Adding Natural Language Processing Support to your (Semantic) MediaWiki

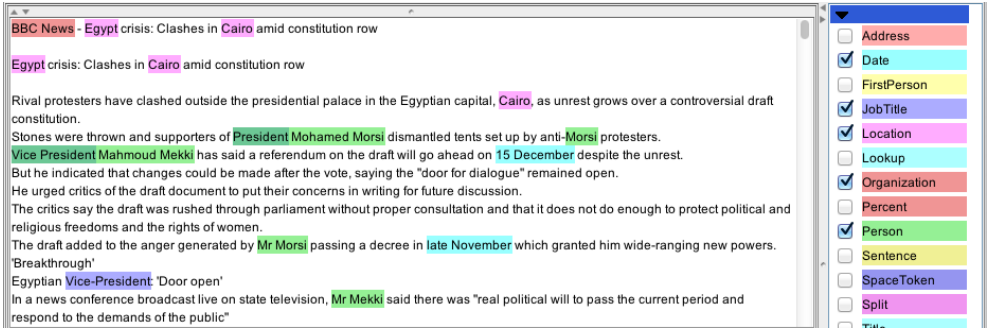
René Witte Bahar Sateli

Semantic Software Lab
Concordia University
Montréal, QC, Canada

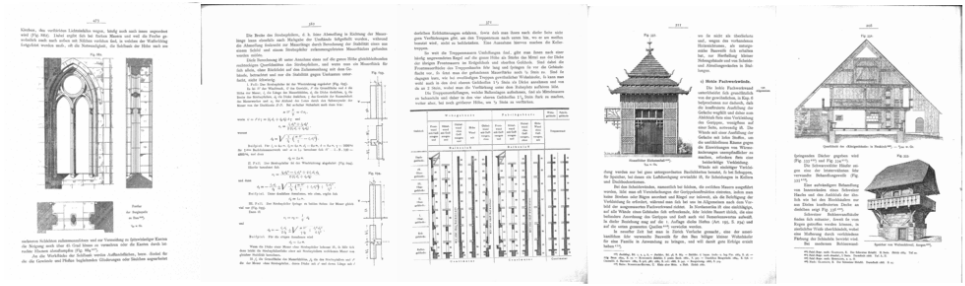
SMWCon Spring 2014

Natural Language Processing (NLP)

- A branch of Artificial Intelligence
  - uses various techniques to process content written in natural language
- Multitude of NLP techniques
  - Named Entity Recognition
  - Question-Answering
  - Summarization
- Various NLP tools (e.g. GATE, UIMA, . . .)
  and APIs (e.g. OpenCalais, AlchemyAPI, . . .)



Cultural Heritage Data Management



Large unstructured corpora

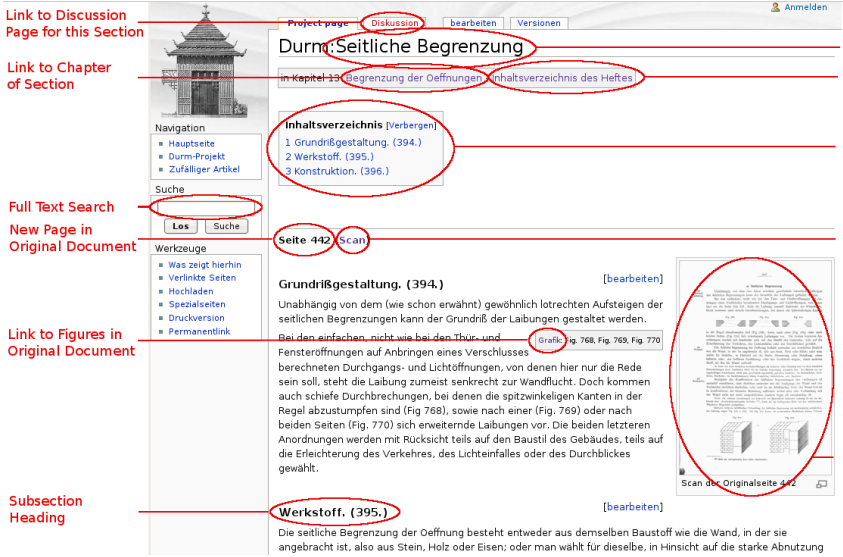
- Outdated terms, style of writing, huge amount, no categorization or assessment
- Comparing and evaluating with current content

Wiki Version of 19th-century documents

The Durm project

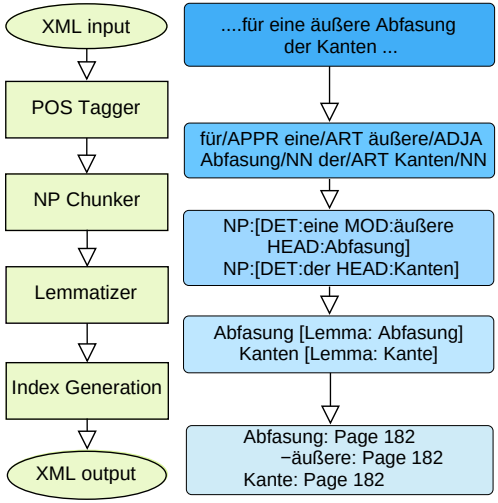
- Digitized version of a historical encyclopedia for architecture
- Provides automatic index generation

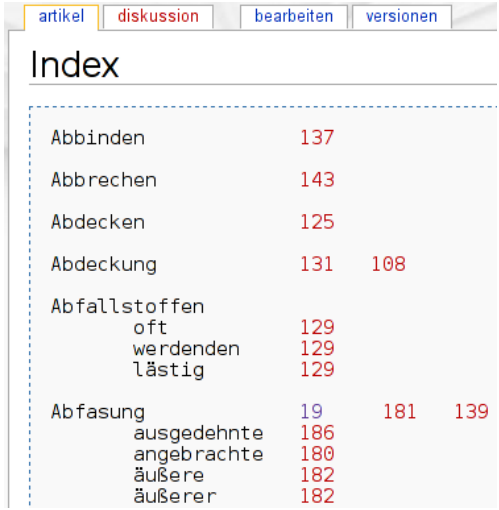
The first Wiki-NLP Experiments (ca. 2004)



Converting a Historical Architecture Encyclopedia into a Semantic Knowledge Base [WKKL10]

Back-of-the-Book Index Generation





Opportunities for NLP in Wikis

What could NLP provide for wikis?

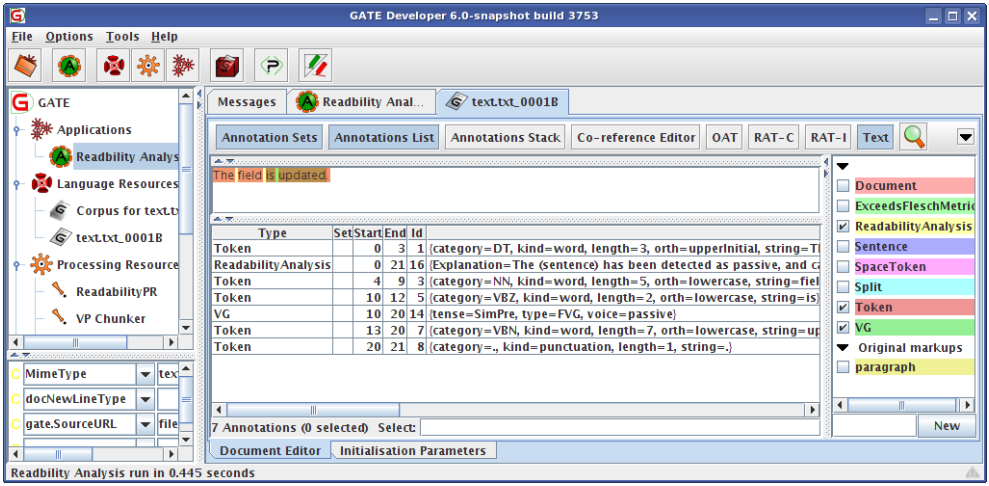
- Detect various defects in its content
  e.g., spelling mistakes, ambiguities, readability issues
- Extract entities that are relevant to a user's interest or context
  e.g., extract all person names mentioned in the wiki
- Formally model the knowledge contained inside the wiki
  e.g., generate Semantic MediaWiki markup from unstructured wiki text
  `[[hasType::Enzyme|Xylanase]]`
- Offer searching for content beyond keyword-based approaches
  e.g., find all articles containing an enzyme name
- Generate its own content
  e.g., create summaries from long articles
- Help structuring a wiki
  e.g., suggest terms to link to Wikipedia

WikiSym 2007: Connecting Wikis and Natural Language Processing Systems [WG07]

General Architecture for Text Engineering (GATE)

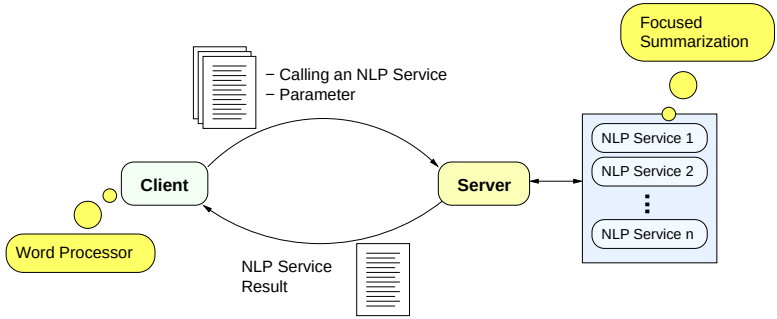
Open Source Text Analytics

Mature framework, more than 10 years of development
Development team at University of Sheffield, UK (gate.ac.uk)

