

Supporting Researchers with a Semantic Literature Management Wiki

Bahar Sateli and René Witte

Semantic Software Lab

Department of Computer Science and Software Engineering
Concordia University, Montréal, Canada

Abstract. Nowadays, research group members are often overwhelmed by the large amount of relevant literature for any given topic. The abundance of publications leads to bottlenecks in curating and organizing literature, and important knowledge can be easily missed. While a number of tools exist for managing publications, they generally only deal with bibliographical metadata and their support for further content analysis is limited to simple manual annotation or tagging. Here, we investigate how we can go beyond these approaches by combining semantic technologies, including natural language processing, within a user-friendly wiki systems to create an easy-to-use, collaborative space that facilitates the semantic analysis and management of literature in a research group. We present the *Zeeva* system as a first prototype that demonstrates how we can turn existing papers into a queryable knowledge base.

1 Introduction

The increasing growth of scientific publications has become the centre of attention of researchers from various domains, ranging from cognitive studies to computational linguistics. The apparent disproportion of human capabilities versus the pace of information generation has encouraged researchers to look for new approaches that can help to extract, organize, and manage knowledge from the immense amount of publications available in ever-growing repositories. To overcome this bottleneck, we envision a collaborative, wiki-based solution for the semantic management of research literature that integrates (*i*) a web-based interface, (*ii*) semantic knowledge representation, and (*iii*) text mining for automatic content analysis. Here, we report on the feasibility and usability of such an approach, based on a first prototype called *Zeeva*.

As a running example, consider a research group where various members work collaboratively on a specific topic. An ongoing task is the curation of relevant existing research publications, including background and related work. These need to be systematically organized, stored with bibliographical metadata, and shared in a way that allows team members to search, annotate, and comment on specific works. There exists a multitude of bibliographical management systems that offer basic functionality for this, both as web-based solutions (e.g., BibSonomy¹) or

¹BibSonomy, <http://www.bibsonomy.org/>

a self-hosted system (e.g., Aigaion²). However, none of these tools provide for semantic management of information that goes beyond managing bibliographical metadata or simple tagging, for example, to list the *contributions*, *claims*, *hypotheses*, or *results* stated in a paper. Representing these concepts explicitly would facilitate semantic linking, querying, and analyzing a body of research, and allow to relate the findings with a research topic under investigation (e.g., “semantic publishing”) within a research group. By using semantic standards like RDF³ we can open up knowledge “bottled up” in a paper to tools and methods from the semantic web, including semantic browsing, search, and visualization.

Ideally, information such as claims or contributions would be explicitly marked up in published research. However, this is unfortunately not the case for nearly all existing papers. Instead of relying on our research team members to provide these semantic annotations manually, we aim to support them with text mining pipelines that can automatically analyze and extract structural and semantical information from research papers. While a number of text mining tools and pipelines exist, none of them have so far been seamlessly integrated into a research literature management platform suitable for a research group. In our work, we apply the “Semantic Assistants” approach [1], where automated NLP assistants work collaboratively with humans on analyzing and semantically managing publications, thereby significantly increasing a group’s capacity for knowledge discovery, planning research, and conducting experiments.

The rest of this paper is organized as follows: The fundamentals of the techniques used in the Zeeva system are iterated in Section 2. Section 3 provides a high-level description of the Zeeva infrastructure followed by its implementation details in Section 4. The Zeeva wiki user interface is shown in Section 5, where we describe how users can use the various components in Zeeva to effectively curate scientific publications.

2 Background

In this section, we briefly introduce two fundamental concepts for our work: semantic wikis and natural language processing.

