiphany's annotation process, provenance aspects, its data interface, and visualizations.

3.1 RDFa Generation

Figure 3 shows an overview of Epiphany's annotation process. Epiphany ties together Linked Data models and the content of web pages. It depends on Linked Data [8] to ensure that the user is able to request more information about an RDFa annotated text phrase via HTTP URIs. Figures 1 and 2 show an example where DBpedia is taken as domain model. Based on ontology-based information extraction methods, Epiphany extracts an RDF graph (called scenario graph) that consists of recognized instances with datatype property values that match

[13] `http://www.opencalais.com/Gnosis`
[14] `http://dbpedia.org`
[15] `http://www.freebase.com`
[16] `http://projects.dfki.uni-kl.de/epiphany/`

with text content, and known object property values between these instances (see Sect. 4 for details). The scenario graph is stored in an RDF store as Named Graph. Epiphany's RDFa Provider (see Fig. 3) parses a web page and compares datatype property values of the scenario graph with the page's text nodes. It returns a transformed version of the web page that contains positive matches for semantic annotations in RDFa. The following transformations are done by the RDFa Provider:

- The HTML or XHTML document type definition of the original web page is replaced with W3C's XHTML+RDFa document type definition.
- The HTML header of the original web page is extended with the URI of the scenario graph as meta information (see Listing 1.1). If RDF is generated from the RDFa inside the website, the `<link rel="meta"...>` statement adds an extra triple referring to the scenario graph. This reinforces the Linked Data aspect of the whole process: Users can find extra information, not necessarily present on the page itself, by consulting that scenario graph.
- Inside the page's body, each match between scenario graph and text content creates an RDFa annotation, i.e., HTML `span` elements (see Listing 1.1).
- Epiphany adds CSS information to the RDFa enhanced web page that highlights RDFa content with colored borders (see screenshot in Fig. 1).
- In addition, added Javascript functions render a lighting box (see screenshot in Fig. 2) when clicking on RDFa content with the mouse cursor. This lighting box contains configurable text and image information about the annotated instance taken from the domain model published as Linked Data.

Epiphany's RDF Provider manages persistence, access, and creation of RDF scenario graphs about web pages. Each scenario graph is stored as a named graph in an RDF store (an OpenLink Virtuoso Server). Accessing scenario graphs is done in Linked Data style, as every graph is identified by an HTTP URI that leads to the RDF document.[17]

3.2 Provenance

The RDF Provider enriches extracted scenario graphs with additional meta information. These are used to determine whether an existing scenario graph about a dynamic web is still up-to-date with respect to page changes or different Linked Data models in Epiphany. In addition, meta data contains optional information about the user or group who triggered the creation of the scenario graph. The Vocabulary of Interlinked Datasets (VOID [15]) is used to describe the version of Epiphany's underlying Linked Data model. The Dublin Core Metadata Element Set (DC [16]) is used to describe the web page the scenario graph is about (dc:subject), the last modified date of the web page (dc:modified), creation date of scenario graph (dc:created), and user or group identifiers (dc:audience). The score property defined by Open Calais[18] is used to describe the minimum confidence value an extracted instance or fact has inside a scenario graph.

[17] please refer to `http://projects.dfki.uni-kl.de/epiphany/db`
[18] `http://s.opencalais.com/1/pred/score`

```
PREFIX dc: <http://purl.org/dc/terms/>
PREFIX oc: <http://s.opencalais.com/1/pred/>
ASK { GRAPH ?g {
  ?s dc:subject PAGE_URI;
     dc:audience USER_URI;
     dc:created ?creation;
     oc:score ?confidence.
  PAGE_URI dc:modified ?modified.
FILTER (
  xsd:float(?confidence) >= xsd:float(THRESHOLD) &&
  xsd:integer(?modified) >= xsd:integer(CURRENT_TIME_STAMP))
}}
```

Listing 1.2. Epiphany's SPARQL ASK query pattern querying the RDF store for an existing scenario graph with given provenance information. Variable names written in capitals are configurable or dynamically replaced.

