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, ...)

BBC News - Egypt crisis: Clashes in Cairo amid constitution row

Egypt crisis: Clashes in Cairo amid constitution row

Rival protesters have clashed outside the presidential palace in the Egyptian capital, Cairo, as unrest grows over a controversial draft constitution.

Stones were thrown and supporters of President Mohamed Morsi dismantled tents set up by anti-Morsi protesters. Vice President Mahmoud Mekki has said a referendum on the draft will go ahead on 15 December despite the unrest.

But he indicated that changes could be made after the vote, saying the “door for dialogue” remained open.

He urged critics of the draft document to put their concerns in writing for future discussion.

The critics say the draft was rushed through parliament without proper consultation and that it does not do enough to protect political and religious freedoms and the rights of women.

The draft added to the anger generated by Mr. Morsi passing a decree in late November which granted him wide-ranging new powers. Breakthrough?

Egyptian Vice-President: 'Door open'

In a news conference broadcast live on state television, Mr Mekki said there was “real political will to pass the current period and respond to the demands of the public”
Large unstructured corpora

▶ Outdated terms, style of writing, huge amount, no categorization or assessment
▶ Comparing and evaluating with current content
Adding NLP to (Semantic) MediaWiki
René Witte, Bahar Sateli

Introduction
Natural Language Processing
Motivation
Semantic Assistants
The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation
Hands-On
Configuration
Information Extraction with ANNIE
Semantic Queries
Application Examples
ReqWiki
Motivation
ReqWiki Design
ReqWiki Evaluation
IntelliGenWiki
Conclusions
References

Wiki Version of 19th-century documents

THE DURM PROJECT

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

\[\text{Wiki Version of 19th-century documents}\\ \text{The Durm project}\\
\begin{itemize}
  \item Digitized version of a historical encyclopedia for architecture
  \item Provides automatic index generation
\end{itemize}\

\begin{figure}
\centering
\includegraphics[width=\textwidth]{example.png}
\caption{Page from the Durm project wiki, showing the digitized version of a historical encyclopedia for architecture.}
\end{figure}
The first Wiki-NLP Experiments (ca. 2004)

Converting a Historical Architecture Encyclopedia into a Semantic Knowledge Base [WKKL10]
Back-of-the-Book Index Generation

**XML input**

```
....für eine äußere Abfasung der Kanten ...
```

**POS Tagger**

```
für/APPR eine/ART äußere/ADJA Abfasung/NN der/ART Kanten/NN
```

**NP Chunker**

```
NP:[DET: eine MOD: äußere HEAD: Abfasung] 
NP:[DET: der HEAD: Kanten]
```

**Lemmatizer**

```
Abfasung [Lemma: Abfasung]
Kanten [Lemma: Kante]
```

**Index Generation**

```
Abfasung: Page 182
–äußere: Page 182
Kante: Page 182
```

**Index**

```
Abbinden: 137
Abbrechen: 143
Abdecken: 125
Abdeckung: 131 108
Abfallstoffen oft: 129
werdenden: 129
lästig: 129
Abfasung: 19 181 139
ausgedehnte: 186
angegrachte: 180
äußere: 182
äußerer: 182
```
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*
  
  \[
  \text{[[hasType::Enzyme|Xylanase]]}
  \]

- 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*
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)
Adding NLP to (Semantic) MediaWiki

René Witte, Bahar Sateli

Introduction

The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation

Hands-On

Application Examples

Conclusions

References
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

Diagram showing the collaborative approach with JavaScript, Web Server, Database Interface, Graphical User Interface, Rendering Engine, Database, Wiki System, NLP Service Connector, Client-Side Abstraction Layer, and Wiki-SA Connector.
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
Adding NLP to (Semantic) MediaWiki

René Witte, Bahar Sateli

Introduction

Natural Language Processing
Motivation
Semantic Assistants

The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation

Hands-On
Configuration
Information Extraction with ANNIE
Semantic Queries

Application Examples
ReqWiki
Motivation
ReqWiki Design
ReqWiki Evaluation
IntelliGenWiki

Conclusions

References

Wiki Independence

Semantics-based Architecture
A repository of formalized descriptions of wikis written in OWL
Adding NLP to (Semantic) MediaWiki