2.1 Semantic Wikis

Wikis are web applications that allow people to add, modify or delete content in a collaborative manner. Using a web browser interface and a simple markup language, wiki users can create and hyperlink *wiki pages*, making wikis “quick” and easy-to-use for authoring documents. Semantic wikis extend the idea of a collaborative authoring environment, where the content that is written for human reading purposes is combined with an underlying knowledge model that describes wiki content in a formal language suitable for automatic machine processing techniques. Among the existing semantic wikis, Semantic MediaWiki (SMW) [2] is

²Aigaion, <http://www.aigaion.de/>

³Resource Description Framework, <http://www.w3.org/RDF/>

a notable example. It extends the MediaWiki⁴ engine functionality by introducing a special markup that can be used to create semantic triples from the knowledge contained in a wiki.

In SMW, the underlying ontology is formed as semantic metadata and inserted into wiki pages by human users. A specific markup notation is manually typed into wiki pages to describe a property and its related value. Internally, SMW creates a semantic triple from the existing markup in the page and stores it in a relational database that can be queried from within the wiki pages using so-called *inline queries*. For example, users can dynamically create lists of cities located in a specific country with a population of over a million. Such queries directly make use of the semantic metadata available in the wiki repository to create and update such lists, thereby removing the overhead of manually maintaining the results by human users. While such capabilities indeed increase the usefulness of wikis in different applications, the downside of this approach is that the semantic markup has to be manually provided and maintained by human users.

2.2 Natural Language Processing

Natural Language Processing (NLP) is a fast-moving domain of research that uses various techniques from the Artificial Intelligence and Computational Linguistics areas to process text written in natural languages. NLP is a broad term encompassing general-purpose text processing techniques like text segmentation, to domain-specific applications, such as question-answering systems. One application from the NLP domain is *text mining*. Text mining aims at extracting high-quality structured information from usually free form text and represent it in a (semi-)structured format. As an essential part of the literature curation process, text mining techniques proved to be effective in terms of the time needed to extract and formalize the knowledge contained within a document. As the use of NLP techniques in software is being gradually adopted by developers, various applications have emerged to enable software engineers to integrate NLP capabilities in their applications, e.g., based on web services. To facilitate the development of re-useable components and their configuration into NLP pipelines, frameworks, such as the General Architecture for Text Engineering (GATE) [3] used in our project, have become a standard foundation. However, a seamless integration of these NLP techniques within external applications is still a major challenge. This is addressed by the Semantic Assistants project [4]. The core idea behind the Semantic Assistants framework is to create a wrapper around NLP pipelines and publish them as W3C standard Web services,⁵ thereby allowing a vast range of software clients to consume them within their environment.

3 Design of a Collaborative Semantic Literature Platform

We now list the requirements for our collaborative semantic literature analysis platform and then describe the design that we derived from these requirements.

⁴MediaWiki, <http://www.mediawiki.org>

⁵Web Services Architecture, <http://www.w3.org/TR/ws-arch/>

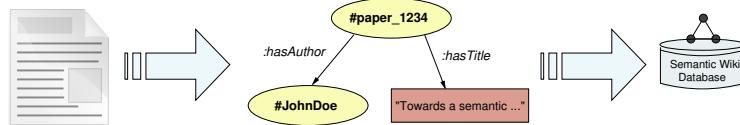


Fig. 1. Formalizing Paper Content in Semantic Wikis

3.1 Requirements

Coming back to our scenario from the introduction, we can identify three major requirements for semantic literature management:

Centralized Repository of Knowledge (R1). Scientists need a tool that can manage different types of artifacts generated throughout an analysis task (e.g., articles, bibliography data, metadata, personal notes) in a centralized repository. Therefore, the proposed system must provide users with the ability to store raw data (the original articles), as well as any information generated by users (e.g., textual annotations) and analysis tools (e.g., automated extraction of contributions).

Automatic Text Analysis Support (R2). Different tasks in literature analysis can be supported with various automatic text processing techniques. These techniques are themselves diverse in their concrete implementation and the resources they use. Hence, the proposed system must provide access to various NLP pipelines in a unified manner.

Collaborative Analysis Environment (R3). Many scientific articles are the result of collaborative studies between two or more researchers. Therefore, the proposed system shall provide an environment where all researchers have access to the most up-to-date information and can keep track of content modifications.