Based on this provenance information, by executing the SPARQL ASK query in Listing 1.2, the RDF Provider can decide if a scenario graph exists inside the RDF store. If no graph exists, Epiphany creates a new one.

3.3 Epiphany's Data Interfaces

Epiphany provides four data interfaces to create RDFa annotations:

1. Web authors can use a web form to generate RDFa for text snippets. These RDFa annotated text snippets can be used as static content in web pages. Scenario graphs about text snippets are not persisted in the RDF store.
2. Web surfers can configure their browsers to use an HTTP-Proxy to call the Epiphany service for web pages. Modern browsers allow the setup of proxies with white- or blacklists of Internet domain names to control proxy requests. Using proxies ensures preserving the original URL of the web page.
3. Web surfers can also use a bookmarklet, which allows to encapsulate arbitrary Javascript code into a bookmark. At will, the users can click on the bookmarklet, which can then
   - redirect to an Epiphany URL quoting the current web page
   - directly replace parts of the page's DOM with RDFa annotated content from Epiphany. This approach also preserves the original URL, but requires the browser to interpret the parameter `AccessControl-AllowOrigin *` in HTTP response headers[19] in order to allow cross site scripting for this domain.

   The bookmarklets are implemented by Epiphany's RESTful API.[20] To enhance usability even more, the Firefox plugin WebSmartyPants is provided and can be downloaded under Epiphany's website.

[19] See W3C working draft at `http://www.w3.org/TR/access-control`
[20] See the API description at `http://projects.dfki.uni-kl.de/epiphany/api`

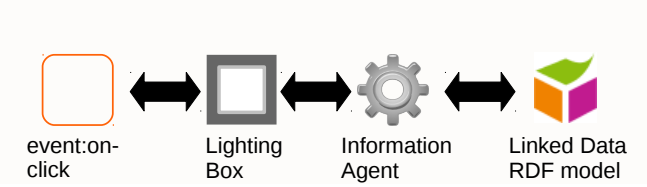


Fig. 4. Epiphany's lighting box rendering process

4. As soon as the W3C RDFa working group publishes an RDFa DOM API[21] in a definite form, it is planned to provide a conforming Epiphany Javascript API.

3.4 Epiphany's RDFa Visualizations

Without any browser plugin support, existing RDFa content in web pages remains hidden to users. Existing RDFa visualizations, such as Ozone Browser [4], or W3C's RDFa Bookmarklets[22] visualize information rather technically. In Epiphany, lighting boxes are used to visualize additional information about annotated text passages (see screenshot in Figure 2).

According to Figure 4, the Javascript event onmouseclick on an RDFa span leads to an AJAX request to the Information Agent, passing the subject's URI of the RDFa span. The Information Agent requests the RDF graph of the given HTTP URI, parses it, and then filters RDF triples for specified properties. These properties can be grouped by template categories listed in a configuration file (see Table 1). The lighting box is a simple HTML template with slots that correspond to existing template categories. These slots can be designed by CSS documents that define CSS classes with the category as name.

Template Category	**RDF Property List**
label	`foaf:name`, `rdfs:label`
image	`foaf:depiction`, `dbpedia:thumbnail`
description	`rdfs:comment`, `dbprop:abstract`
reference	`foaf:homepage`, `foaf:page`

Table 1. Categories with RDF properties used to populate the lighting box in Figure 2

4 Ontology-based Information Extraction

Epiphany's generated RDFa annotations are based on RDF data, which is generated by ontology-based information extraction (OBIE) methods [17]. Epiphany's

[21] See agenda at `http://www.w3.org/2010/02/rdfa/`
[22] `http://www.w3.org/2006/07/SWD/RDFa/impl/js/`