René Witte,
Bahar Sateli

Introduction
Natural Language Processing
Motivation
Semantic Assistants
The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation
Hands-On
Configuration
Information Extraction with ANNIE
Semantic Queries
Application Examples
ReqWiki
Motivation
ReqWiki Design
ReqWiki Evaluation
IntelliGenWiki
Conclusions
References

Wiki Upper Ontology
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

```xml
<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

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

<table>
<thead>
<tr>
<th>Content</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elizabeth Middleton</td>
<td>Person</td>
<td>236</td>
<td>255</td>
<td>gender: female</td>
</tr>
</tbody>
</table>
The NLP Interface

The NLP user interface can now offer various text mining services:

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

Text Mining Assistants inside the wiki

Available Assistants

Collection

Step 1. Select the service you wish to execute on your collection. Once you add this page to your collection, you can continue browsing as your collection is saved.
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)
Introduction

The Wiki-NLP Integration

Hands-On

Configuration

Information Extraction with ANNIE

Semantic Queries

Application Examples

Conclusions

References
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!
Introduction

The Wiki-NLP Integration

Hands-On

Application Examples
- ReqWiki
  - Motivation
  - ReqWiki Design
  - ReqWiki Evaluation
- IntelliGenWiki

Conclusions

References
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

Issues

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

Industry distribution of SRS [MFI04]
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)
Adding NLP to (Semantic) MediaWiki

René Witte, Bahar Sateli

Introduction

Natural Language Processing
Motivation
Semantic Assistants

The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation

Hands-On
Configuration
Information Extraction with ANNIE
Semantic Queries

Application Examples
ReqWiki
Motivation
ReqWiki Design
ReqWiki Evaluation
IntelliGenWiki

Conclusions

References

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

<table>
<thead>
<tr>
<th>The problem of</th>
<th>Difficulty comparing nutrition values of similar aliment-products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affects</td>
<td>System Users</td>
</tr>
<tr>
<td>The impact of which is</td>
<td>- Inaccurate or raw-estimated food consumptions</td>
</tr>
<tr>
<td></td>
<td>- Unable to make balanced food purchases to optimize healthy eating habits.</td>
</tr>
<tr>
<td></td>
<td>- Poor nutrition</td>
</tr>
<tr>
<td>A successful solution would be</td>
<td>- Clarify nutrition-fact-labels</td>
</tr>
<tr>
<td></td>
<td>- Food classification</td>
</tr>
</tbody>
</table>
Adding NLP to (Semantic) MediaWiki

René Witte,
Bahar Sateli

Introduction

Natural Language Processing

Motivation

Semantic Assistants

The Wiki-NLP Integration

Requirements Analysis

System Design

Implementation

Hands-On

Configuration

Information Extraction with ANNIE

Semantic Queries

Application Examples

ReqWiki

Motivation

ReqWiki Design

ReqWiki Evaluation

IntelliGenWiki

Conclusions

References

1. Introduction

State purpose of the vision document and describe the purpose of the project/software solution

2. Positioning

Provide a statement summarizing the problem being solved by this project.

2.1 Problem Statement

Create a stakeholder

Create a problem

The problem of

Stake Holder

The Impact of Which is

A Successful Solution would be

Difficulty comparing nutrition values of similar aliment-products

System Users

- Inaccurate or raw-estimated food consumptions
- Unable to make balanced food purchases to optimize healthy eating habits:
  - Poor nutrition

- Clarify nutrition-fact-labels
- Food classification

Dynamic tables generated from semantic queries

Links to semantic forms

Top-level pages

Plug-in Semantic Assistants

Semantic Assistants Plug-in

Semantic Assistants

Hands-On Configuration

Information Extraction with ANNIE Semantic Queries Application Examples ReqWiki Motivation ReqWiki Design ReqWiki Evaluation IntelliGenWiki Conclusions References
ReqWiki System

Using “Semantic Forms” Extension

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

Edit Form

Problem: Difficulty comparing nutrition values of similar aliment-products

Affects: System Users

Impact:
- Inaccurate or raw-estimated food consumptions
- Unable to make balanced food purchases to optimize healthy eating habits.
- Poor nutrition

Successful Solution:
- Clarify nutrition-fact-labels
- Food classification
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)

<table>
<thead>
<tr>
<th>Difficulty comparing nutrition values of similar aliment-products</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BelongsTo</strong></td>
</tr>
<tr>
<td><strong>HasImpact</strong></td>
</tr>
<tr>
<td><strong>HasSolution</strong></td>
</tr>
</tbody>
</table>

**Modification date**: 29 November 2011 06:44:28

**Categories**: Problem

hide properties that link here

No properties link to this page.
**Templates**

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

---

**Difficultly comparing nutrition values of similar aliment-products**

<table>
<thead>
<tr>
<th>Affects</th>
<th>System Users</th>
</tr>
</thead>
</table>
| 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:

```wikipedia
{{{#ask: [[[<page>]]] [[<param1> | <param2> ..] ]}}
```

3  Example:

```wikipedia
{{#ask: [[[Category:StakeHolder]]
| mainlabel=StakeHolder
| ?HasDescription= Description
| format= table
}}
```
User Interface: Embedded Queries

StakeHolder Summary

Create a responsibility

<table>
<thead>
<tr>
<th>StakeHolder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer</td>
<td>The person who is developing the application</td>
</tr>
<tr>
<td>Student</td>
<td>The person who is using the application</td>
</tr>
<tr>
<td>Teacher's Assistant</td>
<td>The person who is supervising a tutorial or lab</td>
</tr>
</tbody>
</table>
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

>User Needs versus Features

<table>
<thead>
<tr>
<th>Need</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify policy detail information</td>
<td>Alter policy information</td>
</tr>
<tr>
<td>Modify policy detail information</td>
<td>Query the status of policy information</td>
</tr>
<tr>
<td>Alteration of unit link product</td>
<td>Input conversion to unit investment</td>
</tr>
<tr>
<td>Alteration of unit link product</td>
<td>Modify conversion to unit investment</td>
</tr>
<tr>
<td>Alteration of unit link product</td>
<td>Query unit investment</td>
</tr>
<tr>
<td>Alteration of unit link product</td>
<td>Query unit price</td>
</tr>
</tbody>
</table>

{#ask: [[Category:Features]]
| ?BelongsToNeed
| ? = Feature
| format= table
}
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
**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
- . . .
Adding NLP to (Semantic) MediaWiki

René Witte,
Bahar Sateli

Introduction
Natural Language
Processing
Motivation
Semantic Assistants
The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation
Hands-On
Configuration
Information Extraction with ANNIE
Semantic Queries
Application Examples
ReqWiki
Motivation
ReqWiki Design
ReqWiki Evaluation
IntelliGenWiki
Conclusions
References

User Interface: Wiki-NLP Plug-in

Catherine’s parents were based in Amman, Jordan, working for British Airways from May 1984, to September 1986, where Catherine went to an English language nursery school, before returning to their home in Berkshire. After her return from Amman, Catherine was educated at St Andrew’s School near the village of Pangbourne in Berkshire, then briefly at Downe House. She continued her studies at Marlborough College, a co-educational independent boarding school in Wiltshire, followed by the University of St Andrews in Fife, Scotland where she met William. She graduated with an MA (2:1 Hons) in the History of Art.

<table>
<thead>
<tr>
<th>Person and Location Extractor on Test_Page</th>
<th>(View)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Type</td>
</tr>
<tr>
<td>Catherine</td>
<td>Person</td>
</tr>
<tr>
<td>Olma</td>
<td>Person</td>
</tr>
<tr>
<td>Rev. Thomas Davis</td>
<td>Person</td>
</tr>
<tr>
<td>William</td>
<td>Person</td>
</tr>
<tr>
<td>Leeds</td>
<td>Location</td>
</tr>
<tr>
<td>Yorkshire</td>
<td>Location</td>
</tr>
<tr>
<td>Amman</td>
<td>Location</td>
</tr>
<tr>
<td>Jordan</td>
<td>Location</td>
</tr>
</tbody>
</table>

Step 1. Select the service your wish to execute on your collection. Once you add this page to your collection, you can continue browsing as your collection is saved.
**ReqWiki NLP Services**

- **Various NLP services**
  - Available Assistants:
    - Information Extractor
    - Writing Quality
    - English Durm Indexer
    - Requirements QA Defects
    - Requirements QA Stats
    - Readability Metric Stats
  - Runtime Parameters:
    - Person and Location Extractor

- **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**

<table>
<thead>
<tr>
<th>Description</th>
<th>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 assignment functionality that were defined in the Supplementary Specification document.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>user-goal</td>
</tr>
<tr>
<td>Primary Actor</td>
<td>A / Manager</td>
</tr>
<tr>
<td>StakeHolders</td>
<td>Manager, Senior technician, Junior technician</td>
</tr>
<tr>
<td>Pre-Conditions</td>
<td>The manager must be identified and authenticated in the application</td>
</tr>
<tr>
<td>Success end condition</td>
<td>The task is created and assigned to the technicians with status Assigned. The task is queried. The task is deleted.</td>
</tr>
<tr>
<td>Failure end condition</td>
<td>The task is created with status Submitted.</td>
</tr>
<tr>
<td>Features</td>
<td>Manage Task</td>
</tr>
</tbody>
</table>

**Writing Quality** on UC/Manage Tasks (View)

<table>
<thead>
<tr>
<th>Content</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>were defined</td>