3.2 Design Decisions

Based on these requirements, we made three fundamental design decisions: Provide a wiki-based interface to support collaboration (R3), use a semantic engine for knowledge representation (R1) and integrate both with text mining services (R2).

Wiki-based Collaborative Web Interface. To address (R3), we have developed a wiki-based application as an evaluation platform for experimenting how various semantic services can support different user groups in concrete literature analysis tasks like reviewing papers. Wiki systems like MediaWiki are lightweight, collaborative authoring environments that are easy-to-use and highly scalable.

Semantic MediaWiki as a Knowledge Base. One of the core ideas behind the Zeeva system is to formalize the knowledge contained in scientific publications (R1). Fig. 1 shows the workflow of the system in an example scenario where a researcher wants to store some bibliographic metadata as well as a list of contributions of a paper into the wiki.

To provide for a semantic representation of a publication, semantic markup needs to be created (manually or automatically) and entered into a repository. To support the semantic representation, we selected a semantic wiki engine, as it can be seamlessly integrated with the wiki user interface (R3) on the one hand and the text mining pipelines on the other hand (R2). In a semantic wiki, semantic markup for the content that was found is saved in a wiki page. Upon saving the page, the underlying wiki engine processes the semantic markup present in the page and transforms the natural language representation of the semantic metadata into RDF triples using its custom markup parsers. The semantic triples are then stored in the wiki repository where they can be queried directly within the wiki environment. They also become accessible from external applications through an RDF feed.

Text Mining Pipelines for Literature Analysis. The described scenario above heavily relies on human capabilities in reading text and formalizing the extracted knowledge in a semantic wiki. Luckily, a multitude of NLP techniques, such as *Information Extraction* (IE) already exist that can support researchers in automatically extracting entities of interest from text (R2). Consequently, we designed the Zeeva platform in a way that arbitrary NLP pipelines can be seamlessly made available to researchers within the wiki interface, eliminating any unnecessary context switching to an external NLP applications during a researcher's workflow.

By leveraging the service-oriented architecture of the Semantic Assistants [4], we are able to add or remove NLP pipelines from the Zeeva wiki interface without the need to modify its core engine (R2). We have deployed a number of text mining pipelines in Zeeva that are suitable for the context of literature analysis:

Automatic Indexer: The indexer pipeline can generate a classical back-of-the-book style index. The pipeline uses the open source Multi-Lingual Noun Phrase Extractor (MuNPEx)⁶ and generates an inverted index of the noun phrases found in a text. The index is stored in a new wiki page, with hyperlinks to the corresponding wiki articles. This pipeline can help researchers obtaining a high-level overview of a set of papers ‘at a glance’. It also enables them to discover unknown concepts mentioned in articles, which they would not be able to find with a keyword-based search approach [5].

Readability Metrics: This pipeline calculates standard readability metrics, like Flesch and Kincaid [6], from a given document and generates various scores for the readability quality of a document. This pipeline can help researchers to assess the general writing quality of an article.

Claims and Contribution Extraction: A custom pipeline designed specifically for the purpose of literature analysis. It targets the particular need of researchers to automatically extract claims and contributions of a given paper in a verbatim format. Such extracted metadata can be used to find related work in their domain of interest.

⁶Multi-Lingual Noun Phrase Extractor (MuNPEx), <http://www.semanticsoftware.info/munpex>

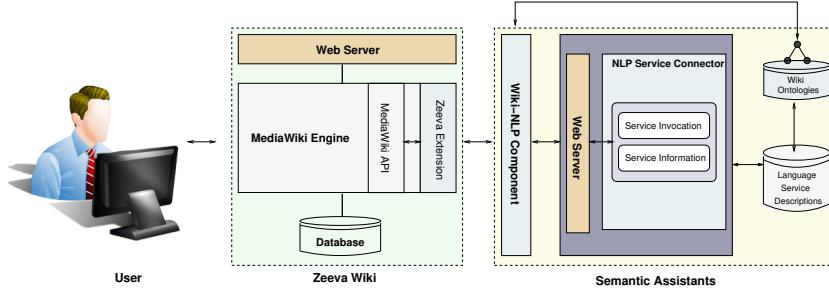


Fig. 2. A high-level overview of Zeeva system architecture

These NLP pipelines are seamlessly integrated with the wiki user interface, based on our Wiki-NLP integration architecture [1].

4 Implementation

In this section, we explain the technical details of the Zeeva system implementation and illustrate how NLP results are transformed into semantic metadata.

4.1 System Architecture

The front-end of the Zeeva system is a wiki application that provides users with a platform to store and record their findings with versioning mechanisms. Users interact with the Zeeva wiki using their Web browser and can view and edit the wiki content using a simple *markup* language. The core functionality of the MediaWiki engine powering our Zeeva system can be extended through installing *extensions*. The Semantic MediaWiki⁷ (SMW) extension allows us to use a special markup in wiki pages to annotate parts of their content with formal descriptions. Each semantic markup translates into a semantic triple with the wiki page as the subject, the declared property as the predicate and the given value as the object. SMW stores the generated triples in the wiki repository that can be later queried both within the wiki, as well as from external applications through an RDF feed.

In addition, we have designed *Zeeva Pubs*, a custom extension for MediaWiki that allows wiki users to invoke arbitrary NLP pipelines on a given article. The Zeeva Pubs extension uses the MediaWiki API to communicate with users through the wiki interface. Through a special page provided by the extension, users can specify a document URL for an analysis task, along with a list of NLP pipelines that may aid them at their task at hand. The extension then sends the user-provided content to the Semantic Assistants server via a web service call, where it is received by the Wiki-NLP component. The Semantic Assistants framework then takes care of executing the designated pipelines on the paper text and writing the results back to the Zeeva wiki.

⁷Semantic MediaWiki, <http://semantic-mediawiki.org>

The Wiki-NLP component in the Semantic Assistants framework also bears the responsibility of transforming NLP pipelines output to semantic metadata. For example, when an NLP pipeline generates a readability score for a given paper, in addition to writing the score into the wiki page for displaying to human users, it generates a semantic triple with a *hasReadabilityScore* predicate and makes it persistent in the wiki database. This way, papers analyzed by the NLP pipelines will be implicitly enriched with metadata automatically extracted from their content and formalized for machine processing purposes. Fig. 2 provides a high-level overview of the Zeeva system architecture.

4.2 Literature Analysis Workflow

Having described the Zeeva system architecture, we now have to specifically define how we make use of the NLP pipelines results in order to generate semantic metadata in the wiki. In this paper, we treat the NLP pipelines as a black box, i.e., we do not describe the text mining techniques used in them. Rather, we are interested in seeing how the collaboration between the users and the Zeeva text mining pipelines can help with a given analysis task.

The input and output type of each NLP pipeline in the Semantic Assistants framework is precisely defined using the Semantic Assistants ontology. Once a user invokes a pipeline through the Zeeva wiki interface, a RESTful service request with the URL of the paper and a list of pipeline names is sent to the Semantic Assistants server. The Semantic Assistants server then fetches the content of the paper and executes all the user-selected pipelines one by one on the provided content. When the execution is finished, an XML document containing the analysis results is generated (Fig. 3) and sent to the Wiki-NLP component.

```

<saResponse>
  <annotation type="Title">
    <document url="http://www.semanticsoftware.info/.../mobiwis13_android.pdf">
      <annotationInstance content="Smarter Mobile Apps through Integrated ..."/>
    ...
  </document>
  </annotation>
</saResponse>

```

Fig. 3. XML document containing NLP pipeline results

Since the resulting XML document cannot be written directly to the wiki database, nor it is suitable for human reading, the Wiki-NLP component has to parse the XML document into another representation format. In Zeeva, we make use of the MediaWiki *templating* mechanism. The Wiki-NLP component transforms the pipelines' results to wiki-specific markup by parsing the XML document and embedding the output in pre-defined templates in the wiki. These templates are provided through installing our Zeeva Pubs extension, and define (*i*) the look and feel of the results when embedded in wiki pages and (*ii*) the semantic metadata that should be attached to each pipeline output instance,

since every field can be associated with a semantic property. Fig. 4 shows a *Publication* template with NLP pipeline results embedded in it.

```
{ {Publication
|Title = [[hasTitle :: Smarter Mobile Apps through Integrated Natural Language ...]]
|Author = [[hasAuthor :: Bahar Sateli, Gina Cook, and Rene Witte]]
...
}}
```

Fig. 4. A Publication template with embedded NLP results

The template markup is then written to the Zeeva database, where it can be accessed by users, either through a browser (Fig. 7) or exported by the SMW engine as an RDF document (Fig. 5).

```
<rdf:RDF>
<owl:Ontology rdf:about="http://localhost/Zeeva/.../Sateli-MOBIWIS2013">
  <owl:imports rdf:resource="http://semantic-mediawiki.org/swift/1.0"/>
</owl:Ontology>
<swift:Subject rdf:about="http://localhost/Zeeva/.../Sateli-2DMOBWIS2013">
<rdf:type rdf:resource="http://localhost/Zeeva/.../Category-3APublication"/>
<property:HasTitle rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
  Smarter Mobile Apps through Integrated Natural Language Processing Services
</property:HasTitle>
...
<!-- Created by Semantic MediaWiki, http://semantic-mediawiki.org/ -->
</rdf:RDF>
```

Fig. 5. An RDF document excerpt generated from the template

5 Application

We refer back to our running example from the introduction to show how a group of researchers can use the Zeeva system. The task under study is to obtain an overview of the research in a particular domain through curating relevant publications. More specifically, our researchers need to systematically organize the papers that they have found on the web and for each article extract (*i*) the bibliographical metadata, and (*ii*) a list of claims and contributions of that paper.

A *special* page in Zeeva wiki, shown in Fig. 6, allows users to provide a URL and a desired page name for the paper to be analyzed, as well as selecting one or multiple NLP *assistants* for the analysis task. Provided that the Wiki-NLP integration has adequate permissions to retrieve the article (e.g., from an open access repository or through an institutional license), the article is then passed on to all of the NLP pipelines chosen by the user in the Semantic Assistants server. Once all the pipelines are executed, the user is automatically redirected to the newly created page with the analysis results transformed into user-friendly representations, like lists or graphs. Fig. 7 shows the wiki page created with bibliographical metadata, like title and author names, extracted from a paper.

As for the semantic entities, Fig. 8 shows two rhetorical entities, namely, claims and contributions, automatically extracted by the “Claims and Contribution

Fig. 6. Creating a new publication page in Zeeva wiki

Extraction” text mining pipeline. Since Zeeva’s underlying wiki engine revisions all changes to wiki pages, users can review the pipelines’ output and modify them in case of erroneous results.

In our example, multiple users can follow the same process to automatically extract bibliographic and rhetorical entities from their designated papers. By exploiting the metadata from analyzed papers, the research team can now obtain an overview of the existing papers in the wiki by looking at their bibliographical data and their contained claims and contributions. Fig. 9 shows the results of an example query asking for all contributions of a specific author from the papers in the wiki, using the SMW inline query syntax:

```
{{#ask: [[Category: Publication]] [[hasAuthor:: Bahar Sateli]]
| ?hasTitle = Title
| ?hasContribution = Contribution}}
```

Tables, such as the one shown in Fig. 9, are created automatically by querying the semantic metadata that researchers generate in the wiki together with the intelligent NLP assistants. The advantage of this approach is that not only these tables are dynamically created and kept up-to-date by the wiki system, they also allow researchers to *discover* related findings present in the wiki which may have been imported and analyzed by other users of the system. Such semantic support in a collaborative environment can improve the productivity of researchers in tasks like literature reviews or finding experts.

The screenshot shows a semantic infosheet for a publication titled "Smarter Mobile Apps through Integrated Natural Language Processing Services". The interface includes a navigation sidebar with links like Main page, Recent changes, New Publication, Tools, and Browse properties. The main content area displays the following information:

Title	Smarter Mobile Apps through Integrated Natural Language Processing Services
Author	Bahar Sateli, Gina Cook, and René Witte
Affiliation	Semantic Software Lab, Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada, iLanguage Lab, Montréal, Canada
Abstract	Smartphones are fast becoming ever-present personal assistants. Third-party 'apps' provide users with nearly unlimited customization options. A large amount of content read on these devices is text based—such as emails, web pages, or documents. Natural Language Processing (NLP) can help to make apps smarter, by automatically analyzing the meaning of content and taking appropriate actions on behalf of their users. However, due to its complexity, NLP has yet to find widespread adoption in smartphone or tablet applications. We present a novel way of integrating NLP into Android applications. It is based on a library that can be integrated into any app, allowing it to execute remote NLP pipelines (e.g., for information extraction, summarization, or question-answering) through web service calls. Enabling a separation of concerns, our architecture makes it possible for smartphone developers to make use of any NLP pipeline that has been developed by a language engineer. We demonstrate the applicability of these ideas with our open source Android library, based on the Semantic Assistants framework, and a prototype application 'ForgotWho' that detects names, numbers and organizations in user content and automatically enters them into the contact book.
URL	http://www.semanticsoftware.info/system/files/mobiwis13_android.pdf

Fig. 7. Semantic infosheet of bibliographical metadata extracted from a sample paper

6 Related Work

WikiPapers⁸ is a semantic wiki with the goal of creating “the most comprehensive compilation” of literature focused on research of wikis through a community of volunteers. Users can create new entries for publications, journals, authors, event and datasets using semantic forms. Thereby, dynamic lists of publications by category, author or keywords can be generated and maintained online for researchers. AcaWiki⁹ is another semantic wiki system designed to “collect summaries and literature reviews of peer-reviewed academic research” and make them available to the general public. Any user can post a summary about an article on AcaWiki and provide additional bibliographic and practical relevance data (e.g., links to related news articles or blog posts) using the provided semantic forms. Although pursuing a similar goal of formalizing the body of knowledge contained in scientific publications, our approach does not rely solely on human users as primary providers of semantic metadata, but offers an innovative approach where users can benefit from state-of-the-art techniques from the natural language processing domain in the metadata generation process.

Orthogonal to the development of literature analysis tools, frameworks like SALT [7] have been developed to capture the knowledge in papers prior to publishing the actual documents. For example, the SALT framework provides a number of ontologies to formally describe the internal structure of documents and their related rhetorical elements, like *claims* or *evidences*. In addition, it offers special LATEX commands that authors can use to create metadata while they are

⁸WikiPapers, <http://wikipapers.referata.com>

⁹AcaWiki, <http://www.acawiki.org>

The screenshot shows a web-based application interface for managing research publications. At the top, there's a navigation bar with links for Refresh, Watch, Protect, Move, Delete, History, Edit, Edit with form, Discussion, and Page. A search bar is also at the top. The main content area has a title 'Sateli-MOBIWIS2013'. Below the title, there are tabs for Publication Infosheet, Claims and Contributions, Readability Metrics, and Reviews. There's a '(edit)' link in the top right of this section. The main content area contains two sections: 'Claims:' and 'Contributions:', each listing several bullet points describing the publication's features and contributions.

Fig. 8. Extracted semantic metadata from a sample publication

writing a document. While the development of such ontologies are effective steps towards providing interoperability of the extracted semantic entities, there is still no tool support that can help authors automatically enrich their documents with such markup. The focus of our approach in this research work, is to assess the feasibility of a semantic wiki-based literature analysis environment with integrated text mining support, rather than constructing a new ontology.

7 Conclusion and Future Work

Research groups are in dire need of novel methods and tools for managing scientific publications that go beyond simply storing bibliographical data. We propose Zeeva, a proof-of-concept system that demonstrates how the next generation of literature management tools can support research groups by transforming publications into an active knowledge base. With the notion of embedded “Semantic Assistants” performing text analysis, Zeeva plays the role of an intelligent agent within a collaborative research task.

There are a number of further steps, both in ongoing research and implementation. We plan to develop and integrate additional NLP pipelines that further automate the analysis of research publications. We are investigating the re-use of existing RDF Schemas and OWL ontologies for research publications and their annotation, in order to integrate them for a Linked Data context. Finally, we will perform a user study on a large group, to investigate what are currently the most time-consuming tasks and how much precisely a tool like Zeeva can help in terms of both effort and accuracy.

All Contributions of Bahar Sateli

Paper ID	Title	Contribution
ReqWiki-MUD2012	Can Text Mining Assistants Help to Improve Requirements Specifications?	<p>We integrated a number of text mining assistants into a wiki-based requirements engineering platform to investigate two key questions: Can software engineers without prior training in NLP effectively leverage these techniques?</p> <p>In order to understand how software engineers would interact with these new text mining assistants during the development of the specification, we performed a case study within two university courses in requirements engineering (one at the undergraduate level, one at the graduate level).</p> <p>We conducted a number of experiments to evaluate the NLP integration for our ReqWiki system along two dimensions, namely the usability of the text mining service for non-experts and their effectiveness for quality improvements.</p>
Sateli-MOBIWIS2013	Smarter Mobile Apps through Integrated Natural Language Processing Services	<p>We present a novel way of integrating NLP into Android applications. We demonstrate the applicability of these ideas with our open source Android library, based on the Semantic Assistants framework, and a prototype application ‘iForgotWho’ that detects names, numbers and organizations in user content and automatically enters them into the contact book.</p> <p>In this paper, we present the first open source NLP library for the Android platform that allows various applications to benefit from arbitrary NLP services through a comprehensive, service-oriented architecture.</p> <p>In what follows, we present a number of standard NLP tasks, with a focus on those relevant for mobile applications.</p> <p>As a part of our contribution and in order to demonstrate a general-purpose app offering arbitrary NLP services to Android mobile users, we have implemented an Android app, called the Semantic Assistants App, that offers a unique user interface to inquire and invoke NLP services on a user-provided content.</p> <p>To better demonstrate this use case, we implemented the iForgotWho (iFW)Android app and used its NLP capability on an example email message.</p> <p>Dates, locations and people can be automatically detected using named entity recognition and integrated in the creation of new events in a user’s agenda and entries in the contact book as we demonstrated with the iFW app.</p>

Fig. 9. Semantic queries to retrieve contributions of an author

References

1. Sateli, B., Witte, R.: Natural Language Processing for MediaWiki: The Semantic Assistants Approach. In: The 8th International Symposium on Wikis and Open Collaboration (WikiSym 2012), Linz, Austria, ACM (Aug 2012)
2. Krötzsch, M., Vrandečić, D., Völkel, M.: Semantic MediaWiki. In Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L., eds.: The Semantic Web (ISWC 2006). Volume 4273 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2006) 935–942
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggin, H., Petrank, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6). University of Sheffield, Department of Computer Science (2011)
4. Witte, R., Gitzinger, T.: Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: 3rd Asian Semantic Web Conference (ASWC 2008). Volume 5367 of LNCS., Bangkok, Thailand, Springer (2008) 360–374
5. Witte, R., Krestel, R., Kappler, T., Lockemann, P.C.: Converting a Historical Architecture Encyclopedia into a Semantic Knowledge Base. IEEE Intelligent Systems **25**(1) (January/February 2010) 58–66
6. DuBay, W.H.: The Principles of Readability. (2004)
7. Groza, T., Handschuh, S., Müller, K., Decker, S.: SALT - Semantically Annotated LATEX for Scientific Publications. In Franconi, E., Kifer, M., May, W., eds.: The Semantic Web: Research and Applications. Volume 4519 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 518–532