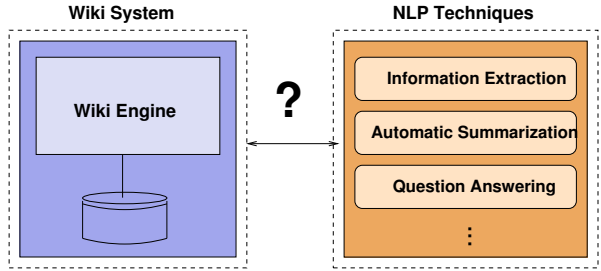


Semantic Assistants

- Service-oriented Architecture (SOA) [WG08]
- Publishes various NLP pipelines as W3C Standard Web services
- Developed by Semantic Software Lab (http://www.semanticsoftware.info)
- Client plug-ins for OpenOffice, Eclipse, MediaWiki, Liferay, Android
- Open source framework (http://www.semanticassistants.com)



Problem Statement



Solve common problems

- Wikis' Loose Structure
- Information Overload

Introduce new features

- Enable Human-AI collaboration
- Bring semantics to wiki content

Solution

Semantic Assistants Wiki-NLP Integration

- See [SW12] and http://www.semanticsoftware.info/semantic-assistants-wiki-nlp

Requirements Analysis

Three perspectives:

1. Wiki End-user Requirements
   - From laypersons to technical employees
   - Different cognitive backgrounds
2. Wiki Developer Requirements
   - Engine developers
   - Wiki administrators
3. System Requirements

Requirements Analysis

NLP Service Independence

Run any NLP pipeline on wiki content

Seamless Integration

No context switching between wiki and the NLP framework

Easy Deployment

No major modification on the wiki engine

Collection-based Analysis

Multi-document analysis of wiki content

Other Requirements

Read Content from Wiki, Write Results to Wiki, Flexible Response Handling, Proactive Service Execution, ...

Requirements Definitions

Seamless Integration

Using the Wiki-NLP integration must not largely deviate from the established usage patterns of wikis

Easy Deployment

Wiki-NLP integration must not require major changes to the wiki engine or to the means to access the wiki

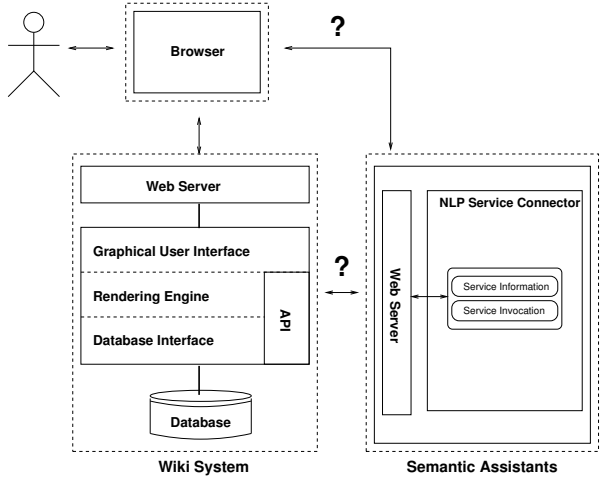
Collection-based Analysis

Users must be able to collect pages of interest and run an NLP service on the collection at once

Design Decisions

Design Questions

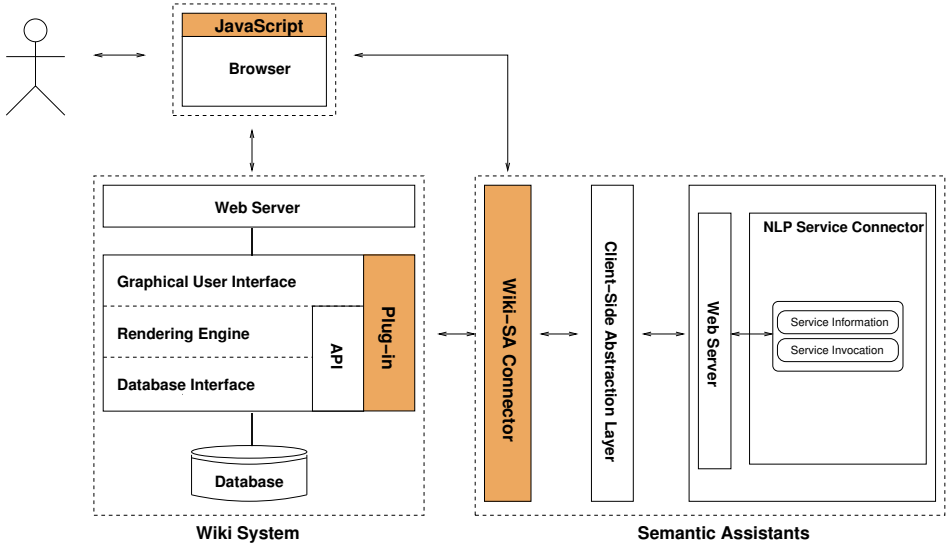
- Where should the Wiki-NLP integration be implemented?
- How can the system interactions be realized?



Design Decisions (II)

Collaborative approach

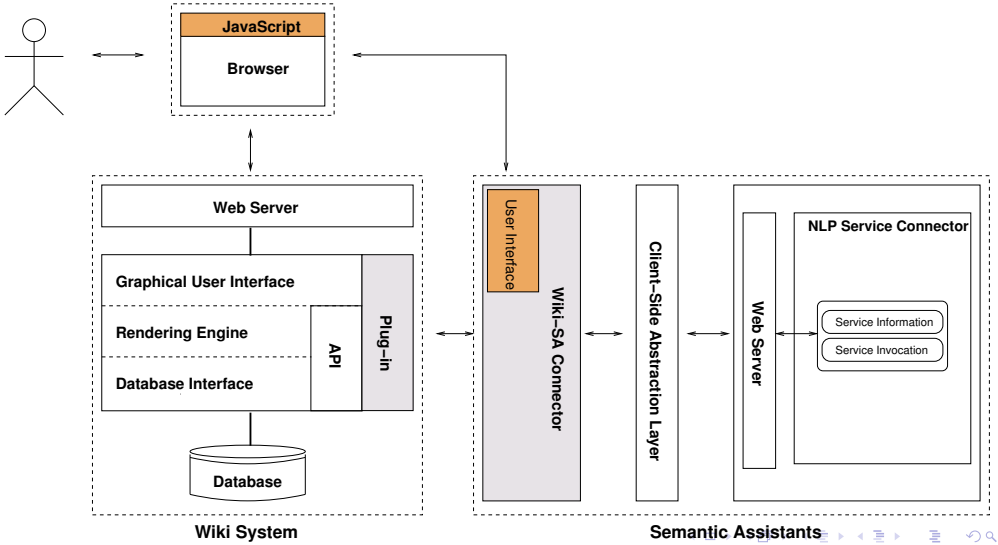
Division of responsibilities based on each system capabilities



User Interaction

Dynamic User Interface

Generating the user interface on-the-fly



Dynamic User Interface

Requirements

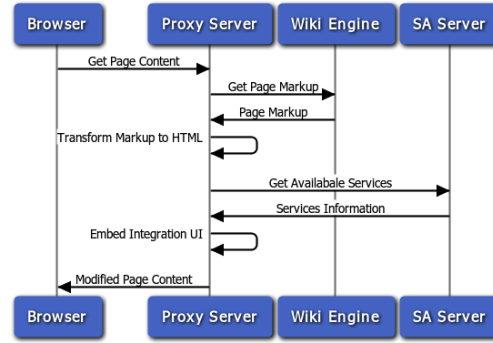
- Seamless Integration
- Easy Deployment

Solution

Proxy the wiki page request and embed the user interface on-the-fly

Advantages

- Wiki-independent
- No deployment required
- Seamless environment



Dynamic User Interface

Requirement

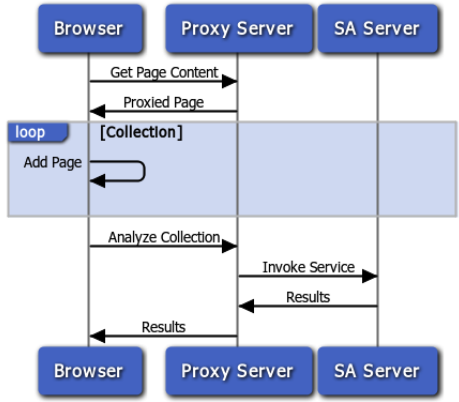
Collection-based Analysis

Solution

Preserve user's selection in the browser before the actual service invocation (e.g., cookies)

Advantages

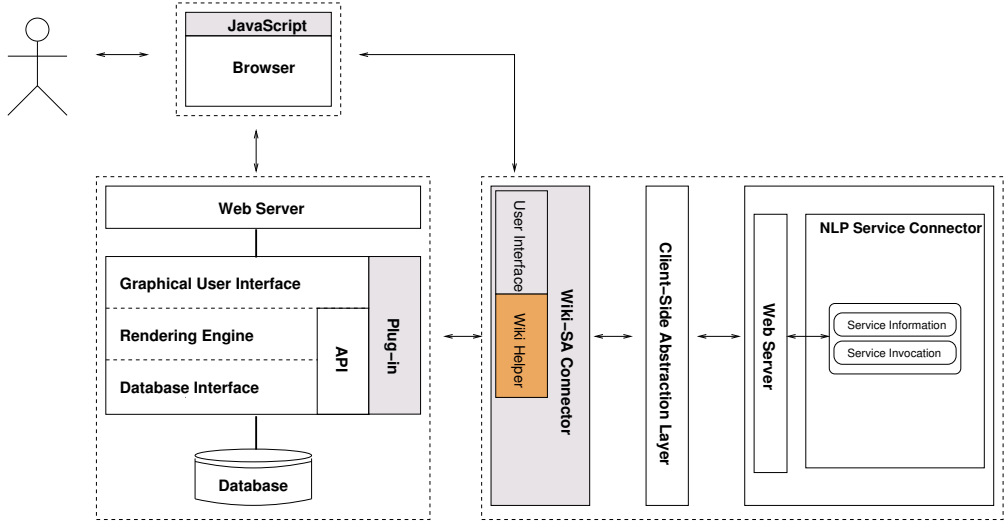
- Multi-document Analysis
- Increased efficiency



Wiki Communication

Wiki Bot Frameworks

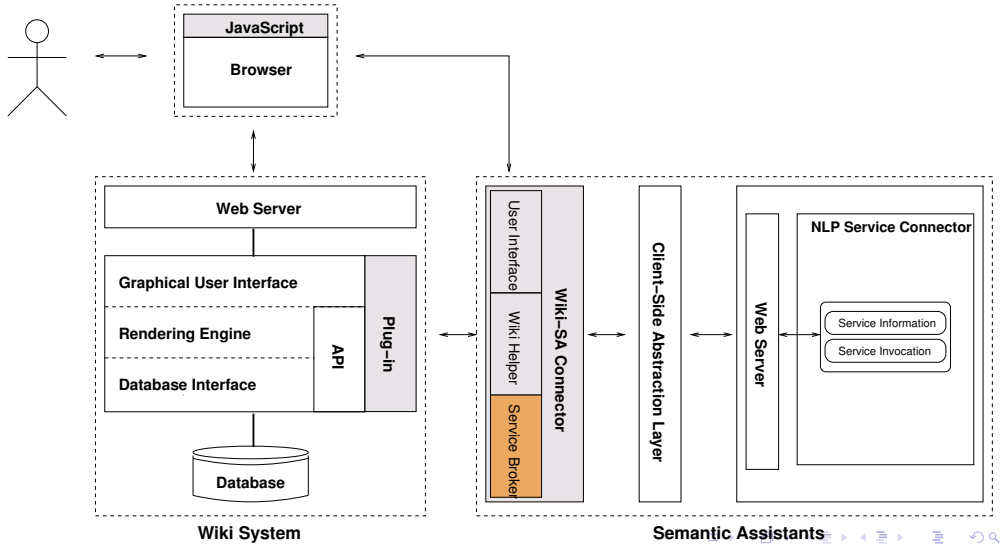
Accessing wiki content via its API with a higher level of abstraction



Service Invocation

Service Broker Component

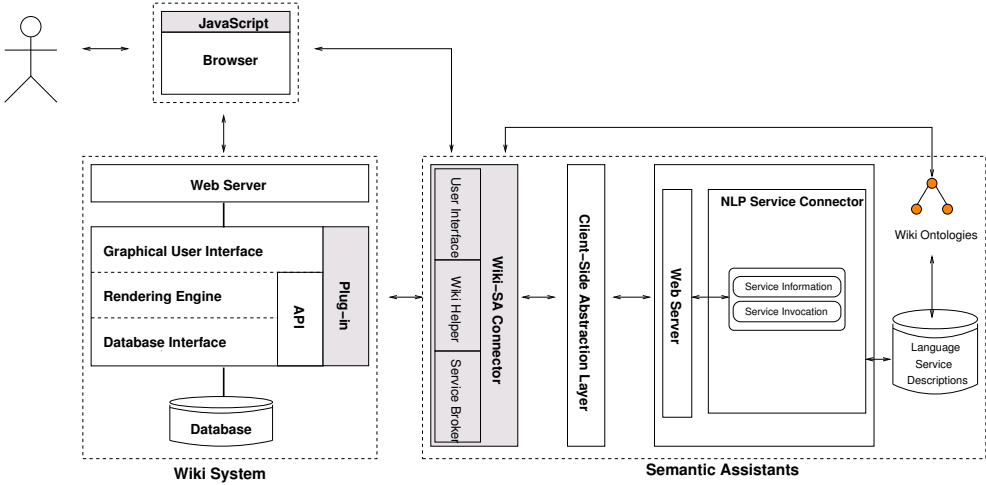
Delegating service invocation requests to the Semantic Assistants server



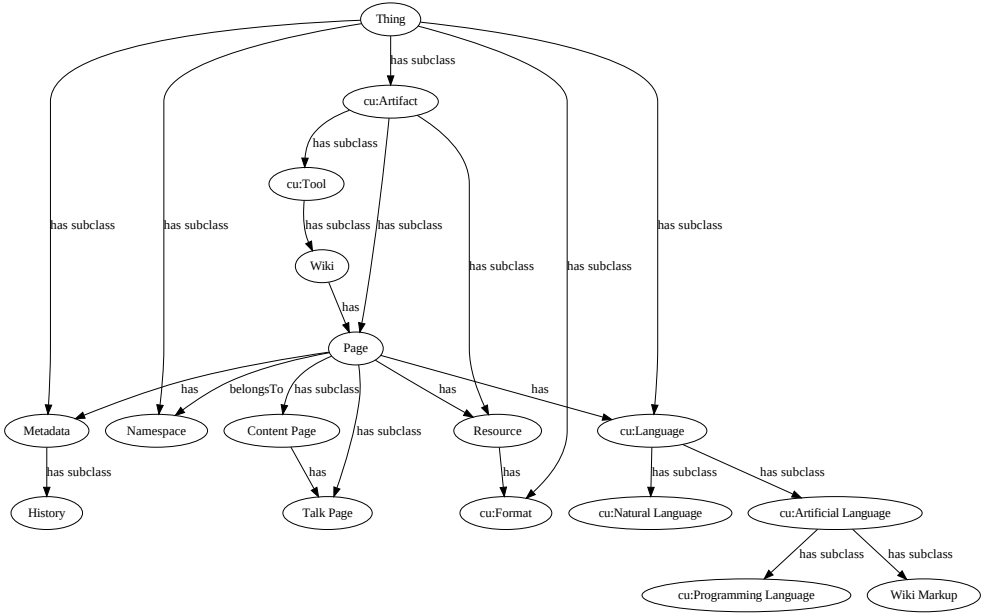
Wiki Independence

Semantics-based Architecture

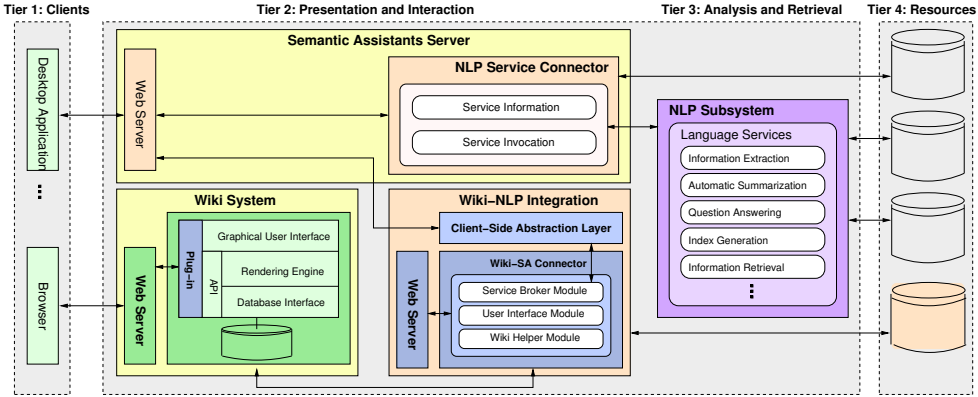
A repository of formalized descriptions of wikis written in OWL



Wiki Upper Ontology



Merged System Architecture



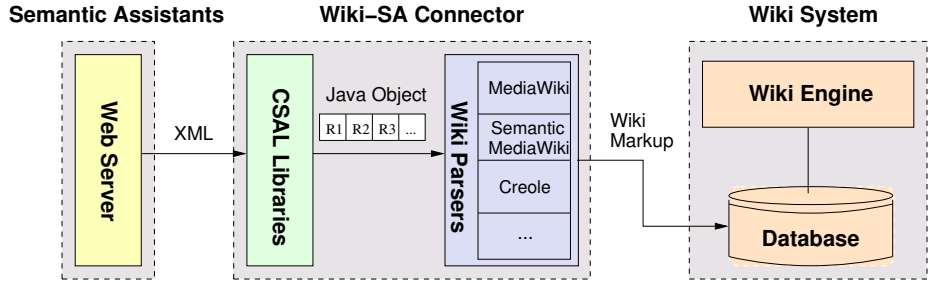
The new Wiki-NLP integration module in the Semantic Assistants architecture

Transformation of Results

From the Semantic Assistants server response to wiki markup

Sample Service Invocation Response

```
<saResponse>
 <annotation type="Location" annotationSet="Annotation" isBoundless="false">
     <document url="http://localhost/wiki/sample_page">
       <annotationInstance content="Canada" start="16" end="22">
          <feature name="locType" value="country"/>
       </annotationInstance>
     </document>
 </annotation>
</saResponse>
```



Presentation of Results

Templating mechanism

Separating data model from its presentation

```
1 {| class="wikitable" style="height:50px"
2 ! width="200" | Content
3 ! width="80" | Type
4 ! width="50" style="text-align: center;" | Start
5 ! width="50" style="text-align: center;" | End
6 ! Features
7 |– valign="top" |{{{content}}} | style="text-align: center;" | ((Property:{{{type}}}|{{{type}}})) |
      style="text-align: center;" | {{{start}}} | style="text-align: center;" | {{{end}}} | {{{
      features}}}
8 |}
```

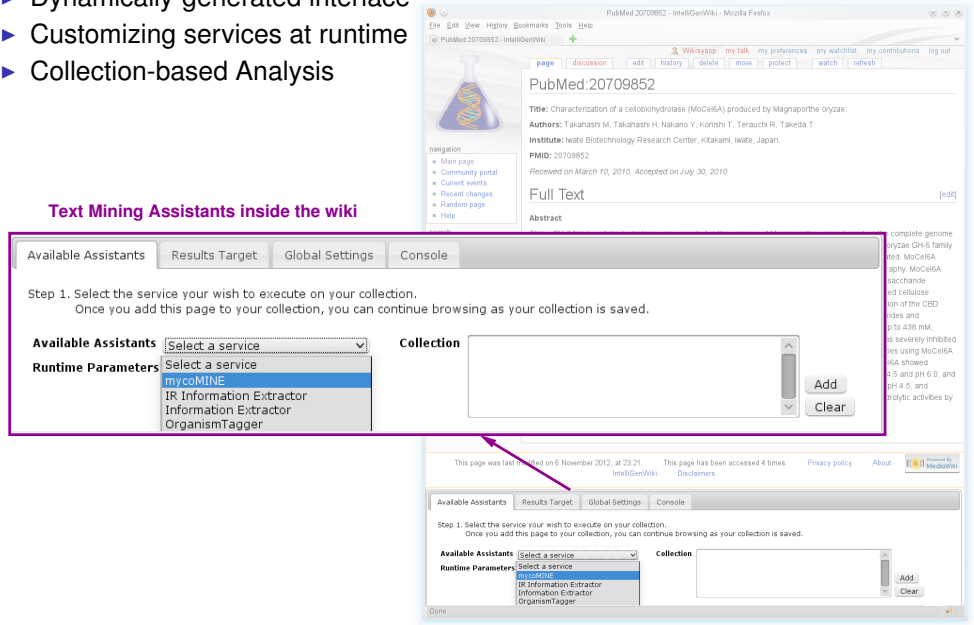
```
1 {{SemAssist–TableRow| content= Elizabeth Middleton | type=Person | start = 236 | end = 255 | features =
      gender:female}}
```

Content	Type	Start	End	Features
Elizabeth Middleton	Person	236	255	▪ gender: female

The NLP Interface

The NLP user interface can now offer various text mining services

- Dynamically-generated interface
- Customizing services at runtime
- Collection-based Analysis



Example: ANNIE NLP Service Results

BP Oil Spill

On 20 April 2010, a semi-submersible exploratory offshore drilling rig in Mexico exploded after a blowout. The US Government named British Petroleum, a London headquartered company, the responsible party. Eventually, the company stopped the flow of crude by shutting off pipes that were funneling some of the oil to ships on the surface on 14 July.

Following the oil spill the company stock fell in half and there were speculations in the press, guided by the commentary of Fred Lucas, that there would be a takeover of the company, focusing on possible bids from Exxon and Shell at a presumed price of £88 billion.

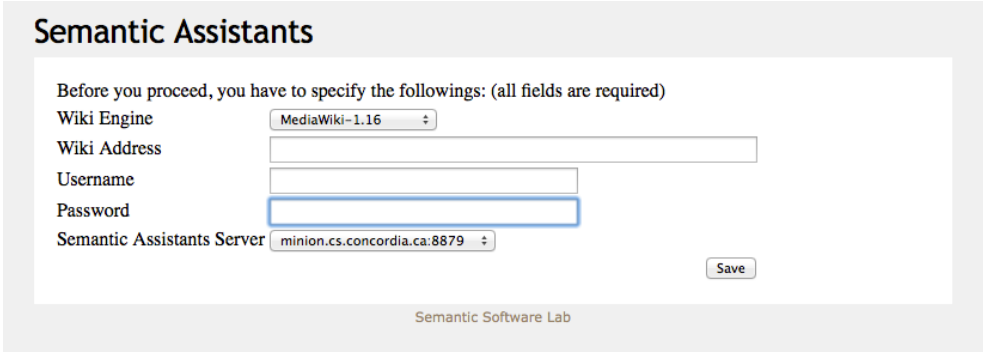
Information Extractor on BP_Oil_Spill (View)

Content	Type	Start	End	Features
20 April 2010	Date	3	16	▪ kind: date
14 July	Date	340	347	▪ kind: date
No annotations found	Address			
£88 billion	Money	603	614	▪ kind: number
Fred Lucas	Person	474	484	▪ gender: male
US Government	Organization	111	124	▪ orgType: government
British Petroleum	Organization	131	148	▪ orgType: company
Exxon	Organization	564	569	▪ orgType: company
Shell	Organization	574	579	▪ orgType: company
Mexico	Location	74	80	▪ locType: city
London	Location	152	158	▪ locType: city
No annotations found	Percent			

Wiki-NLP: Initial Configuration

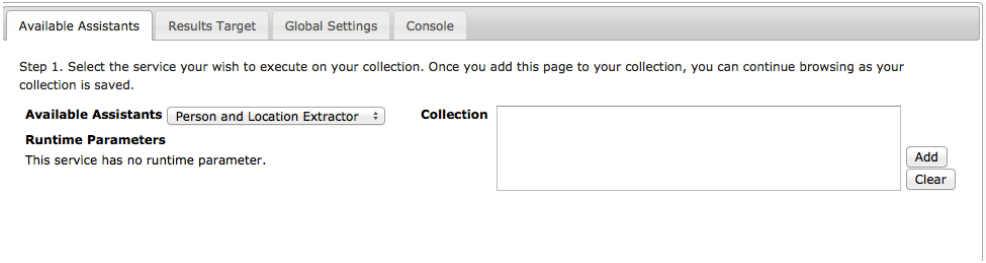
We recommend you use Firefox for the hands-on exercises.

1. Browse to http://hostname/wikinlp
2. Login with the following credentials
   - Username: ********
   - Password: ********
3. Click on the “Semantic Assistants” link on the left
4. Configure the Wiki-NLP integration with the same credentials as above



Wiki-NLP: Information Extraction

1. Who's your favorite celebrity or scholar? Pick one.
2. Create a sample page with his/her mini-biography (Copy-Paste from Wikipedia)
3. Click on the "Semantic Assistants" link on the left
4. Choose a service from the "Available Assistants"
5. Add the page to your "Collection"
6. Go to the "Results Target" tab, investigate the options and click "Run Service"
7. When finished, click on the "Page" tab on top to refresh the page and see the results



Wiki-NLP: Semantic Queries

1. In the page you created, type the following query to find all the “Person” entities:

```
{{#ask: [[hasType::Person]]
 |?Person= Entities Found
 |format=table
 |headers=plain
 |default=No pages found!
 |mainlabel=Page Name
}}
```

   (Copy-paste from the “Sample Semantic Query” on the Main Page)

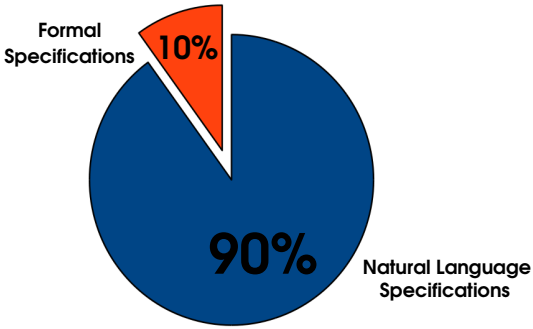
2. Discover all other persons mentioned in the wiki!

Requirements Engineering (RE) Tool Support

Wiki Systems for Software Requirements Engineering

- Collaborative environment for elicitation
  - Alternative to existing tools, better support, cost
  - Discussion media: sharing, traceability, guidelines
- Moderator
  - Channel stakeholders' contribution: templates, assignments, overviews
  - Indicators analysis for conflicts detections
- Requirements refinements
  - Reviews
  - Extension to capture additional Requirements (NFR)

Requirements in Practice



Industry distribution of SRS [MFI04]

Issues

- Natural Language SRS are fragile & difficult to maintain
- Limited tool support for automatic analysis

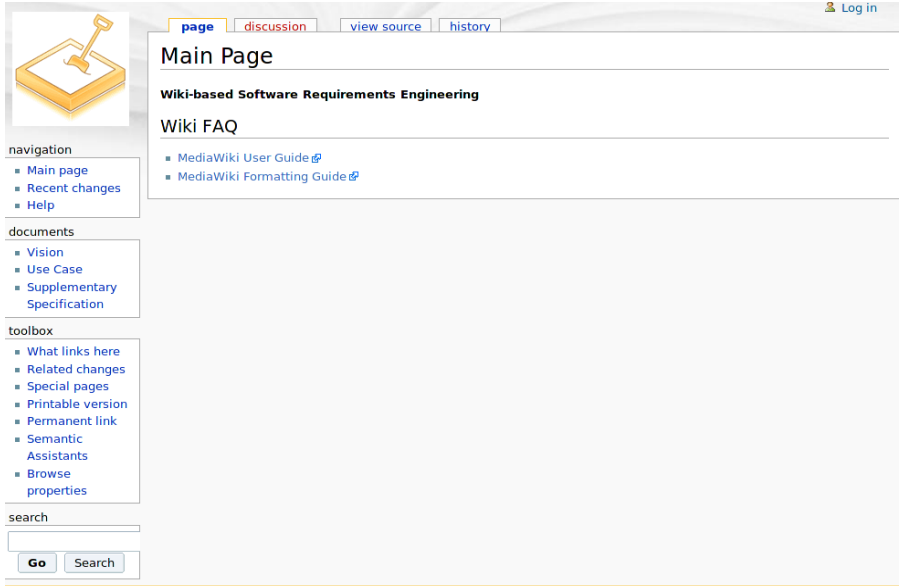
RE Tool Support

Wiki Improvements for RE

- Existing Wikis need further support for RE (added using plug-in architecture)
  - Improved page linking: automation, better contribution (traceability)
  - Lack of integration: connectors for other tools (e.g., CVS/Subversion) or Wiki integration into software development tools (Trac, Redmine)
  - Lack of semantics: add semantic layer (using ontologies), e.g., Semantic MediaWiki (SMW)

ReqWiki: A Semantic System for Collaborative Requirements Engineering

Open Source Project at Concordia's Semantic Software Lab



ReqWiki System

Goals

- Wiki-based user interface for easy collaboration
- Creation of structured, semantic markup
- Standard data format for web integration
- Integrated semantic support, like traceability link management

Platform

- MediaWiki Engine
- Semantic MediaWiki extension
- Semantic Forms, Templates
- Semantic Assistants for NLP support

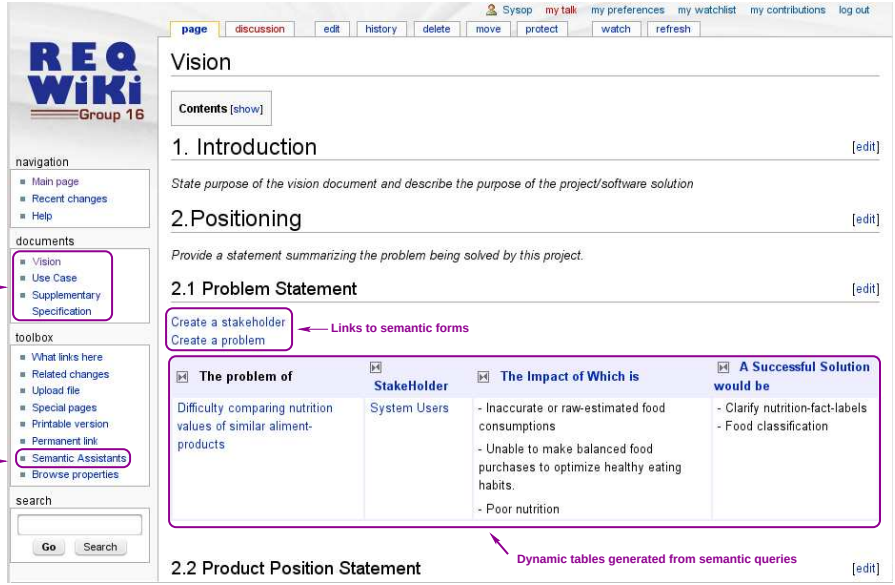
ReqWiki System Development

Transforming the 20th-century UP templates into a Semantic Wiki

- Idea: start from original *Word* document templates (which are based on the [Rational] Unified Process ([R]UP) industry standard)
- Transform them into a modern, Semantic Wiki version
- See [SAW13] and http://www.semanticsoftware.info/reqwiki

2.1. Problem Statement

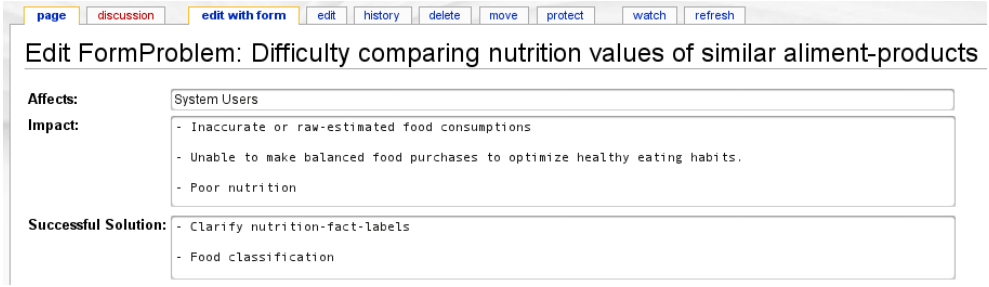
The problem of	*Difficulty comparing nutrition values of similar aliment-products*
Affects	*System Users*
The impact of which is	*- Inaccurate or raw-estimated food consumptions* *- Unable to make balanced food purchases to optimize healthy eating habits.* *- Poor nutrition*
A successful solution would be	*- Clarify nutrition-fact-labels* *- Food classification*



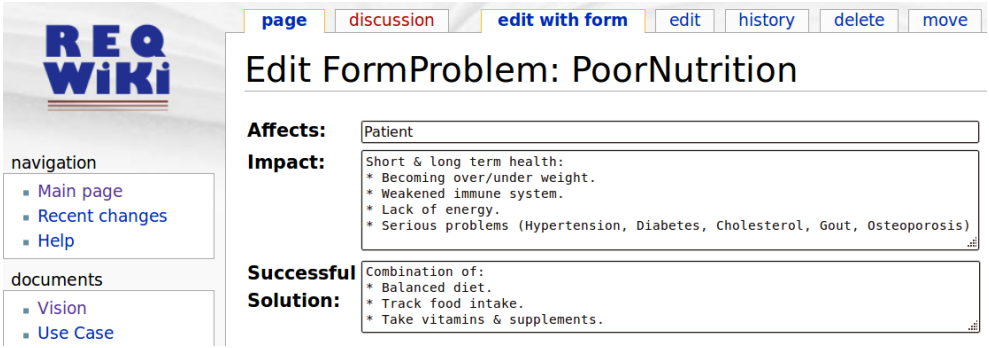
ReqWiki System

Using “Semantic Forms” Extension

MediaWiki Extension “Semantic Forms” allows to create forms that generate semantic markup



User Interface: Semantic Forms



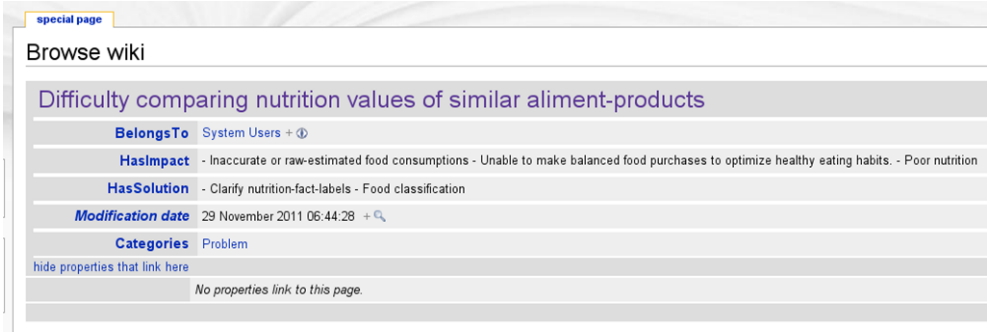
Features

- Structures content at finer granularities
- Auto-completion for referencing entities (e.g., actors)
- Automatically generates RDF markup for linking & querying

ReqWiki System

Semantic Markup

The Semantic Forms automatically generate semantic markup (RDF format)


special page

Browse wiki

Difficulty comparing nutrition values of similar aliment-products

BelongsTo	System Users +
HasImpact	- Inaccurate or raw-estimated food consumptions - Unable to make balanced food purchases to optimize healthy eating habits. - Poor nutrition
HasSolution	- Clarify nutrition-fact-labels - Food classification
Modification date	29 November 2011 06:44:28 +
Categories	Problem

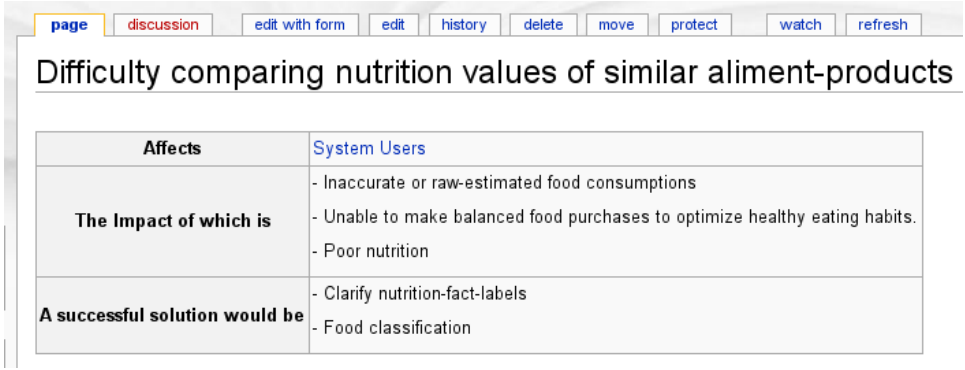
hide properties that link here

No properties link to this page.

ReqWiki System

Templates

To make the semantic markup readable by end users, templates are used:



page | discussion | edit with form | edit | history | delete | move | protect | watch | refresh

Difficulty comparing nutrition values of similar aliment-products

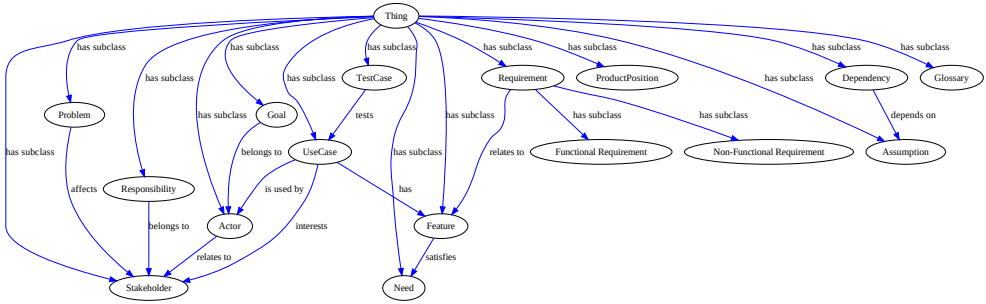
Affects	System Users
The Impact of which is	- Inaccurate or raw-estimated food consumptions - Unable to make balanced food purchases to optimize healthy eating habits. - Poor nutrition
A successful solution would be	- Clarify nutrition-fact-labels - Food classification

Semantic Model for SRS

Ontology

Content is structured using a Semantic Web ontology:

- Formally describe software artifacts and their components
- Used to model, connect and query SRS statements with ontology concepts

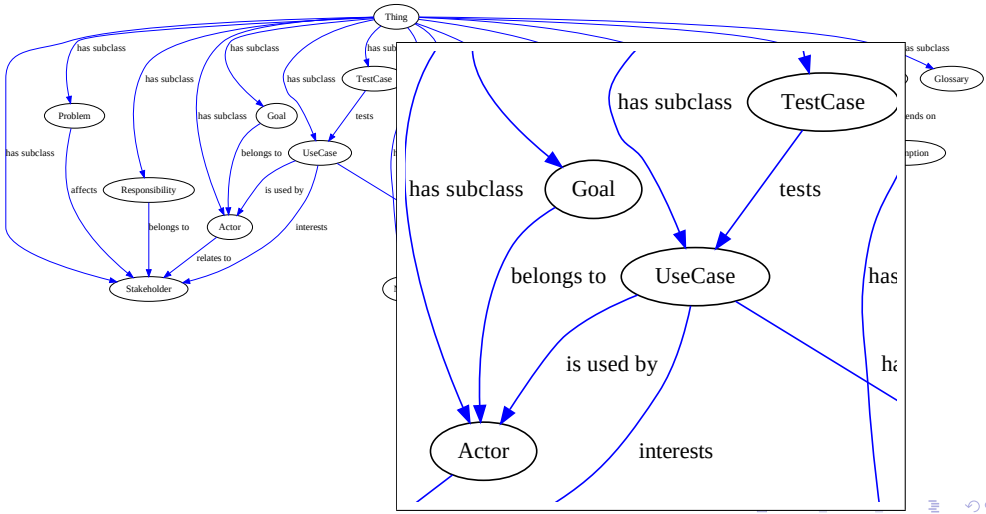


Semantic Model for SRS

Ontology

Content is structured using a Semantic Web ontology:

- Formally describe software artifacts and their components
- Used to model, connect and query SRS statements with ontology concepts



ReqWiki System

Embedded Queries

- Query for all stakeholders in the wiki:

```
1  {{#ask: [[<page>]]| [<param1> | <param2> ..] }}
2
3  Example:
4
5  {{#ask: [[Category:StakeHolder]]
6  | mainlabel=StakeHolder
7  | ?HasDescription= Description
8  | format= table
9  }}
```

User Interface: Embedded Queries

StakeHolder Summary [edit]

Create a responsibility

StakeHolder	Description
Developer	The person who is developing the application
Student	The person who is using the application
Teacher's Assistant	The person who is supervising a tutorial or lab

ReqWiki: Traceability

Automatic Traceability Support

- Traceability is concerned with interrelating various software artifacts
- Manually cross-referencing documents is time-consuming and error-prone
- Exploiting the semantic metadata in ReqWiki
- Supported traceability links types:
  1. Revision links
  2. Semantic links
  3. Query-based links

```
{{#ask: [[Category:Features]]
| ?BelongsTo=Need
| ? = Feature
| format= table
}}
```

User Needs versus Features

Need	Feature
Modify policy detail information	Alter policy information
Modify policy detail information	Query the status of policy information
Alteration of unit link product	Input conversion to unit investment
Alteration of unit link product	Modify conversion to unit investment
Alteration of unit link product	Query unit investment
Alteration of unit link product	Query unit price

ReqWiki System: Automatic Traceability Link Support

- Traceability Links can now all be automatically generated
- Tables are created with a combination of Queries & Templates
- Dead links (e.g., after deleting an entity) can also be automatically detected
- Like all other semantic metadata, RDF can be exported from the Wiki and consumed by other tools

Traceability Links

User Needs versus Features

Need	Feature
Food Consumption	Nutrition Fact

Features versus Use Cases

UseCases	Features
Collect Aliment-Intake	Nutrition Fact

Features versus Supplementary Requirements

Features	Functional Requirements
Nutrition Fact	Food Intake

Features	Non Functional Requirements
Nutrition Fact	Mean Time

Features	Supplementary Requirements
Nutrition Fact	Food Intake
Nutrition Fact	Mean Time

Use Cases to Test Cases

UseCase	TestCases
Collect Aliment-Intake	Aliment Testcase

ReqWiki: Content Analysis

Motivation

Most requirements are expressed in natural-language form. . .

Example Requirement

"*The system shall configure the frame-rate as specified by the camera provided during initialization & the frame-rate shall be capable of being automatically modified based on bandwidth.*"

Natural Language Processing

. . . but these requirements are prone to expensive-to-find defects.
Solution: use Natural Language Processing (NLP) to help users (see Requirements Quality Assurance, [vL09])

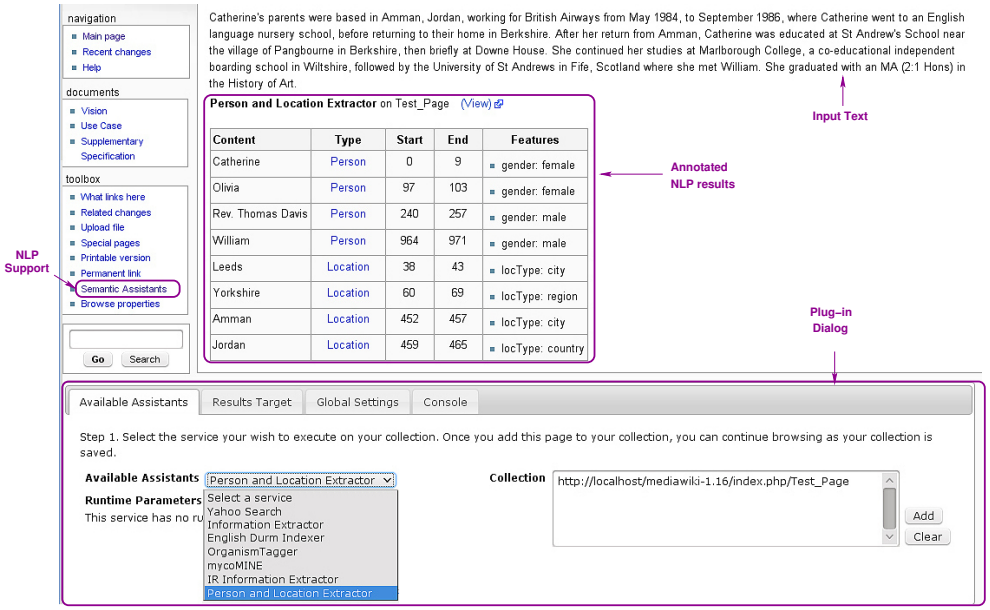
Text Mining Assistants

Natural-Language Requirements Defects

SRS defects addressable with ReqWiki [Lap09, vL09]:

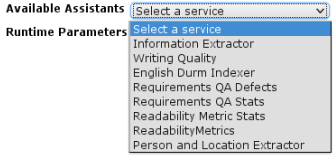
- Spelling & Grammar
- Incompleteness
- Contradiction
- Ambiguity
- Unmeasurability
- Noise
- Over Specification
- Inadequacy
- Unintelligibility
- Poor Structuring
- Unfeasibility
- Options
- ...

User Interface: Wiki-NLP Plug-in



ReqWiki NLP Services

- Various NLP services



- SRS defects addressable
  - Spelling & Grammar
  - Incompleteness
  - Ambiguity
  - Poor Structuring
  - Passive voice
  - . . .
- Automatically index the SRS
  - back-of-the-book style
  - complement the glossary
  - helping domain analysis

UC/Manage Tasks

Description	The manager receives a customer service request. The manager directs the operation for creating, updating, deleting and querying a task. Some operations use either the automatic or manual task assignation functionality that were defined in the Supplementary Specification document.
Level	user-goal
Primary Actor	A / Manager
StakeHolders	Manager, Senior technician, Junior technician
Pre-Conditions	The manager must be identified and authenticated in the application
Success end condition	The task is created and assigned to the technicians with status Assigned. The tasks is updated and assigned to the technicians with status Assigned. The task is queried. The task is deleted.
Failure end condition	The task is created with status Submitted.
Features	Manage Task

Writing Quality on UC/Manage_Tasks (View)

Content	Type	Start	End	Features
were defined	AtD	236	248	▪ problem: Passive voice ▪ suggestion: -
must be	AtD	434	441	▪ problem: Passive voice ▪ suggestion: -
is created	AtD	521	531	▪ problem: Passive voice ▪ suggestion: -
The tasks is	AtD	587	599	▪ problem: Subject Verb Agreement ▪ suggestion: The tasks are, The task is

Index Generation (through Noun Phrase Identification)

Automatically created “back-of-the-book index”

navigation
- Main page
- Recent changes
- Help

documents
- Vision
- Use Case
- Supplementary Specification

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Semantic Assistants
- Browse properties

page | discussion | edit | history | delete | move | watch | refresh

English Durm Indexer

```
activity        1. Introduction

amount  1. Introduction

appendix        9. Glossary

application     1. Introduction
        many    9. Glossary
        other   1. Introduction

availability    9. Glossary

belongsto       1. Actor-Goal List
        Goal    1. Actor-Goal List

bulk    9. Glossary

capability      9. Glossary

capture
        Supplementary Specification     9. Glossary

case
        use     9. Glossary

category        9. Glossary  1. Actor-Goal List  1. Introduction

cycle
        task    1. Introduction

data    9. Glossary
```

User Study I – Effectiveness

Research Question

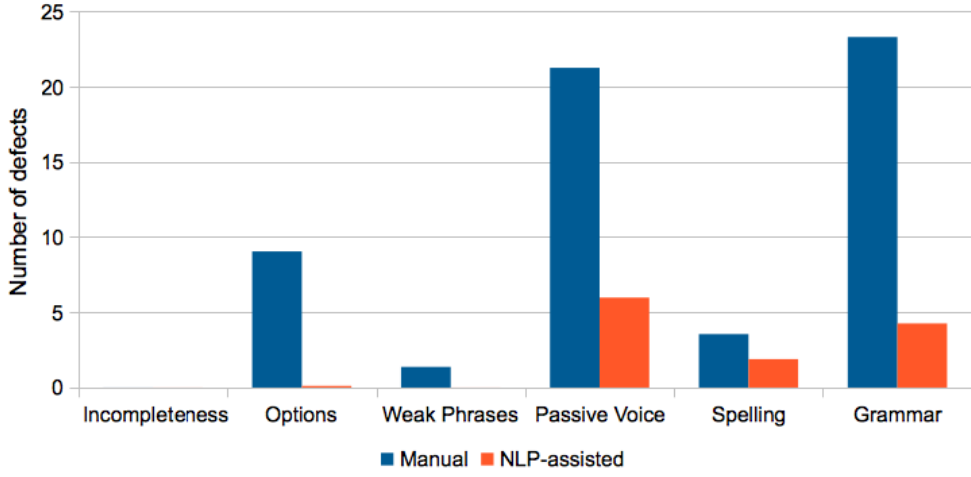
Can text mining assistants help to improve requirements specifications?

Methodology

User study with two software engineering classes at Concordia University

- **Goal:** Identifying defects in manual vs. NLP-assisted requirements specifications
- **NLP Services:** Spell checking, Readability Analysis, Passive Voice Detection, ...
- **Measure:** Average number of defects found in the two assignment revisions
- **Method:** Comparison of manual vs. NLP-assisted quality assurance

User Study I – Effectiveness: Results



Conclusion

ReqWiki NLP capabilities were indeed effective to significantly reduce SRS defects.

User Study II – Usability

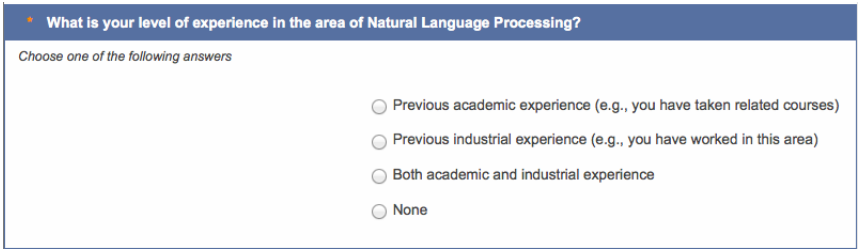
Research Question

How much NLP background do users need in order to use semantic capabilities?

Methodology

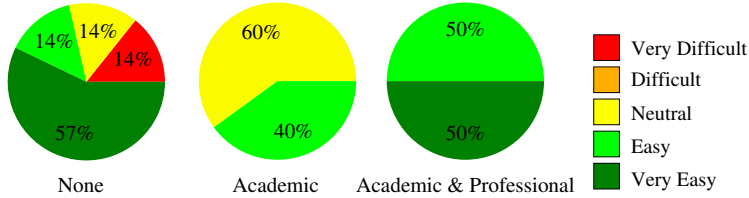
Same scenario as User Study I; Anonymized questionnaire asking participants about:

1. Their proficiency level in NLP
2. ReqWiki ease-of-use



User Study II – Usability: Results

Students' Questionnaire Feedback

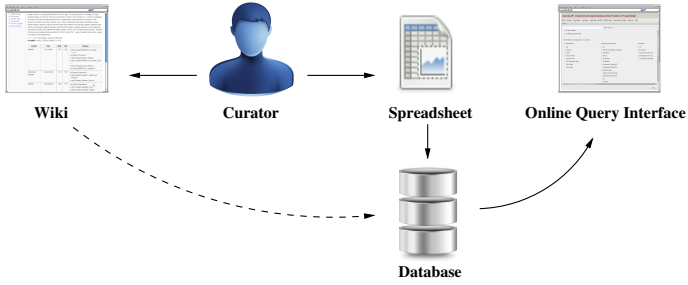


Conclusion

Concrete NLP background is *not* required to make use of sophisticated semantic support provided in ReqWiki.

Wikis as Knowledge Management Platforms

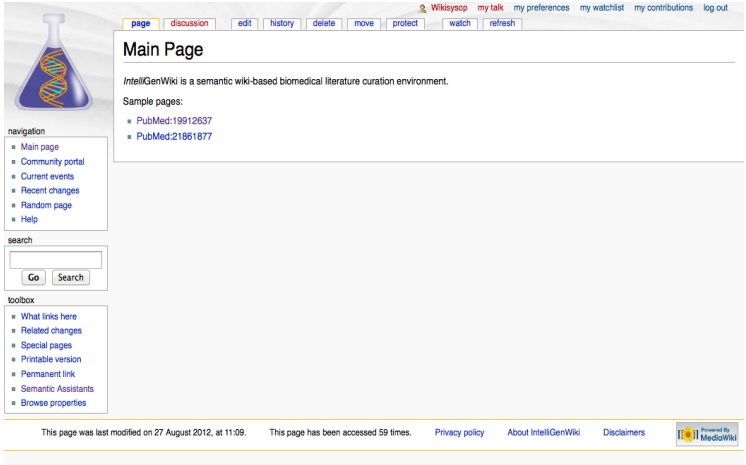
- Biomedical Literature Curation
  - Finding and extracting relevant knowledge from the domain literature
  - Manually refining and updating bioinformatics databases



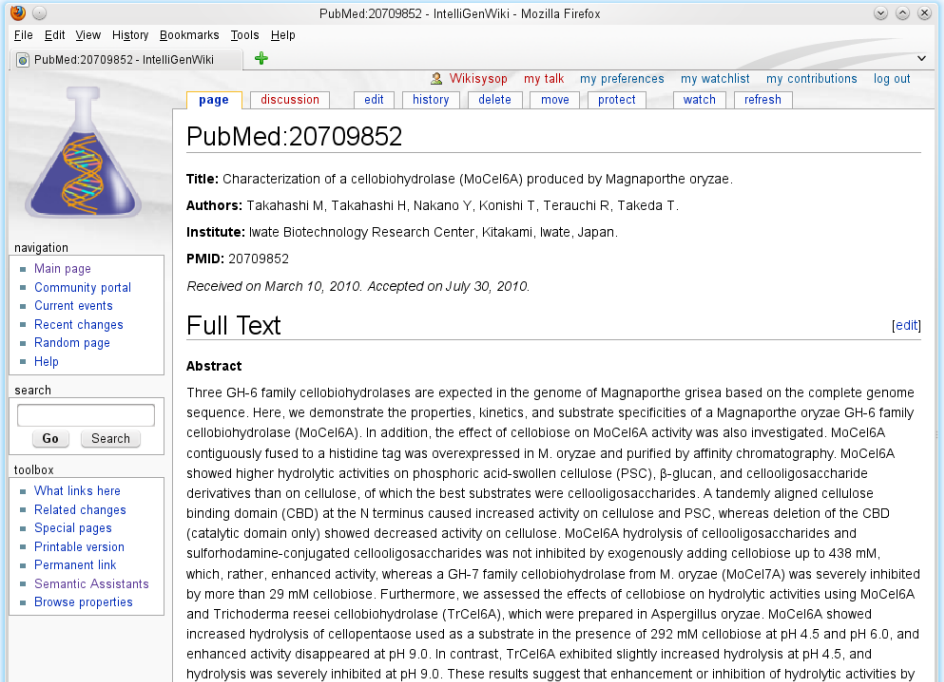
- Manual literature curation is...
  - Expensive → requires domain experts
  - Labour-intensive → ever growing amount of scientific publications
  - Error-prone → critical knowledge can be easily missed

IntelliGenWiki

- An intelligent semantic wiki for life sciences [SMB+12]
- Integrated bio-related NLP services for literature curation
- Offers basic semantic entity retrieval
- See http://www.semanticsoftware.info/intelligenwiki



IntelliGenWiki: User Interface



Bio-NLP: Information Extraction

- Automatically extracting knowledge from papers
- Various IE services
  - OrganismTagger
  - Open Mutation Miner
  - Idots
- Enrichment of literature content with semantic markup

Example:

[[hasType::Enzyme|cellobiohydrolase]]

severely inhibited at pH 9.0. These results suggest that enhancement or inhibition of hydrolytic activities by cellobiose is dependent on the reaction mixture pH.

PMID: 20709852 [PubMed - indexed for MEDLINE] PMCID: PMC2950481 Free PMC Article

mycoMINE on PMID:_20709852_Abstract (View)

Content	Type	Start	End	Features
cellobiohydrolase	Enzyme	103	120	▪ enzyme_alias: cellobiohydrolase ▪ BRENDA_SystematicName: oligoxyloglucan reducing-end cellobiohydrolase ▪ BRENDA_ECNumber: 3.2.1.150 ▪ abbreviation_alias: - ▪ google_search: http://www.google.com/search?q=cellobiohydrolase ▪ BRENDA_RecommendedName: oligoxyloglucan reducing-end-specific cellobiohydrolase ▪ SwissProt_ID: - ▪ BRENDA's page: http://www.brenda-enzymes.org/php/result_flat.php4?ecno=3.2.1.150
Magnaporthe oryzae	Organism	143	161	▪ NCBI_Taxonomy_WebPage: http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=318829&mode=info ▪ organism_scientific_name: Magnaporthe oryzae ▪ organism_alias: Magnaporthe oryzae ▪ google_search: http://www.google.com/search?q=Magnaporthe+oryzae ▪ NCBI_Taxonomy_ID: 318829

Found Entity | Entity Type | Entity Location | NLP–Provided Additional Information

IntelliGenWiki: NLP

- Extracted biomedical entities through NLP pipelines

These results suggest that enhancement or inhibition of hydrolytic activities by cellobiose is dependent on the reaction mixture pH.

PMID: 20709852 [PubMed - indexed for MEDLINE] PMCID: PMC2950481 Free PMC Article
mycoMINE on PMID:_20709852_Abstract (View)

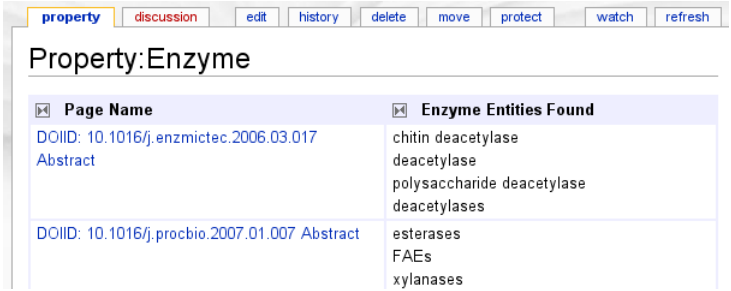
Content	Type	Start	End	Features
In contrast, TrCel6A exhibited slightly increased hydrolysis at pH 4.5, and hydrolysis was severely inhibited at pH 9.0.	pH	1847	1967	▪ pH_alias: In contrast, TrCel6A exhibited slightly increased hydrolysis at pH 4.5, and hydrolysis was severely inhibited at pH 9.0.
cellobiohydrolase	Enzyme	89	106	▪ enzyme_alias: cellobiohydrolase ▪ BRENDA_SystematicName: 4-beta-D-glucan cellobiohydrolase ▪ BRENDA_ECNumber: 3.2.1.91 ▪ abbreviation_alias: - ▪ google_search: http://www.google.com/search?q=cellobiohydrolase ▪ BRENDA_RecommendedName: cellulose 1,4-beta-cellobiosidase ▪ SwissProt_ID: O68438 ▪ BRENDA's page: http://www.brenda-enzymes.org/php/result_flat.php4?ecno=3.2.1.91

Semantic Entity Retrieval

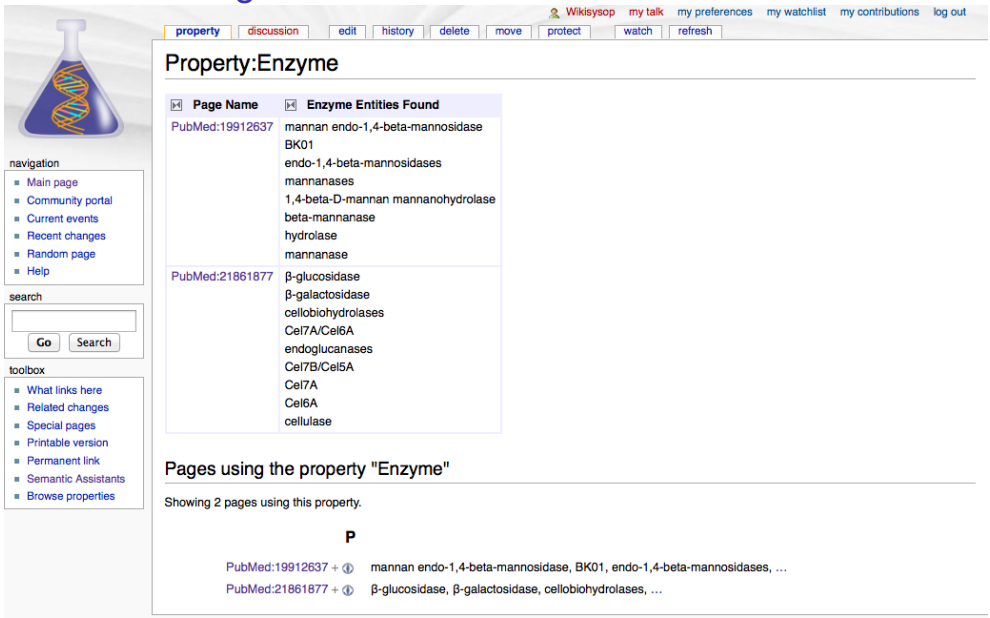
Semantic Queries

- e.g., *"Which papers in this wiki mention an enzyme entity in their text?"*

```
{{#ask: [[hasType::Enzyme]]
   | ?Enzyme = Enzyme Entities Found
   | format = table
   | headers = plain
   | default = No pages found!
   | mainlabel = Page Name
 }}
```



Semantic Navigation of NLP Results



User Study

- Is the integration of text mining assistants in a wiki environment actually effective?
- User study within the Genozymes project context (www.fungalgenomics.ca)
  - **Goal:** Identifying and characterizing fungal enzymes
  - **Dataset:** 30 documents
  - **Users:** 2 expert biocurators
  - **NLP Service:** mycoMINE
  - **Measure:** Time spent on curation
  - **Method:** Comparison against time spent on manual curation
- Results:

Average Curation Time

Abstract Selection		**Full Paper Curation**	
no support	IntelliGenWiki	no support	IntelliGenWiki
1 min.	0.3 min.	37.5 min.	30.6 min.

- Conclusion: IntelliGenWiki was indeed efficient and reduced the paper selection and curation time by almost **70%** and **20%**, respectively.

Introduction

The Wiki-NLP Integration

Hands-On

Application Examples

Conclusions

References

Conclusions

Add NLP capabilities to your wiki for a variety of use cases

- Allows use of existing text mining techniques in your wiki
- Find scenarios in which NLP assistance can be useful, e.g., Summarization
- Develop the actual NLP pipelines based on GATE
- Deploy the pipelines on a Semantic Assistants server
- Alternatively, use the existing text mining services in our public server
- Create machine-accessible information
- Add another party to the wiki user community: AI

Download and deploy the Wiki-NLP integration

- Deploy the Wiki-NLP servlet on a container, e.g., Tomcat or Jetty
- Install the Semantic Assistants MediaWiki extension on your wiki
- Configure the extension to point to the servlet endpoint

References I

Phillip A. Laplante.
Requirements Engineering for Software and Systems.
Auerbach Publications, 1st edition, 2009.

Luisa Mich, Mariangela Franch, and Pierluigi Inverardi.
Market research for requirements analysis using linguistic tools.
Requirements Engineering, 9:40–56, 2004.
http://dx.doi.org/10.1007/s00766-003-0179-8.

Bahar Sateli, Elian Angius, and René Witte.
The ReqWiki Approach for Collaborative Software Requirements Engineering with Integrated Text Analysis Support.
In *The 37th Annual International Computer Software & Applications Conference (COMPSAC 2013)*, page 405–414, Kyoto, Japan, July 22–26 2013. IEEE.
http://www.semanticsoftware.info/system/files/reqwiki-compsac13.pdf.

Bahar Sateli, Marie-Jean Meurs, Greg Butler, Justin Powlowski, Adrian Tsang, and René Witte.
IntelliGenWiki: An Intelligent Semantic Wiki for Life Sciences.
In *NETTAB 2012*, volume 18 (Supplement B), pages 50–52, Como, Italy, 11/2012 2012. EMBnet.journal, EMBnet.journal.

References II


Bahar Sateli and René Witte.
Natural Language Processing for MediaWiki: The Semantic Assistants Approach.
In *The 8th International Symposium on Wikis and Open Collaboration (WikiSym 2012)*, Linz, Austria, 08/2012 2012. ACM.


Axel van Lamsweerde.
Requirements Engineering: From System Goals to UML Models to Software Specifications.
John Wiley & Sons, 2009.


René Witte and Thomas Gitzinger.
Connecting Wikis and Natural Language Processing Systems.
In *WikiSym '07: Proceedings of the 2007 International Symposium on Wikis*, pages 165–176, New York, NY, USA, 2007. ACM.


René Witte and Thomas Gitzinger.
Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients.
In *3rd Asian Semantic Web Conference (ASWC 2008)*, volume 5367 of *LNCS*, pages 360–374, Bangkok, Thailand, Feb. 2–5, 2009 2008. Springer.

References III

René Witte, Ralf Krestel, Thomas Kappler, and Peter C. Lockemann.
Converting a historical architecture encyclopedia into a semantic knowledge base.
IEEE Intelligent Systems, 25(1):58–66, January/February 2010.