<td>AID</td>
<td>236</td>
<td>248</td>
<td>problem: Passive voice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>suggestion: -</td>
</tr>
<tr>
<td>must be</td>
<td>AID</td>
<td>434</td>
<td>441</td>
<td>problem: Passive voice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>suggestion: -</td>
</tr>
<tr>
<td>is created</td>
<td>AID</td>
<td>521</td>
<td>531</td>
<td>problem: Passive voice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>suggestion: -</td>
</tr>
<tr>
<td>The tasks is</td>
<td>AID</td>
<td>587</td>
<td>599</td>
<td>problem: Subject Verb Agreement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>suggestion: The tasks are, The task is</td>
</tr>
</tbody>
</table>
Adding NLP to Semantic MediaWiki
René Witte, Bahar Sateli

Introduction

Natural Language Processing
Motivation
Semantic Assistants

The Wiki-NLP Integration
Requirements Analysis
System Design
Implementation

Hands-On
Configuration
Information Extraction with ANNIE
Semantic Queries

Application Examples
ReqWiki
Motivation
ReqWiki Design
ReqWiki Evaluation
IntelliGenWiki

Conclusions
References

Index Generation (through Noun Phrase Identification)
Automatically created “back-of-the-book index”
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

What is your level of experience in the area of Natural Language Processing?

Choose one of the following answers

- Previous academic experience (e.g., you have taken related courses)
- Previous industrial experience (e.g., you have worked in this area)
- Both academic and industrial experience
- None
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

PubMed:20709852

**Title:** Characterization of a cellulohydrolase (MoCel6A) produced by Magnaporthe oryzae.

**Authors:** Takahashi M, Takahashi H, Nakano Y, Konishi T, Teruchi R, Takeda T.

**Institute:** Iwate Biotechnology Research Center, Kitakami, Iwate, Japan.

**PMID:** 20709852

*Received on March 10, 2010. Accepted on July 30, 2010.*

**Abstract**

Three GH-6 family cellobiohydrolases are expected in the genome of Magnaporthe grisea based on the complete genome sequence. Here, we demonstrate the properties, kinetics, and substrate specificities of a Magnaporthe oryzae GH-6 family cellulohydrolase (MoCel6A). In addition, the effect of cellobiose on MoCel6A activity was also investigated. MoCel6A, contiguously fused to a histidine tag was overexpressed in M. oryzae and purified by affinity chromatography. MoCel6A showed higher hydrolytic activities on phosphoric acid-swollen cellulose (PSC), β-glucan, and cellobiose derivatives than on cellulose, of which the best substrates were celllobiose and cellobiose. A tandemly aligned cellulose binding domain (CBD) at the N terminus caused increased activity on cellulose and PSC, whereas deletion of the CBD (catalytic domain only) showed decreased activity on cellulose. MoCel6A hydrolysis of cellobiose and celllobiose was not inhibited by exogenously adding cellobiose up to 438 mM, which, rather, enhanced activity, whereas a GH-7 family cellobiohydrolase from M. oryzae (MoCel7A) was severely inhibited by more than 23 mM cellobiose. Furthermore, we assessed the effects of cellobiose on hydrolytic activities using MoCel6A and Trichoderma reesi cellobiohydrolase (TrCel6A), which were prepared in Aspergillus oryzae. MoCel6A showed increased hydrolysis of cellopentaose used as a substrate in the presence of 292 mM cellobiose at pH 4.5 and pH 6.0, and enhanced activity disappeared at pH 9.0. In contrast, TrCel6A exhibited slightly increased hydrolysis at pH 4.5, and hydrolysis was severely inhibited at pH 9.0. These results suggest that enhancement or inhibition of hydrolytic activities by cellobiose is dependent on the fungal species.
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|cellulbiohydrolase]]
# 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)

<table>
<thead>
<tr>
<th>Content</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>In contrast, TrCel6A exhibited slightly increased hydrolysis at pH 4.5, and hydrolysis was severely inhibited at pH 9.0.</td>
<td>pH</td>
<td>1847</td>
<td>1967</td>
<td><strong>pH_alias</strong>: In contrast, TrCel6A exhibited slightly increased hydrolysis at pH 4.5, and hydrolysis was severely inhibited at pH 9.0.</td>
</tr>
</tbody>
</table>
| 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](http://www.google.com/search?q=cellobiohydrolase)
| **BRENDA_RecommendedName**: cellulose 1,4-beta-cellobiosidase
| **SwissProt_ID**: O68438
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
}
```
Adding NLP to (Semantic) MediaWiki