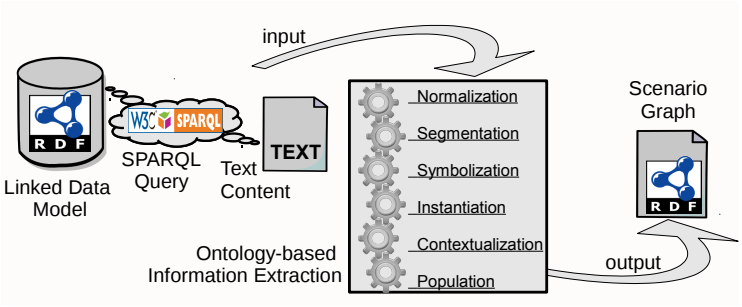


Fig. 5. Usage scenario of Epiphany's OBIE system: Based on an RDF domain model, a user asks a SPARQL query about a text document. Taking domain model, text, and query as input, Epiphany's extraction pipeline creates an RDF scenario graph.

OBIE facility incorporates domain-specific RDF data into the IE pipeline [10] (see Fig. 5) and returns extracted results in RDF format by reusing the RDFS schema of the input data. The IE pipeline is designed to supports optional SPARQL queries as input which specify the types of entities and relations to extract from text. By changing the domain model, the user is allowed to "ask" different queries covering other domains and receive different IE results without any reimplementation or rule engineering efforts.

The following system description summarizes ontology-based information extraction tasks used in Epiphany. More detailed information are given in [10, 17].

4.1 Preprocessing the RDF Domain Model

In a preprocessing step Epiphany analyzes the input RDF model, i.e., existing instances and classes, datatype property values (e.g., `foaf:name`) and object property values (e.g., `foaf:knows`). Datatype property values are converted to efficient data structures (e.g., B*-Trees, Suffix Arrays) for pattern matching on character strings. Properties are represented in adjacency lists and stored in bitmaps.

4.2 Extraction Pipeline

The RDF model preprocessor returns a so-called extraction session. Based on this session, Epiphany's OBIE pipeline is ready to extract model-specific information from text. This comprises six major process steps (see Fig. 5) covering necessary IE tasks. Each task generates a set of hypotheses weighted with confidence values that are combined by using Dempster-Shafer's belief function [18].

Normalization transforms a document into a textual representation. Here, plain text content and existing metadata (e.g., title, author) are extracted based on the Aperture framework.[23]

[23] `http://aperture.sourceforge.net`

Segmentation partitions the plain text content into units of tokens and sentences. The implementation token and sentence detection is based based on regular expressions. In steps of sentences, each token is classified by a *POS tagger*.[24] Noun phrases (that are sequences of tokens) are detected by a Noun phrase chunker that is implemented as conditional random field. These noun phrases are stored and finally sorted in a suffix array.

Symbolization recognizes datatype property values in text. It matches the noun phrases in text that are stored inside the suffix array and sorted values of datatype properties inside the domain model. (e.g., assuming the existence of the triple (`:_ foaf:label 'DFKI'.`), in text: *DFKI was founded in 1988*, 'DFKI' is recognized as content symbol of type `foaf:label`).

Instantiation resolves instances of the domain-specific data model for each recognized datatype property value (e.g., assuming the existence of the triple (`dbpedia:DFKI foaf:label 'DFKI'.`) and text snippet: *DFKI was founded in 1988*, '*DFKI*' is resolved as `foaf:label` of instance `dbpedia:DFKI`). An instance candidate recognition resolves possible candidates for recognized datatype property values. Here, ambiguities may occur if more than one instance possesses the same datatype property values (e.g., first names of *Helmut Kohl* and *Helmut Schmidt*). Candidates are disambiguated by counting resolved instances in the domain model that are related directly with an object property[25] or indirectly via another instance of the domain model.[26] As result, the ambiguous instance with a higher count of related and recognized instances is taken.

Contextualization extracts facts (RDF triples) about resolved instances. At first, a fact candidate extraction computes all possible facts between resolved instances. Then, a set of fact selectors rates these facts according to heuristics. Currently Epiphany contains a known fact selector and a spreading activation based fact selector. The known fact selector heightens rates of extracted facts that exist as triples inside the domain model.

For a given SPARQL query, the **Population** creates scenario graphs in RDF format. They contain extracted values, i.e., HTTP URIs of resolved instances with those datatype property values that match with text sequences and RDF triples about object properties between these resolved instances. Scenario graphs can be filtered and ordered by confidence values in range between zero and one.

4.3 Usage in Epiphany

Currently, Epiphany uses a configuration of the OBIE pipeline which focuses on text annotation. It covers text extraction, tokenization, content symbol recognition, instance recognition and disambiguation, fact extraction and known fact selection, and finally the population of scenario graphs. Epiphany uses the generic SPARQL query as template for scenario graphs: `SELECT * WHERE {?s ?p ?o}`.

[24] `http://opennlp.sourceforge.net`

[25] e.g., `dbpedia:Helmut_Kohl rdf:type dbpedia:Chancellor`

[26] e.g., `dbpedia:Helmut_Kohl dbprop:politicalParty dbpedia:CDU` and `dbpedia:Angela_Merkel dbprop:politicalParty dbpedia:CDU`

Facet	Cardinalities
web pages	12,462
words	5,530,477
`mo:MusicGroup`	12,462
`mo:SoloMusicArtist`	31,429
`<> foaf:name <>.`	36,397
`<> mo:member <>.`	32,104

Music group name	Frequency
Off	3,991
Free	5,715
Contact	12,461
Fin	12,461
Food	12,461
Sport	12,461

Table 2. (a): Cardinality statistics of BBC corpus values, (b): Frequent music group names extracted by Epiphany

For future work, it is planned to let Epiphany even recommend domain specific new instances for given Linked Data.

5 Evaluation

The evaluation proved that the quality of Epiphany's extraction results (and finally of the generated RDFa annotations) is comparable to results from Open Calais. An advantage of Epiphany is adaptability. It is not tied to the News domain like Open Calais. Epiphany may be initialized with any information domain that is described as Linked Data.

We decided to evaluate Epiphany by analyzing the quality of extracted scenario graphs, as these graphs form the base of the generated RDFa annotations. Furthermore, we compared RDF graphs generated by Epiphany with those generated by Open Calais.

5.1 Experimental Setup

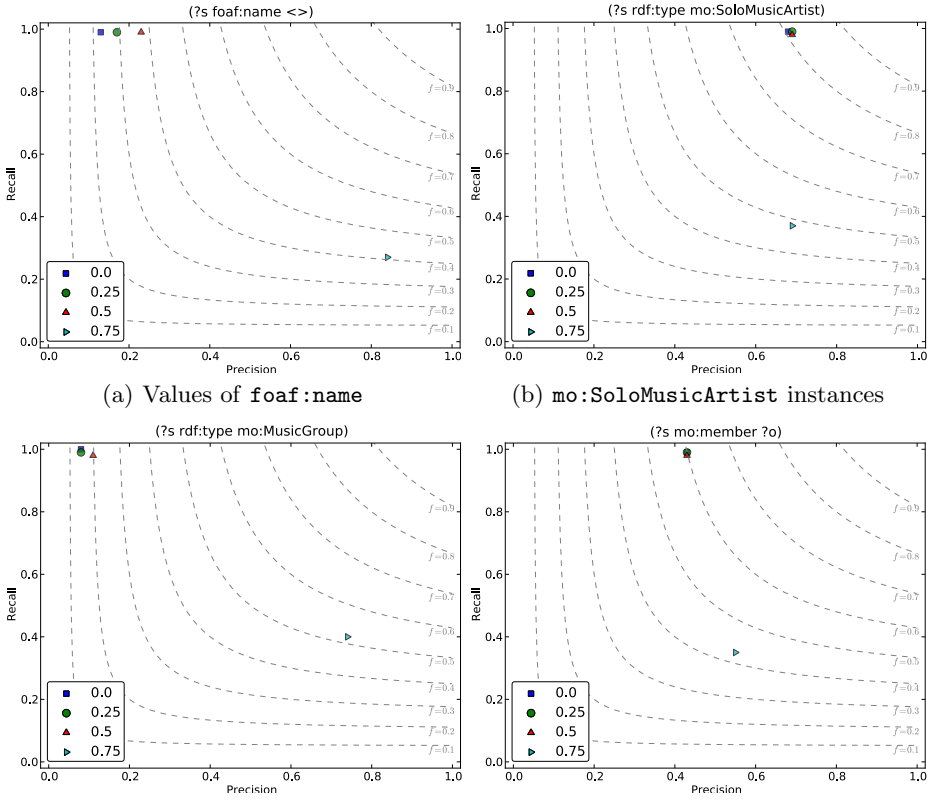
Three essential things were identified for evaluating Epiphany as domain-adaptive and ontology-based information extraction system:

1. A document corpus is needed. The content of each document should cover a single domain and refer to multiple instances and facts.
2. These instances and facts should exist as gold standard for each document. Ideally, RDF graphs exist for each document, that formalize its content.
3. This RDF data should be formalized clearly by using a set of ontologies. Ideally, these ontologies should be commonly used in Linked Data.

As data basis, we used web pages from `bbc.co.uk/music`[27] describing biographies about music groups. For each biography on a web page, BBC provides metadata in form of a Linked Data model.[28] The ontologies FOAF, Music Ontology (MO), and Dublin Core are used to describe music groups and their

[27] `http://www.bbc.co.uk/music/developers`

[28] e.g., BBC's Linked Data graph about the `mo:MusicGroup` Queen: `http://www.bbc.co.uk/music/artists/0383dadf-2a4e-4d10-a46a-e9e041da8eb3.rdf`



(a) Values of `foaf:name` (b) `mo:SoloMusicArtist` instances

(c) `mo:MusicGroup` instances (d) `mo:member` relations.

Fig. 6. Diagrams about Epiphany's extraction results. Four measured values define confidence threshold of scenario graphs about extracted instances (a,b,c) or facts(d).

members. The RDF graphs were used as baseline. Extracted RDF graphs from Epiphany for a given web page are compared against corresponding metadata by BBC.

HTTP URIs of music group members refer to additional Linked Data. We collected all RDF graphs about music groups that could be found by querying BBC's backstage SPARQL endpoint[29] and added the RDF graphs of all group members. The resulting mashup was used as domain-specific Linked Data input for Epiphany. Table 2(a) lists statistics about the amount of documents and tokens inside the test corpus. It also lists the count of properties about music groups and their solo music artist members inside the mashup.

We evaluated the quality of the following extraction results: (Fig. 6.a) all extracted instances with `foaf:name` values, (Fig. 6.b+c) just extracted instances with `foaf:name` values of type `mo:MusicGroup` and `mo:SoloMusicArtist`, (Fig. 6.d) `mo:member` relationships between `mo:MusicGroups` and

[29] `http://api.talis.com/stores/bbc-backstage`

mo:SoloMusicArtists. Therefore we checked, if certain RDF triples (Fig. 6.a+d) or RDF molecules (Fig. 6.b+c) inside baseline RDF graphs were extracted and thus exist in Epiphany's scenario graphs.

5.2 Comparing Epiphany's Scenario Graph with BBC's Baseline

Figure 6 describes evaluation results for each extracted instance or fact. Four measure points (≥ 0.75, ≥ 0.5, ≥ 0.25, and ≥ 0.0.) represent scenario graphs about all 12,462 web pages. The names define a confidence threshold of extraction results. Measure points are rated by precision and recall. Curves inside diagrams represent layers of harmonic F-measure ratios. Three points show that Epiphany extracts instances and facts with recall ratios above 96.0% for thresholds up to ≥ 0.5. Precision values except for extracted mo:SoloMusicArtist instances stay below 35%. Extracted instances of mo:SoloMusicArtist gained precision values above 65%. In general, an increase of threshold up to ≥ 0.75 leads to precision values higher than 50%. The distribution of precision can be explained by some foaf:name values of mo:MusicGroups (see Table 2(b)) which occur in nearly all web pages in a different language context.

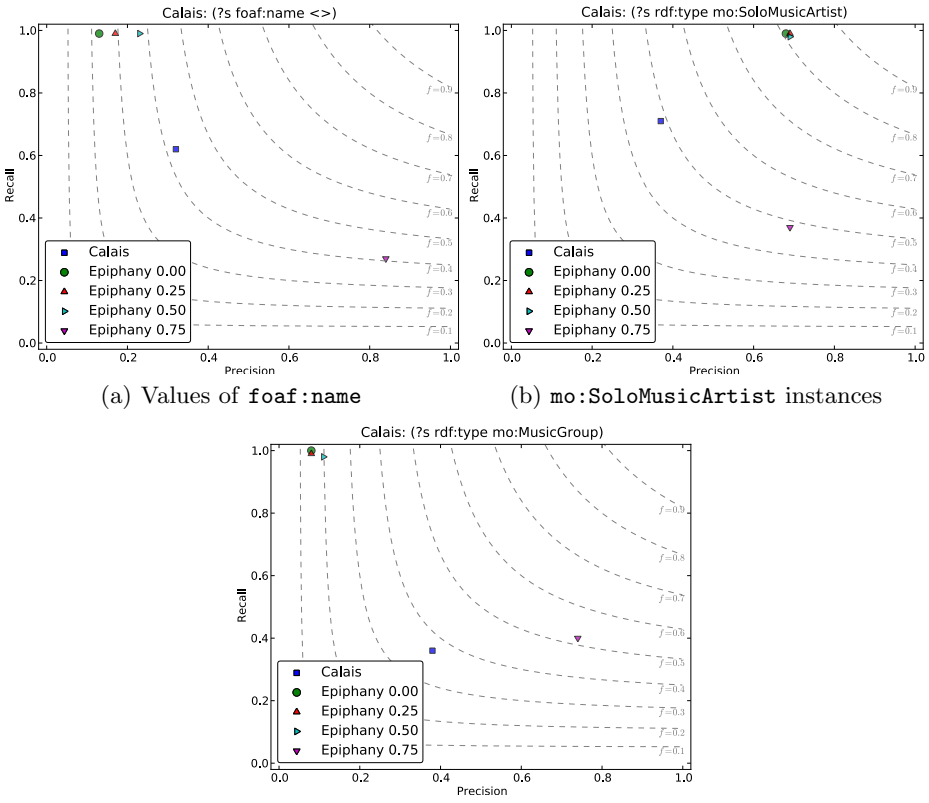
5.3 Comparing Results from Open Calais and Epiphany

We compared results obtained from Open Calais and Epiphany about the same data set. Open Calais is not domain-specific, thus extracted more types of instances than we needed. It also uses its own RDFS vocabulary[30] to represent RDF results. So, we had to filter results, transformed the classes oc:Person and oc:MusicGroup to mo:SoloMusicArtist and mo:MusicGroup, and transformed the properties oc:name and oc:match to foaf:name. This facilitated comparing Calais' RDF to BBC's baseline. Calais could not extract group member relationships. The diagrams in Figure 7 are structured as Figure 6, but also contain results of Open Calais. For instances with foaf:name values and those of type mo:MusicGroup, Open Calais' results gained higher precision values compared to Epiphany's measure points with thresholds below ≥ 0.75. In general, Epiphany's results were rated with higher recall values. Epiphany reached better precision values for measure points ≥ 0.75.

5.4 Result Discussion

Comparing results of Epiphany and Open Calais shows, that Epiphany is able to annotate existing instances and facts of the input Linked Data if the web page refers to these. Epiphany even achieved slightly better Recall results than Open Calais. One reason is that Epiphany's data base is much more related to the web pages content than the generic data base of Open Calais. Open Calais gained better precision values than Epiphany because Open Calais' domain model did not cover such a huge amount of music group names as they exist in BBC Programs.

[30] http://d.opencalais.com/1/type/



(a) Values of `foaf:name` (b) `mo:SoloMusicArtist` instances

(c) `mo:MusicGroup` instances

Fig. 7. Comparing results from Epiphany's OBIE component and Open Calais.

(Especially not the ambiguous band names listed in Table 2 b.) For dealing with ambiguous instance labels, we plan to look for a contextual analysis that re-ranks extraction results based on how they are interrelated inside the domain model. Please consider that compared to Open Calais, Epiphany is domain adaptable and supports more than just one domain model.

6 Summary and Outlook

We described Epiphany, a web service that annotates web pages with RDFa which is linked to a Linked Data model (e.g., DBpedia , BBC programs, etc.). The service is published at `http://projects.dfki.uni-kl.de/epiphany/` and provides Bookmarklets, an HTTP proxy server, a RESTful API, and the Firefox plugin WebSmartyPants. Currently the service provides Linked data from DBpedia and BBC programs for being annotated as RDFa to web pages. The evaluation confirmed that the coverage of extracted instances from web pages is comparable between Epiphany and Open Calais. Epiphany is adaptable and can

be configured to support different Linked Data models for annotating web pages with additional Linked Data content. Future work comprises a Javascript API and the use of Epiphany in web crawlers in order to provide semantic indexing about text without any semantic annotations.

Acknowledgements This work was financed in part by the BMBF project Perspecting (Grant 01IW08002).

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – the story so far. Int. Journal on Semantic Web and Information Systems (IJSWIS) (2009)
2. W3C: RDFa in XHTML: syntax and processing. W3C rec., W3C (2008)
3. Khare, R.: Microformats: The next (small) thing on the semantic web? IEEE Internet Computing **10**(1) (2006) 68–75
4. Burel, G., Cano1, A.E., Lanfranchi, V.: Ozone browser: Augmenting the web with semantic overlays. In: Proceedings of the Fifth Workshop on Scripting and Development for the Semantic Web SFSW '09. Volume 449 of CEUR Workshop Proceedings. (June 2009)
5. Corlosquet, S., Delbru, R., Clark, T., Polleres, A., Decker, S.: Produce and Consume Linked Data with Drupal! In: 8th International Semantic Web Conference (ISWC2009). (October 2009)
6. Google: About RDFa - Webmasters/Site owners Help. (November 2009)
7. Yahoo! Inc.: SearchMonkey Guide - A Manual for SearchMonkey Developers and Publishers. (2008)
8. Bizer, C., Cyganiak, R., Heath, T.: How to publish linked data on the web. Web page (2007) Revised 2008. Accessed 07/08/2009.
9. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM - Semi-automatic CREAtion of Metadata. In: Proc. of EKAW '02, London, UK, Springer-Verlag (2002) 358–372
10. Adrian, B.: Incorporating ontological background knowledge into information extraction. In Maynard, D., ed.: ISWC 2009 Doctoral Consortium. (October 2009)
11. Huynh, D., Mazzocchi, S., Karger, D.: Piggy bank: Experience the semantic web inside your web browser. Web Semantics **5**(1) (2007) 16–27
12. W3C: Gleaning resource descriptions from dialects of languages (GRDDL). W3C rec., W3C (September 2007)
13. Pilgrim, M.: Greasemonkey Hacks: Tips & Tools for Remixing the Web with Firefox (Hacks). O'Reilly Media, Inc. (2005)
14. Tori, A.: Zemanta service. Zemanta. (07 2008) `http://developer.zemanta.com/docs/Zemanta_API_companion`.
15. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: voiD Guide - Using the Vocabulary of Interlinked Datasets (2009) `http://rdfs.org/ns/void-guide`.
16. Dublin Core Metadata Initiative: DCMI Metadata Terms (2006) `http://dublincore.org/documents/dcmi-terms`.
17. Adrian, B., Hees, J., van Elst, L., Dengel, A.: iDocument: using ontologies for extracting and annotating information from unstructured text. In: KI 2009: Advances in Artificial Intelligence. Volume 5803 of LNAI., Springer (9 2009) 249–256
18. Adrian, B., Dengel, A.: Believing finite-state cascades in knowledge-based information extraction. In: KI. Volume 5243 of LNAI., Springer (9 2008) 152–159