Crowdsourced Semantic Annotation of Scientific Publications and Tabular Data in PDF[*]

Jaana Takis
Fraunhofer IAIS, Germany
jaana.takis
@iais.fraunhofer.de

AQM Saiful Islam
University of Bonn, Germany
saiful.nipo@gmail.com

Christoph Lange, Sören Auer
University of Bonn /
Fraunhofer IAIS, Germany
{langec|auer}@cs.uni-bonn.de

ABSTRACT

Significant amounts of knowledge in science and technology have so far not been published as Linked Open Data but are contained in the text and tables of legacy PDF publications. Making such information available as RDF would, for example, provide direct access to claims and facilitate surveys of related work. A lot of valuable tabular information that till now only existed in PDF documents would also finally become machine understandable. Instead of studying scientific literature or engineering patents for months, it would be possible to collect such input by simple SPARQL queries. The SemAnn approach enables collaborative annotation of text and tables in PDF documents, a format that is still the common denominator of publishing, thus maximising the potential user base. The resulting annotations in RDF format are available for querying through a SPARQL endpoint. To incentivise users with an immediate benefit for making the effort of annotation, SemAnn recommends related papers, taking into account the hierarchical context of annotations in a novel way. We evaluated the usability of SemAnn and the usefulness of its recommendations by analysing annotations resulting from tasks assigned to test users and by interviewing them. While the evaluation shows that even few annotations lead to a good recall, we also observed unexpected, serendipitous recommendations, which confirms the merit of our low-threshold annotation support for the crowd.

Categories and Subject Descriptors

I.7.1 [**Document and Text Editing**]: Document management; K.4.3 [**Organizational Impacts**]: Computer-supported collaborative work; H.3.5 [**Online Information Services**]: Web-based services

Keywords

RDF Data Cube, PDF, Semantic annotation, DBpedia

1. INTRODUCTION

Although the Internet and digital technologies have considerably improved the accessibility of research communication, the means how scientific knowledge is encoded, represented and shared have not significantly changed. Scientists still spend most of their time encoding the insights gained in texts (articles, books) and decoding the knowledge shared by their peers from texts. However, scientific knowledge exchange often involves structured information, such as experimental results, collected data, taxonomies or formulas. Data portals can be used to publish data underlying a certain publication. However, even the actual text of scientific publications often contains structured information currently hidden in prose. Examples include a) claims and supporting evidence for these, b) related approaches with their advantages and disadvantages, or c) a taxonomic classification of the approach described in a certain publication. Such information could easily be expressed and represented in a structured way in RDF; suitable vocabularies exist (cf. section 2). Once scientific publications are increasingly represented in a way that preserves the structure of information, related or similar information from different publications can easily be interlinked and integrated. A survey on a certain research area, for example, could then possibly be generated almost automatically, by collecting the taxonomic classifications as well as advantages and disadvantages of various approaches from different publications comprising structured information in addition to the human-readable text. As a result, scientific knowledge sharing would be improved substantially, since researchers and other stakeholders would be enabled to search and discover research results not only by using keyword search and following citations, but by formulating sophisticated queries such as 'List me all Named Entity Recognition approaches published in the last 5 years, together with the corresponding precision and recall they achieve on a certain benchmark corpus'. Currently, answering such a relatively simple question costs a researcher several weeks or even months of research. Especially for young researchers it is extremely difficult to navigate through the jungle of research.

Although a general solution for this problem is relatively straightforward to realise – researchers could simply publish some RDF Linked Data describing their research along with a paper – the main challenge is to create a network effect through an architecture of participation. This is required, since very few researchers would make the additional effort of creating a semantic description in addition to a paper if the benefit of doing so were not immediate.

In this work, we present an approach for facilitating the collaborative annotation of 'legacy' scientific publications. Authors or readers are empowered to annotate a PDF publication directly from within their browser. Annotations are represented in RDF reusing existing vocabularies and ontologies, while at the same time en-

[*]This work has been partially funded by the European Commission under H2020 grant agreement no. 645833.

SEMANTiCS '15, September 15 - 17, 2015, Vienna, Austria

ISBN 978-1-4503-3462-4/15/09…$15.00
DOI: http://dx.doi.org/10.1145/2814864.2814887

couraging annotators to extend and enrich the vocabularies with self-created domain concepts, which others can immediately reuse. Our implementation, the *SemAnn system*, enables collaborative annotation of PDF documents, still the common denominator of publishing, thus maximising the potential user base. To incentivise users with an immediate benefit for making the effort of annotation, SemAnn recommends related papers, taking into account this hierarchical context of annotations in a novel way.

We are consistent with the widely cited 'integrating' definition of 'crowdsourcing' [8] insofar as we consider a 'participative online activity', in which 'individuals of varying knowledge, heterogeneity, and number' volunteer, and as there is 'mutual benefit' as all users receive recommendations. We merely leave the role of the 'ordering party' implicit; this could be anyone who might benefit from collective knowledge about publications.

The article is structured as follows: section 2 discusses challenges and approaches related to improving scholarly communication. section 3 presents the architecture of our implementation. We detail the knowledge model employed for the annotation in section 4. In section 5 we describe the recommendation functionality. section 7 discusses our evaluation involving real users. section 8 concludes with an outlook to future work.

2. RELATED WORK

Software support for scholarly publishing is a hot topic, as, e.g., recent initiatives such as the *Semantic Publishing Challenge* [16] prove. Still, existing solutions have not sufficiently addressed the following problems – which provides the motivation for our work.

Lack of support for the PDF format. PDF enjoys little support from interactive semantic annotation tools as it is hard to automatically retrieve information from it. E.g. *Utopia Documents* (cf. http://utopiadocs.com and [1]) is capable of displaying semantic content but does not support the creation of semantic annotations. *GoNTogle* [2] does not support multiple ontologies and is only suitable for applying categorisation vocabularies like ACM. *PDFTab* [7] stores semantic data exclusively within the PDF itself, which makes its reuse difficult. For tabular data extraction the *CODE Data Extraction* [18] tool can be used for storing the tables as RDF Data Cube Vocabulary in a central repository. Unlike the SemAnn approach that requires the user to manually select the table, this tool employs unsupervised machine learning techniques for recognising tables. Semantic enrichment is done semi-automatically and similar in approach to that in SemAnn. However, this tool is used for creating cubes for storing in a central repository for visualisation purposes. As such it differs from our SemAnn approach where the tabular and non-tabular semantic annotations are kept within the same repository along with the reference to its physical origin within the document for extra querying capabilities. Another tool, *PDFTables*[1], extracts tables by shape from the PDF which can then be exported as CSV or XML. However, it treats the whole document as a table, including text in paragraphs and it is not as good as SemAnn at recognising multi-row cells.

Limited tool support for multiple vocabularies. There is a general lack of freely available simple annotation tools that are not of specialised use and that do not limit the user to an ontology specific to a limited domain. Scientific use cases often requires the use of *multiple* specific vocabularies, which is best supported by tools that support a wide range of ontologies. *DOMEO* [4], a web-based annotation framework for online HTML and XML documents is a good example of such an effort; it started with a focus on biomedicine. Its new version v.2 is yet to be released, so it is currently unclear how flexible it will be in its support of other ontologies due to currently ongoing major changes.

Support beyond typed annotations. Simple classification vocabularies are not sufficient for representing complex concepts or relationships between them. Examples of such use cases include citation links between papers, citation contexts (*CiTO*[2]), modelling arguments (*Argument Model Ontology*[3]).

Invasive editing in traditional authoring software supports the annotation of text at the time of writing, rather than subsequently annotating the published version of a document. It has, for example, been realised for mathematical and rhetorical knowledge structures, by semantic macro packages for LaTeX [15, 9], and by plugins for PowerPoint [14]. None of these solutions have been adopted widely; only the first system (*sTeX*) is still being maintained. With HTML5 advancing, lightweight invasive editing solutions have more recently been realised in web interfaces, which have been extended to enrich the HTML document being authored with RDFa annotations. Examples include the *RDFa Content Editor RDFaCE* [13] and the *One Click Annotator* [10]. Both are based on *TinyMCE*, an HTML editing component widely used in web content management systems. A similar JavaScript-based architecture could be adopted by a browser plugin for annotating read-only HTML documents published on the Web.

Still, PDF remains the most important format in which scientific works are published. Few scientific publications are available in other formats, and the wide variety of source formats and editors for them has so far prevented a wide take-up of invasive editing.

3. ARCHITECTURE

SemAnn[4] (online demo available at http://ok-semann.iais.fraunhofer.de) addresses the following high-level requirements:

Direct annotation of PDF files within a Web browser. Files can be opened from the user's desktop or loaded from a URL.

Semantic annotation of text. After selecting some text or table in PDF, semantic annotations can be created and saved to a triple store. For simple semantic enrichment the DBpedia Lookup[5] web service is used, additional vocabularies are currently available only for non-tabular data.

Selection from multiple vocabularies. A default annotation ontology is integrated, but users can also use others, and create new vocabulary terms.

Recommendations of similar papers. Similar papers are recommended by semantically comparing annotations of the currently loaded document to those of the other papers in the database. Recommendations are displayed with explanations.

Export tables as CSV. Selected tables can also be exported as CSV. Structured data like CSV is very useful for further custom uses i.e. there are a lot of CSV-To-RDF[6] converters available to transform tabular data into RDF data structure.

Figure 1 shows the overall architecture of SemAnn that comprises of the following components: the user interface, triple store

[1] https://pdftables.com

[2] http://purl.org/spar/cito

[3] www.essepuntato.it/2011/02/argumentmodel

[4] Open-source code: http://github.com/AKSW/semann

[5] https://github.com/dbpedia/lookup

[6] www.w3.org/wiki/ConverterToRdf

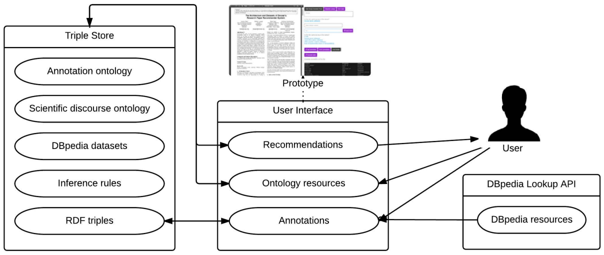


Figure 1: Components of the SemAnn approach.

and DBpedia Lookup API. The triple store stores user-created annotations, manages ontologies, hosts the SPARQL endpoint and enables retrieving recommendations of similar papers. SemAnn will work with SPARQL-compliant triple stores for this purpose (for development and testing we used OpenLink Virtuoso). The DBpedia Lookup API is used for semantically enriching user annotations but additional vocabularies can be activated from the UI.

SemAnn extends *PDF.js*[7], a JavaScript library for parsing and rendering PDF files in HTML5. PDF.js also comes with Firefox browser and has ca 100,000 daily users. This design choice eliminates platform dependence and compatibility issues caused by different PDF readers that users might have installed on their computers. This also means that end users need not install any software beyond a JavaScript-enabled browser.

4. KNOWLEDGE MODEL

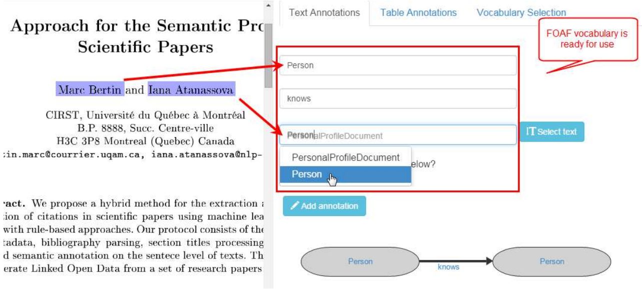
One of the design goals was to enable the end-users to have maximum freedom in the type of annotations they might want to create. That means we need support for both – simple annotations that are mere classifications *and* more complex annotations that describe relationships. We developed our ontology according to the following principles:

- flexibility in supporting various types of annotation.
- minimalistic *core* annotation ontology.
- usefulness of the RDF triples, e.g., when queried

We provide two default ontologies. Additional ontologies can be loaded by the end-user via the user interface which calls Virtuoso's Sponger service[8] for extracting the triples from the specified resource and making them available for subsequent use.

Figure 2 shows how the FOAF ontology can be applied after loading it. The user interface visualises RDF triples in a schematic form. Here, two authors are annotated as *foaf:Person*s, and a *foaf:knows* relationship is defined between them.

The *SemAnn Annotation Ontology* (SAO) is a lightweight ontology that was specifically developed to model information related to annotations. The design goal for SAO, depicted in Figure 3, was to represent information about annotations in a compact format. The emphasis on compactness was driven by the goal of keeping SPARQL queries simple, which is particularly relevant when making available a public SPARQL endpoint. We also considered the use of the *Annotation Ontology* (AO [5]); however, we deem AO too heavyweight, thus unnecessarily over-complicating annotations and queries. To express the same information that can be encoded in a single annotation in SAO, one needs five triples in AO.


Approach for the Semantic Pr
Scientific Papers

Marc Bertin and Iana Atanassova

CIRST, Université du Québec à Montréal
B.P. 8888, Succ. Centre-ville
H3C 3P8 Montreal (Quebec) Canada
:in.marc@courrier.uqam.ca, iana.atanassova@nlp-

act. We propose a hybrid method for the extraction
ion of citations in scientific papers using machine lea
with rule-based approaches. Our protocol consists of the
adata, bibliography parsing, section titles processing
d semantic annotation on the sentece level of texts. Th
erate Linked Open Data from a set of research papers

Person
knows
PersonalProfileDocument

Figure 2: Applying classes and properties from the loaded ontology.

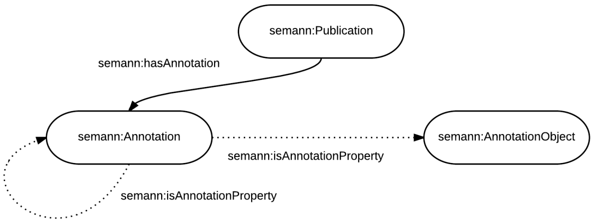


Figure 3: SemAnn Annotation Ontology.

An annotation can be an instance of multiple classes; this approach is used in the semantic enrichment of annotations and also simplifies SPARQL queries. For example, Figure 4 in section 4 depicts how the annotation 'Dynamic Languages' of type http://dbpedia.org/resource/Dynamic_Languages enriches its parent annotations of type *sro:Motivation* and *sro:Abstract*. Below are some more complex annotations supported by SemAnn:

```
_:a1  cito:disagreesWith   _:b1 .
_:a2  argumentmodel:proves  _:b2 .
```

Listing 1: Relationship between annotations

This example demonstrates relationships between two annotation instances. This type of construct is well suited for describing scientific discourse, building citation links, characterising citations (e.g. CiTO), describing experiments, etc. The annotation in the object position of the triple does not have to be in the same paper. This way one can easily describe interesting relationships between text fragments in different papers. For example, instead of creating citation links between papers, which is the currently prevalent practice, one could more specifically reference the exact text the citation was based on within the cited paper. This reduces the amount of time needed to locate the context of a citation.

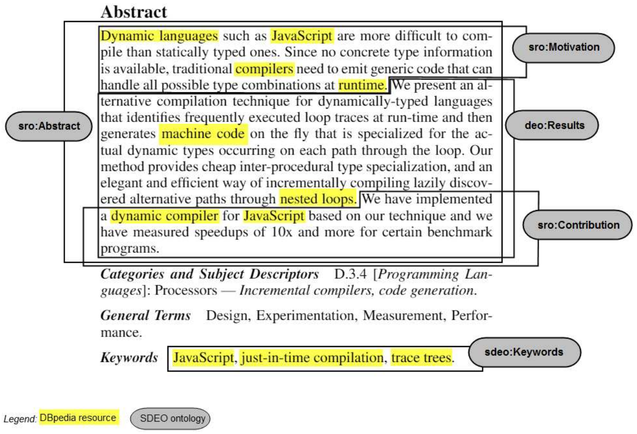
```
_:a3 cito:confirms <http://projectX.org/owl#Experiment1>
```

Listing 2: Relationship between an annotation and some other resource

This example represents a relationship between an annotation and a resource that is not an annotation. This construct is highly suitable for flexible reference management, but also for project-specific knowledge management. Consider a group of researchers collaborating on a project. They might decide to use a custom ontology for the project to organise their research. Subsequently, they would annotate papers with terms from that ontology. If we extended SemAnn to support user profiles, users could even create their personal ontologies on the fly to organise their research.

[7] http://mozilla.github.io/pdf.js/

[8] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtSponger



Abstract

Dynamic languages such as JavaScript are more difficult to compile than statically typed ones. Since no concrete type information is available, traditional compilers need to emit generic code that can handle all possible type combinations at runtime. We present an alternative compilation technique for dynamically-typed languages that identifies frequently executed loop traces at run-time and then generates machine code on the fly that is specialized for the actual dynamic types occurring on each path through the loop. Our method provides cheap inter-procedural type specialization, and an elegant and efficient way of incrementally compiling lazily discovered alternative paths through nested loops. We have implemented a dynamic compiler for JavaScript based on our technique and we have measured speedups of 10x and more for certain benchmark programs.

Categories and Subject Descriptors D.3.4 [*Programming Languages*]: Processors — *Incremental compilers, code generation.*

General Terms Design, Experimentation, Measurement, Performance.

Keywords JavaScript, just-in-time compilation, trace trees.

Figure 4: Annotations as instances of different vocabularies.

The **SemAnn Discourse Elements Ontology (SDEO)** extends the existing *Discourse Elements Ontology (DEO)*[9], an ontology for describing the major rhetorical elements of a scholarly paper, itself a member of the family of *Semantic Publishing and Referencing Ontologies (SPAR)*[10]. SDEO is used for annotations of special interest to the scientific community. The properties in this ontology serve a special purpose in our *hierarchical annotation model*.

We designed this model to be able to narrow down the *context* of an annotation. The necessity to do so is confirmed by our evaluation of the recommender functionality (cf. section 5), which shows how few semantic annotations per paper can result in a surprisingly wide recall of similar papers. It is of paramount importance to identify those matches that are likely to be most relevant to the user. Consider, for example, a single semantic annotation referring to http://dbpedia.org/resource/Marketing in the future work section of an engineering paper. Recommendations based just on this annotation, which is not representative for the paper in question, are unlikely to be useful. One solution is to consider the context of the annotation. Relevant context in scientific papers is often given by the main structures of scientific discourse: motivation, claims, and problem statements (cf. [21]). Hence, by encouraging users to identify fragments of scientific discourse, one can help them to put other annotations within it into a more useful context. We therefore preloaded SemAnn with SDEO.

Figure 4 gives an example of hierarchically nested annotations, which could have been added by different users at different times. SemAnn keeps track of the current hierarchical structure of the document in a separate graph in the triple store and updates it each time a new annotation is added. It is thus aware of the context of the annotation, which is given by the parent annotation.

Whilst hierarchical annotations are straightforward to implement if the publication itself is in a hierarchical format such as XML, this is not the case with PDF files. Novel for PDF annotation applications, the SemAnn architecture overcomes this limitation by taking into account the end and start positions of the annotation within the file in finding the best parent match and thus making additional information available by being aware of that hierarchy.

We favoured DEO as a foundation for its good coverage of the main structural elements that are relevant in the context of semantic search of scientific papers; nor is it too complex to discourage users from using it. It was then extended by additional structural elements of scientific papers such as keywords or title. Also, DEO comes with the transitive properties *hasPart* and *isPartOf*, which support queries that reason over hierarchical annotations. One could then determine all the parent annotations of an annotation or vice versa - a needed functionality in order to answer question like the following: ‘what type of annotations are included in the abstract of a paper?’ The following example shows how one annotation, i.e. the annotation of the text range from characters 74–87, is nested within another one (from characters 74–132):[11]

```
<rdf:RDF xml:base="http://eis.iai.uni-bonn.de/semann/pdf/
     example.pdf">
  <rdf:Description rdf:about="">
    <dct:hasPart rdf:resource="#page=1?char=74,87&amp;id
         =0/20/1/1:0,0/20/1/1:13"/>
    <dct:hasPart rdf:resource="#page=1?char=74,132&amp;id
         =0/20/1/1:0,0/21/1/1:13"/>
  </rdf:Description>
  <rdf:Description rdf:about="#page=1?char=74,132&amp;id
       =0/20/1/1:0,0/21/1/1:13">
    <dct:hasPart rdf:resource="#page=1?char=74,87&amp;id
         =0/20/1/1:0,0/20/1/1:13"/>
  </rdf:Description>
</rdf:RDF>
```

Instead of being limited to querying whether a paper contains an annotation of some type, one can now check whether it appears in the context of an abstract or some other structural element relevant to scientific discourse. The example query below is a compact yet powerful query that a user familiar with SPARQL can easily understand and write. This was achieved by keeping the SemAnn ontologies as lightweight as possible.

```
# return publications that refer to DBpedia resources in
     the abstract
PREFIX semann: <http://eis.iai.uni-bonn.de/semann/0.2/owl
     #>
PREFIX sro: <http://salt.semanticauthoring.org/ontologies
     /sro#>
PREFIX : <http://purl.org/dc/terms/>

SELECT ?file ?dbpediaResource
FROM <http://eis.iai.uni-bonn.de/semann/graph/meta>
    # hierarchy of annotations
FROM <http://eis.iai.uni-bonn.de/semann/graph>
    # annotations
WHERE { # NB! transitive property paths in use
  ?file         a          semann:Publication ;
                :hasPart* ?abstract .
  ?abstract     a          sro:Abstract ;
                :hasPart* ?abstractTerm .
  ?abstractTerm a ?dbpediaResource .
  FILTER (STRSTARTS(STR(?dbpediaResource), "http://
      dbpedia.org"))
}
```

This knowledge model enables context-specific semantic queries such as the following:

‘Which publications are motivated by dynamic programming languages?’ One could query for all papers that contain the annotation http://dbpedia.org/resource/Dynamic_programming_language, its `owl:sameAs` equivalents in the LOD cloud or its subcategories in the context of a *motivation*.

‘What ontologies have been used most to annotate computer science publications?’ Starting from the DBpedia resources that have been used for annotations in the ‘keywords’ section of papers, one could explore the DBpedia category hierarchy or the SKOS-based ACM Computing Classification System[12] or employ more sophist-

[9] http://purl.org/spar/deo/

[10] http://sempublishing.sourceforge.net

[11] We use RDF/XML for readability (!), as in Turtle one would have to escape the characters `?,&/` in URI references of the form `prefix:localname`.

[12] http://www.acm.org/about/class/

icated topic analysis tools such as Rexplore[13], to determine which publications are likely to belong to the same field, in this case computer science. One could then query over those publications to see what concepts from what ontologies have been used to annotate these publications. To readers new to a field, this query could give an overview of the different ontologies that are popular in this field.

'Which ontology concepts could potentially mean the same thing?' Since SemAnn is intended to be used for crowdsourcing in the sense of collectively building a knowledge base, it can easily happen that different users annotate the same text fragment. Such cases can be easily identified, and the second annotator could be asked to verify whether the same thing is meant. The value of this becomes easy to understand when considering two users from different scientific disciplines, who may refer to the same concept with different terms, each with the one commonly used in their own field. This provides an opportunity to find semantically equivalent concepts across different ontologies. This would enable users from different scientific disciplines to better understand research in other areas and above all, make such research results accessible to semantic search. There is also a bonus for ontology engineers, who can now incorporate such information into their ontologies.

5. RECOMMENDING RELATED PAPERS

The SemAnn recommendation functionality finds papers similar to the one being viewed in the following way:

1. Find other papers where the same DBpedia resource has been used in annotations as in the currently opened paper.

2. Find other papers where similarity between papers is established on a common subject category level of DBpedia resources present in both papers.

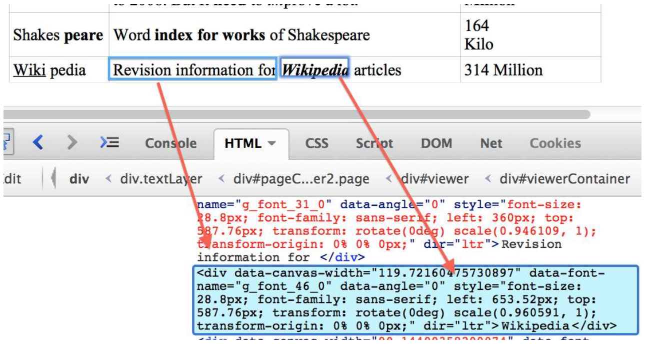
3. Check if any of the papers found so far have similar annotations within the same structural context as the current paper.

This simple algorithm displays all papers where the above conditions hold, without any ordering or filtering of matches. It was developed as a demonstration of how annotations could be further used. The notion of 'subject category' currently depends on DBpedia but could easily be generalised to other annotation vocabularies with a hierarchical structure, e.g. any SKOS scheme.

Figure 5 shows how a match is made between two papers, which both have an annotation from the same DBpedia subject category (http://dbpedia.org/resource/Category:Coatings) in the context of the abstract. The ability to query within the specific context of an annotation type (which is not limited to types from SDEO) is highly useful in the context of recommendations. Evaluation of the precision of various metadata fields in content-based recommendation systems has shown that the most valuable matches are often made based on abstracts, keywords and the title [12]. As a result, special weights could be given to semantic matches within that context when ordering results. If a match is found in the same structural context, then this is emphasised with a corresponding label next to the specific justification.

6. TABLE EXTRACTION

In case of tabular data, the rows and columns are identified for which the user can manually or automatically apply additional semantics. E.g. if a column contains countries then we can automatically look up table cell values based on that information and identify each country as a specific DBpedia[14] resource. This tabular data is then mapped to the RDF Data Cube Vocabulary[15] knowledge model and stored in the triple store.



Shakes **peare**	Word **index for works** of Shakespeare	164 Kilo
Wiki pedia	Revision information for ***Wikipedia*** articles	314 Million

```
name="g_font_31_0" data-angle="0" style="font-size:
28.8px; font-family: sans-serif; left: 360px; top:
587.76px; transform: rotate(0deg) scale(0.946109, 1);
transform-origin: 0% 0% 0px;" dir="ltr">Revision
information for </div>
<div data-canvas-width="119.72160475730897" data-font-
name="g_font_46_0" data-angle="0" style="font-size:
28.8px; font-family: sans-serif; left: 653.52px; top:
587.76px; transform: rotate(0deg) scale(0.960591, 1);
transform-origin: 0% 0% 0px;" dir="ltr">Wikipedia</div>
```

Figure 6: Table cell text spread across multiple DIV elements.

In order to semantically annotate tabular data in PDF documents, tables must be successfully extracted without any loss of information. Unlike desktop applications, for a client-side web-based application such as SemAnn, there are limited means for accessing the raw content of the PDF files. SemAnn uses the *PDF.js* library for rendering PDF documents in a web-browser for reasons addressed in section 3 and there were some rendering specific challenges that needed to be overcome.

Namely, *PDF.js* renders PDF text as HTML DIV elements. Every change in style results in a separate DIV element that is associated with its own CSS style sheet. This means that a single table cell can be split between multiple DIV elements and the challenge is to identify which of them belong to the same cell – see the example in Figure 6. Similarly, various alignments of columns and broken table rows present a challenge in the extraction process. We performed an empirical analysis and categorised those broken styled text segments based on how they appear in the DIV elements – see Figure 7. The SemAnn tool's extraction algorithm is capable of supporting all cases but the last three.

The resulting extracted tables are represented in the SemAnn knowledge model adhering to the RDF data cube vocabulary and thus available for querying through the SPARQL endpoint. For aligning the extracted RDF, we developed a recommendation approach for mapping table columns to RDF properties, which uses LODStats and LOV statistics and suggests suitable RDF properties to the user based on label matches (cf. Figure 8).

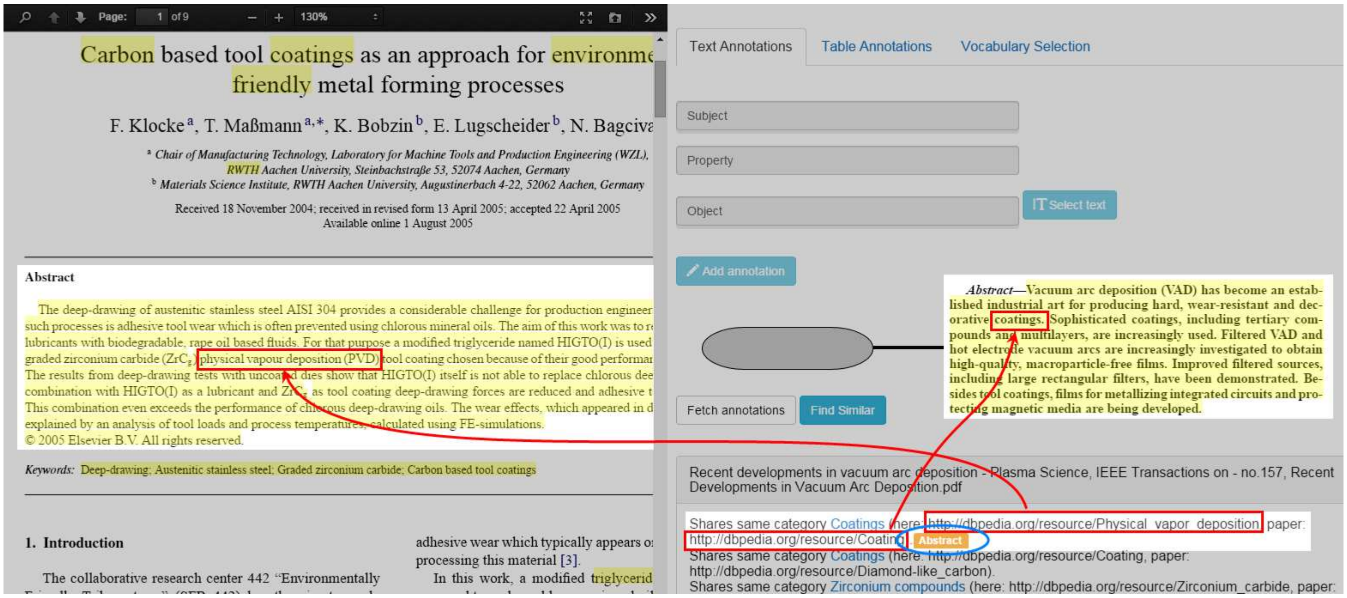
7. EVALUATION

The non-tabular annotations and the usefulness of recommendations were evaluated with 10 test users. The extraction of tabular data was evaluated separately with 5 test users. The test users, whose experience level was determined via a questionnaire, either read research papers daily, had research experience, or were familiar with the RDF data model. For non-tabular annotation evaluation the test group was asked to annotate a previously unannotated test paper from a familiar domain within 10 minutes. For tabular annotation evaluation the test group was asked to perform specific tasks within 10 minutes. As candidates for recommendation, we had prepared two datasets of annotated papers. We determined the usefulness of the recommendations the users' annotation would yield, considering two differently prepared sets of recom-

[13] http://technologies.kmi.open.ac.uk/rexplore/

[14] http://dbpedia.org

[15] www.w3.org/TR/vocab-data-cube



Carbon based tool coatings as an approach for environme
friendly metal forming processes

F. Klocke[a], T. Maßmann[a,*], K. Bobzin[b], E. Lugscheider[b], N. Bagciva

[a] Chair of Manufacturing Technology, Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Steinbachstraße 53, 52074 Aachen, Germany
[b] Materials Science Institute, RWTH Aachen University, Augustinerbach 4-22, 52062 Aachen, Germany

Received 18 November 2004; received in revised form 13 April 2005; accepted 22 April 2005
Available online 1 August 2005

Abstract

The deep-drawing of austenitic stainless steel AISI 304 provides a considerable challenge for production engineer such processes is adhesive tool wear which is often prevented using chlorous mineral oils. The aim of this work was to r lubricants with biodegradable, rape oil based fluids. For that purpose a modified triglyceride named HIGTO(I) is used graded zirconium carbide (ZrC_g) physical vapour deposition (PVD) tool coating chosen because of their good performan The results from deep-drawing tests with uncoated dies show that HIGTO(I) itself is not able to replace chlorous dee combination with HIGTO(I) as a lubricant and ZrC_g as tool coating deep-drawing forces are reduced and adhesive t This combination even exceeds the performance of chlorous deep-drawing oils. The wear effects, which appeared in d explained by an analysis of tool loads and process temperatures, calculated using FE-simulations.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Deep-drawing; Austenitic stainless steel; Graded zirconium carbide; Carbon based tool coatings

1. Introduction

The collaborative research center 442 "Environmentally

adhesive wear which typically appears o processing this material [3].

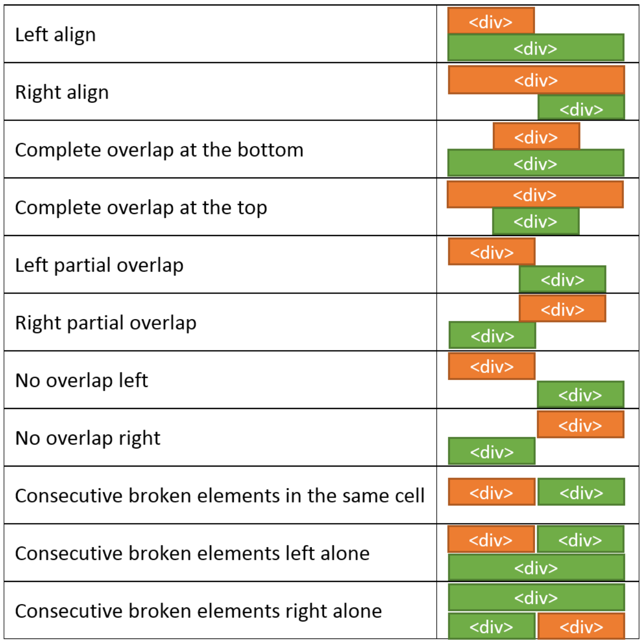
In this work, a modified triglycerid

Abstract—Vacuum arc deposition (VAD) has become an established industrial art for producing hard, wear-resistant and decorative coatings. Sophisticated coatings, including tertiary compounds and multilayers, are increasingly used. Filtered VAD and hot electrode vacuum arcs are increasingly investigated to obtain high-quality, macroparticle-free films. Improved filtered sources, including large rectangular filters, have been demonstrated. Besides tool coatings, films for metallizing integrated circuits and protecting magnetic media are being developed.

Recent developments in vacuum arc deposition - Plasma Science, IEEE Transactions on - no.157, Recent Developments in Vacuum Arc Deposition.pdf

Shares same category Coatings (here: http://dbpedia.org/resource/Physical_vapor_deposition, paper: http://dbpedia.org/resource/Coating). Abstract
Shares same category Coatings (here: http://dbpedia.org/resource/Coating, paper: http://dbpedia.org/resource/Diamond-like_carbon).
Shares same category Zirconium compounds (here: http://dbpedia.org/resource/Zirconium_carbide, paper:

Figure 5: Example of SemAnn's recommendation functionality.



Left align	<div> <div>
Right align	<div> <div>
Complete overlap at the bottom	<div> <div>
Complete overlap at the top	<div> <div>
Left partial overlap	<div> <div>
Right partial overlap	<div> <div>
No overlap left	<div> <div>
No overlap right	<div> <div>
Consecutive broken elements in the same cell	<div> <div>
Consecutive broken elements left alone	<div> <div> <div>
Consecutive broken elements right alone	<div> <div> <div>

Figure 7: DIV layouts recognised by the table extraction (last three cases currently not yet supported).

mendation candidates. We also evaluated usability by analysing the annotations made by both test groups and feedback collected in interviews. For the full detail of the evaluation results, we refer to the authors' master's theses [20, 11].

Relatedness of Recommendations.

To determine the relatedness of papers recommended, we prepared a set of 10 recommendation candidates with annotations, including hierarchically nested ones. By our own subjective assessment of their relation to the unannotated paper, we divided these candidates into three categories: (a) closely related papers (40%), (b) vaguely related papers (30%), (c) unrelated papers (30%).

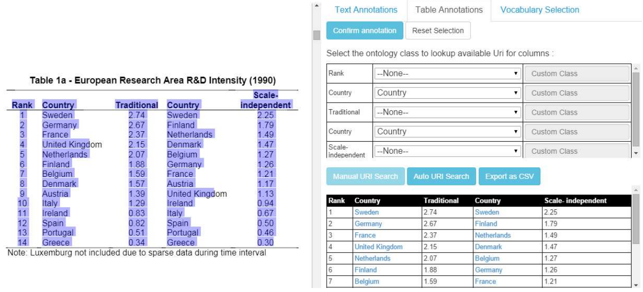


Table 1a - European Research Area R&D Intensity (1990)

Rank	Country	Traditional	Country	Scale-independent
1	Sweden	2.74	Sweden	2.25
2	Germany	2.67	Finland	1.79
3	France	2.37	Netherlands	1.49
4	United Kingdom	2.15	Denmark	1.47
5	Netherlands	2.07	Belgium	1.27
6	Finland	1.88	Germany	1.26
7	Belgium	1.59	France	1.21
8	Denmark	1.57	Austria	1.17
9	Austria	1.39	United Kingdom	1.13
10	Italy	1.29	Ireland	0.94
11	Ireland	0.83	Italy	0.67
12	Spain	0.82	Spain	0.50
13	Portugal	0.51	Portugal	0.46
14	Greece	0.34	Greece	0.30

Note: Luxemburg not included due to sparse data during time interval

Select the ontology class to lookup available Uri for columns :

Rank	--None--	Custom Class
Country	Country	Custom Class
Traditional	--None--	Custom Class
Country	Country	Custom Class
Scale-independent	--None--	Custom Class

Rank	Country	Traditional	Country	Scale- independent
1	Sweden	2.74	Sweden	2.25
2	Germany	2.67	Finland	1.79
3	France	2.37	Netherlands	1.49
4	United Kingdom	2.15	Denmark	1.47
5	Netherlands	2.07	Belgium	1.27
6	Finland	1.88	Germany	1.26
7	Belgium	1.59	France	1.21

Figure 8: Table extraction with the SemAnn tool.

As Figure 9 shows, a full recall of closely related papers was achieved even with few annotations; this observation is further supported by the second experiment below. Contrary to our expectation, one test user was recommended two seemingly unrelated papers. Here, a match was identified via the very general DBpedia subject category *American Inventions*[16], which includes concepts as distant as "markup language" (a subject of the test paper), "JavaScript" and "solar cell" (respectively the subjects of the two recommended papers).

Recall of Recommendations.

We measured the recall of papers recommended from a set of 30 candidate papers pre-annotated with five unique DBpedia resources each. The annotations were placed in the title, abstract, keywords, introduction or conclusion sections and chosen to be representative of the paper as a whole. We used the same user annotations as in the "relatedness" experiment and, once more, divided the candidates into closely related, vaguely related and unrelated ones. From the observed average recall of 97% of closely related papers we conclude that a high recall does not require many annotations, provided that they are well-chosen and representative of the paper as a whole.

[16] http://dbpedia.org/resource/Category:American_inventions

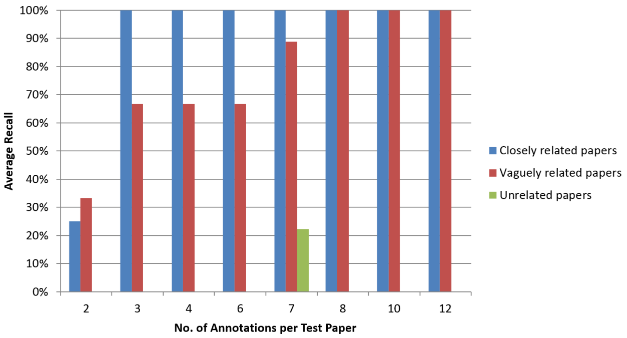


Figure 9: Recall vs. no. of user's annotations for experiment 1

Usability.

The 10 test users rated the overall usability of the annotation facility with an average 3.15 points out of 5[17] (2.6 for the three daily readers of research papers), and 3.45 points for the recommendation facility. The interviews confirmed that the main reasons for the low score of the annotation facility were a lack of precision of selecting text in the PDF document, and a lack of suggested matches from DBpedia. 67% of the annotations pointed to DBpedia terms, 16% used the SDEO ontology, and the rest were annotations with user-defined terms. 7 out of 10 users found the phrasing of justifications of the recommendations hard to understand. 5 out of 10 users requested a ranking of recommendations by relevance. Several suggestions were made about displaying further context about recommended papers, e.g. the abstract (4 out of 10 users).

In the evaluation for tabular data participants rated the difficulties experienced during table annotation as 2 points out of 5 (values normalised to match the scale of the non-tabular annotation evaluation study). Difficulties during export into CSV was rated as 1.6 points out of 5. Participants experienced most issues with SPARQL querying over data cube with the difficulty rating of 4.4 points out of 5. This was largely discovered to be due to the unfamiliarity of the participants with the RDF Data Cube Vocabulary. Hence there is still some improvement for increasing the ease of use of table annotations. During the semantic annotation of table values, all the participants preferred the automatic URI searching functionality to that of the manual. Users reported this to be faster and easier to use than the manual option with good enough precision. Overall, some small UI related issues were reported and some further future work is needed to improve the user experience.

Lessons learned.

Almost all recommended papers were related to the user annotated one; the 'unrelated' recommendation could be serendipitous.[18]

The words in the *abstract* of a paper proved particularly suitable as a basis for good recommendations in our "recall" experiment (cf. section 5), which previous research by Nascimento et al. [17] and the developers of the Mendeley reference management system [12] confirms. Hence, we expect further benefits from encouraging SemAnn users to prioritise the annotation of the abstract.

To determine the *scalability* of the recommendation user interface, we furthermore fed the 30 candidate papers of the "recall" experiment into the recommendation engine in batches of 10 (each comprising the same shares of closely related, vaguely related and unrelated papers). We observed a linear correlation between the number of candidates and the number of recommendations per test user. Coinciding with the users' feedback, this emphasises the need for ranking the recommendations by relevance.

Indeed, the users' feedback about the *recommendation facility* mainly focused on the presentation of the recommendations. Most of these improvements will be straightforward to implement within SemAnn, except for displaying further context about recommended papers. Doing so will require extracting information from the PDFs using, e.g., the *CiteSeerExtractor API*[19], or employing external scientific information services to determine context that is not included in the paper PDF, such as the year of publication.

One of the two reasons for the users' not-excellent rating of the *annotation facility* in both evaluation studies, the lack of precision of selecting text, is to blame on PDF.js. The lack of suggested matches from the general-purpose dataset DBpedia could be addressed by applying a named entity recognition technology similar to DBpedia Lookup to domain-specific datasets. Furthermore, we plan to reduce the complexity of the annotation fields for users unfamiliar with RDF, e.g. by providing a simplified view without subject/predicate/object fields, as suggested by 3 out of 10 users.

8. CONCLUSIONS & FUTURE WORK

In this work we presented a concept and its implementation for semantically annotating scientific publications in the PDF format. Compared to existing solutions, SemAnn offers the following key advantages:

- It supports the user in the *semantic annotation of text and tables in PDF* publications, the format most widely used but largely neglected by existing semantic annotation tools.
- It can be used with arbitrary ontologies as annotation vocabularies. This makes it a *general purpose semantic annotation tool*, whose applicability is not limited to a specific application in a specific domain.
- Its functionality goes *beyond semantic classification* capabilities. Various levels of expressivity are supported, including the ability to express relationships between annotations themselves.
- It is capable of *viewing annotations in the context of scientific discourse* (but not limited to it). This enables reasoners to answer questions such as 'find papers where the problem statement of the paper addresses [. . .].'

With its recommendations of similar papers, SemAnn provides an immediate benefit in return for making the effort of annotation. The justification of recommendations includes information about matches by structural context. SemAnn supports multiple users in collectively building a high-quality knowledge base with little effort, as it deduces the structural context of new annotations from previously existing annotations. Finally, it adheres to the Linked Data principles by enabling users to link their annotations to standard vocabularies and to the widely used DBpedia dataset. SemAnn's own ontologies reuse existing ontologies for annotating scientific publications and the tool itself is open source.

We evaluated the usability of SemAnn and the usefulness of its recommendations by analysing annotations resulting from tasks assigned to test users and by interviewing them. While the evaluation shows that even five well-crafted annotations per paper lead to an

[17] Rating scale ran from 1 to 5 with 5 as maximum.

[18] The one affected test user actually commented 'I understand why it is in the results but it is not useful to me', which suggests a need for further experiments.

[19] http://citeseerextractor.ist.psu.edu

almost total recall, we also observed unexpected, serendipitous recommendations, which confirms the merit of our low-threshold annotation support for the crowd.

Future Work. Besides the improvement suggestions from the evaluation, we consider the most important future work to be in extending SemAnn to support communication with the *Annotopia Open Annotation Server* [3], an open universal hub for storing and publishing of annotations in the Open Annotation ontology. Semantic annotations created with SemAnn could then be used in other tools such as Utopia, once uploaded to the Annotopia server. Likewise, data on the Annotopia server could help to improve the quality of query results in SemAnn, e.g. for suggesting annotations or recommendations. Such an integration of annotations and annotation tools would considerably increase the visibility of annotations and thus the usefulness of the annotated documents. This would bring us a step closer to the vision of semantic publishing: opening up data to everybody, i.e. not just the scientific community.

We also consider it important to partly automate annotation by using external APIs. *DBpedia Spotlight*[20] could, e.g., be used for automatically annotating mentions of DBpedia resources in the PDF text. Using annotation targets other than DBpedia would be desirable but requires integration work, as DBpedia Lookup depends on the DBpedia extraction framework and is thus incompatible with other datasets. As an intermediate step beyond DBpedia, one could explore 'same as' links pointing from DBpedia resources to other datasets. Also improvements to the semi-automatic annotation of tabular data will be considered to include support for additional vocabularies. Integration with a visualisation tool like CubeViz[21] would also result in improving the usefulness of the annotated tabular data. Annotopia comes ready with plug-ins for entity recognition that would enable automatic recognition of ontology concepts. While the SemAnn knowledge model provides excellent conditions for reasoning about annotations, including the ability to reason by context (e.g. abstract, motivation, etc.) and to serve recommendations with precise justifications, there exist much more sophisticated linked data based recommender systems (cf. [6]), which we could employ to further improve the precision and recall of the recommendations – and thus of the benefit paid in return for annotating. Similarly, extending the recommender functionality to automatically recognise similarities between tables in various papers would be a useful and interesting feature. Finally, based on feedback from an expert user, we will explore using SemAnn beyond scientific publications – specifically for patents.

References

1. Attwood, T. K. et al. Utopia documents: Linking Scholarly Literature With Research Data. In: Bioinformatics 26(18) (2010).
2. Bikakis, N. et al. Integrating Keywords and Semantics on Document Annotation and Search. In: *ODBASE*. 2010.
3. Ciccarese, P., Clark, T. Annotopia: An Open Source Universal Annotation Server for Biomedical Research. In: *SWAT4LS*. CEUR-WS 1320. 2014.
4. Ciccarese, P., Ocana, M., Clark, T. Open semantic annotation of scientific publications using DOMEO. In: Biomedical Semantics 3(Suppl 1) (2012).
5. Ciccarese, P. et al. An open annotation ontology for science on web 3.0. In: Journal of Biomedical Semantics 2(4) (2011).
6. Di Noia, T., Cantador, I., Ostuni, V. C. Linked Open Data-enabled Recommender Systems. In: *Semantic Web Evaluation Challenges 2014*. CCIS 457. Springer, 2014.
7. Eriksson, H. An Annotation Tool for Semantic Documents. In: *The Semantic Web: Research and Applications*. LNCS 4519. Springer, 2007.
8. Estellés-Arolas, E., Guevara, F. González-Ladrón-de. Towards an Integrated Crowdsourcing Definition. In: Information Science 38(2) (2012).
9. Groza, T. et al. SALT: Weaving the Claim Web. In: *ISWC/ASWC*. LNCS 4825. Springer, 2007.
10. Heese, R. et al. One Click Annotation. In: *Scripting and Development for the Semantic Web (SFSW)*. CEUR-WS 699. 2010.
11. Islam, A. S. Crowdsourced Semantic Annotation of Scientific Publications and Tabular Data in PDF. MA thesis. Universität Bonn, 2014. http://purl.net/eis/theses/2014/islam.pdf.
12. Jack, K. Mendeley: Recommendation Systems for Academic Literature. Presentation at TU Graz. 2012. http://www.slideshare.net/KrisJack/mendeley-recommendation-systems-for-academic-literature (visited on 2015-01-17).
13. Khalili, A., Auer, S., Hladky, D. The RDFa Content Editor - From WYSIWYG to WYSIWYM. In: *COMPSAC*. 2012.
14. Kohlhase, A. Semantic Interaction Design: Composing Knowledge with CPoint. PhD thesis. Computer Science, Universität Bremen, 2008. http://kwarc.info/ako/pubs/AKo_Promo.pdf.
15. Kohlhase, A., Kohlhase, M., Lange, C. sTeX – A System for Flexible Formalization of Linked Data. In: *International Conference on Semantic Systems (I-Semantics) and International Conference on Pragmatic Web*. ACM, 2010. arXiv: 1006.4474v1 [cs.SE].
16. Lange, C., Di Iorio, A. Semantic Publishing Challenge. Assessing the Quality of Scientific Output. In: *Semantic Web Evaluation Challenges 2014*. CCIS 457. Springer, 2014.
17. Nascimento, C. et al. A Source Independent Framework for Research Paper Recommendation. In: *Joint Conference on Digital Libraries (JCDL)*. ACM, 2011.
18. Seifert, C. et al. Crowdsourcing Fact Extraction from Scientific Literature. In: *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer, 2013.
19. Semantic Web Evaluation Challenges 2014. CCIS 457. Springer, 2014.
20. Takis, J. Crowdsourced Semantic Annotation of Scientific Publications. MA thesis. Universität Bonn, 2014. http://purl.net/eis/theses/2014/takis.pdf.
21. Waard, A. de. From proteins to fairytales: Directions in semantic publishing. In: IEEE Intelligent Systems 25(2) (2010).

[20] http://spotlight.dbpedia.org/

[21] http://aksw.org/Projects/CubeViz.html