René Witte, Bahar Sateli

Introduction

Natural Language Processing

Motivation

Semantic Assistants

The Wiki-NLP Integration

Requirements Analysis

System Design

Implementation

Hands-On

Configuration

Information Extraction with ANNIE

Semantic Queries

Application Examples

ReqWiki

Motivation

ReqWiki Design

ReqWiki Evaluation

IntelliGenWiki

Conclusions

References

Semantic Navigation of NLP Results

Property: Enzyme

<table>
<thead>
<tr>
<th>Page Name</th>
<th>Enzyme Entities Found</th>
</tr>
</thead>
</table>
| PubMed:19912637    | mannan endo-1,4-beta-mannosidase, BK01, endo-1,4-beta-mannosidases, mannanases, 1,4-beta-D-mannan
|                    | mannannhydrolyase, beta-mannanase, hydrolase, mannanase    |
| PubMed:21861877    | beta-glucosidase, beta-galactosidase, celllobiohydrolases, Col7A/Col6A, endoglucanases, Ce17B/Ce5A, Ce17A, Ce5A, cellulase |

Pages using the property "Enzyme"

Showing 2 pages using this property.

PubMed:19912637 + mannan endo-1,4-beta-mannosidase, BK01, endo-1,4-beta-mannosidases, ...

PubMed:21861877 + beta-glucosidase, beta-galactosidase, celllobiohydrolases, ...
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

<table>
<thead>
<tr>
<th></th>
<th>Abstract Selection</th>
<th>Full Paper Curation</th>
</tr>
</thead>
<tbody>
<tr>
<td>no support</td>
<td>IntelliGenWiki</td>
<td>no support</td>
</tr>
<tr>
<td>1 min.</td>
<td>0.3 min.</td>
<td>37.5 min.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30.6 min.</td>
</tr>
</tbody>
</table>

▶ **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

Phillip A. Laplante.  
Requirements Engineering for Software and Systems.  

Luisa Mich, Mariangela Franch, and Pierluigi Inverardi.  
Market research for requirements analysis using linguistic tools.  
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.  

Bahar Sateli, Marie-Jean Meurs, Greg Butler, Justin Powlowski, Adrian Tsang, and René Witte.  
IntelliGenWiki: An Intelligent Semantic Wiki for Life Sciences.  
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.*

René Witte and Thomas Gitzinger.
Connecting Wikis and Natural Language Processing Systems.

René Witte and Thomas Gitzinger.
Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients.