

Mymory: Enhancing a Semantic Wiki with Context Annotations

Malte Kiesel, Sven Schwarz, Ludger van Elst, and Georg Buscher

Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
Trippstadter Straße 122, 67663 Kaiserslautern, Germany

`{firstname.lastname}@dfki.de`

Abstract. For document-centric work, meta-information in form of annotations has proven useful to enhance search and other retrieval tasks. The Mymory project¹ uses a web-based workbench based on the semantic wiki *Kaukolu* that allows annotating texts both with concepts modeled in the user's personal information model and other ontologies in a flexible way. Annotations get enriched with contextual information gathered by a context elicitation component. Reading annotations are created with the help of an eyetracker.

In the demonstration, we use contextualized annotations and semantic search using annotations in order to support knowledge workers in the domain of software licenses.

1 An Introduction to the Mymory Scenario

Knowledge-intensive work often means working with multiple sources of information in parallel, reading large amounts of text, skimming even more text, creating digests, (subjectively) rating information, and many other things. Some of these texts may be thoroughly read and understood very closely, with the respective amount of cognitive processing and resulting in artifacts like annotations, summaries, citations in other texts, etc. For this demonstration, we take the domain of software licenses as an example: License texts are a typical complex texts that are difficult to read completely, that must be interpreted in multiple contexts, and that benefit a lot from both generic and personal annotations: Law experts and knowledge engineers read software license texts and create fine-grained annotations that formalize license texts. Experts and engineers can also use simple highlighting and rating annotations as personal notes useful during their formalization work. People needing to choose a software license for their project can use search and annotations created before to find licenses of interest. License text passages read by people in context of their current project get associated automatically with that project through contextualized reading annotations.

¹ <http://www.dfki.uni-kl.de/mymory>

2 Key Technologies used in Mymory

The Mymory workbench is a comprehensive environment for document-centric work, especially for reading and writing text documents, annotating them, searching in the document pool, etc. In the following we will present some of the (technical) approaches elaborated in the Mymory project and how these technical components handle the above mentioned issues of our scenario. Especially, we deal with

- a *Semantic Wiki* as a central hub for reading, writing, and annotating,
- *automated annotation* based on a user’s reading behaviour as well as on context elicitation technology,
- *manual annotation support* with concepts from the user’s Personal Information Model and other ontologies, and
- an *ontology-based search* interface which allows for dynamic assembly of faceted queries.

The **Mymory workbench** is implemented by Kaukolu², a semantic wiki research prototype. Its annotation features are demonstrated in Figure 1, depicting a software license text and its annotations. A Kaukolu annotation is an RDF resource (typically an instance of a subclass of an **Annotation** class) associated with a part of wiki text. Contrary to most other semantic wikis that typically allow annotation of links and complete pages only, any text part can get annotated here³.

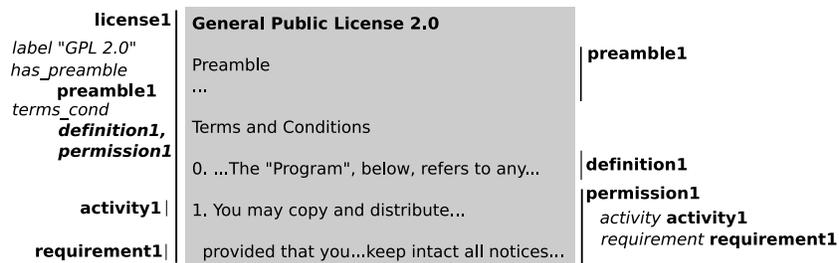


Fig. 1. Structure of software license annotations in Kaukolu.

Creating annotations manually is done using a form-based approach that fetches possible annotation classes from the Mymory RDF repository. Additionally, an **eyetracker**⁴ can create reading annotations automatically. For example,

² <http://kaukoluwiki.opendfki.de/>

³ Character offsets of the annotated text are stored and updated on markup edits by a modified text diff algorithm.

⁴ We use a third-party eyetracker integrated in a display—no glasses or other additional equipment is needed.

these annotations allow searching for individual text passages that have been read closely. For further information about the actual eyetracker’s workings, see [1]. Whenever an annotation is created, information about the current user context is attached to it. This allows filtering annotations when doing search. User context consists of author and creation time of the annotation plus the most relevant PIMO concepts currently active. **PIMO** is the *Personal Information Model (Ontology)*, a layered approach to Personal Information Models [2]. A user’s PIMO contains persons, topics, companies, and projects of importance to her as well as relationships between those entities. **Automatic user context capturing** keeps track of the *attention distribution* of PIMO concepts in form of a set of PIMO concepts along with activation values. This set is created by an additional component that hooks into key applications on the desktop. It exploits usage information and analyses documents currently used to detect PIMO concepts relevant in current context. A data structure for that (dynamic) distribution is kept together with a unique identifier (URI) for that context. If the user switches to another context, Mymory detects this and changes the current context URI as well as PIMO activation values accordingly. As the PIMO attention distribution is kept separately for each context, one context’s distribution is not “polluted” by other context’s attentions. Finally, annotations are used in **semantic search** that returns wiki text paragraphs satisfying search criteria as results. Search criteria can be various types of metadata; searching for authors of texts is supported as well as searching for text annotated using specified annotations and their context. In the end, this allows searching for text passages read in a context of a certain project, or searching for text passages annotated with a domain-specific statement such as “Here, the term *derivative work* is defined” for the software license domain.

3 System Demonstration

The live demonstration uses a document set consisting of software license texts, project websites, and wikipedia articles. It uses a software license ontology (for formalization of software license semantics), a personal ontology (containing projects and people the user knows and that is used for modeling context), and a generic annotation ontology (facilitating simple annotations such as rating texts or highlighting). In the following, we give example use cases of the features used in the demonstration.

Example 1: We want to annotate a software license text using instances according to a license ontology. Kaukolu’s form-based annotation feature is used for this task. In the end, we get fine-grained annotations in the license text that on one hand decompose the text into subparts (sections containing term definitions, etc.) and on the other hand relate statements in the text semantically with each other (*Here, copying of the software is allowed under the condition Y*).

Example 2: Imagine Arthur, a Nepomuk member, given the task to identify potential collaboration with the Mymory team. He has been informed that col-

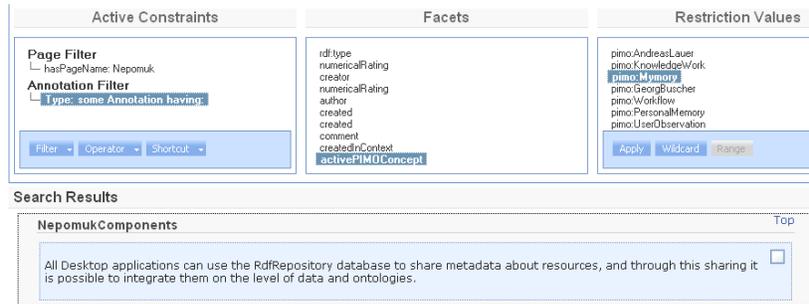


Fig. 2. Searching for text passages in Nepomuk pages with Mymory context.

laboration has taken place recently. As both projects use the Kaukolu wiki, he decides to “ask” the wiki about text passages about Nepomuk with Mymory context (and vice versa). In other words, he searches the wiki using two filters: i) passage is part of a page with a name containing Nepomuk; ii) passage has an annotation with `activePimoConcept` Mymory in its context. Figure 2 shows a text passage found in a Nepomuk page annotated with Mymory context.

Example 3: We want to take a look at text passages in license texts that we have read some time ago. Reading annotations shown in Figure 3 generated by the eyetracker are used for this during search.

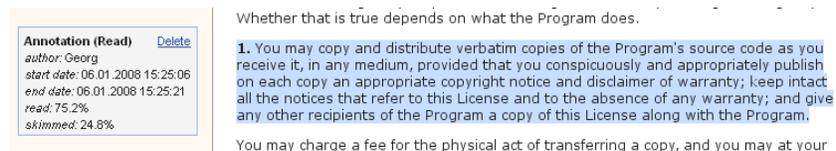


Fig. 3. A reading annotation created by the eyetracker.

Example 4: We want to get an overview of the individual software licenses used in components of the Nepomuk project. Therefore, we want to retrieve text passages read in context of the project *Nepomuk* and the topic *Software License*. This can be done by searching for text paragraphs that have been annotated with the according PIMO concepts during reading.

Example 5: After a while of using the annotation feature, we get a very large number of annotations per document. This is especially true in a multi-user environment and when using the eyetracker—then, for every user every paragraph read gets annotated with a *Read* annotation automatically. Therefore,

just showing every annotation as highlighted text passages does not scale, rendering documents unreadable. In this example we use a filtering feature that shows only annotations relevant in the current user's context. Also, we can show annotations not only as direct highlighting but also as less obtrusive icons in the document's margin which also helps keeping the system usable.

4 Conclusion

In the demonstration, we have shown that *annotations enhance the understanding and management of documents*, especially when the documents are used by more than one person and for more than one context. Annotations allow additional filters to be applied to search, which can *enhance retrieval precision*—this holds particularly for contextual annotations, respectively contextual filtering. The possibility to annotate whole wiki pages as well as small text passages with PIMO concepts allows fine-grained semantic markup of such text and enables *semantic search for information elements*. Automatic elicitation of the user's context, based on user observation providing a stream of contextual evidence, allows automatic contextualizations of annotations or even *automatic generation of attentional annotations* (e.g., annotating that a text passage has been read/skimmed by the user). We have demonstrated how storing the user's context as part of the annotations enables *contextual search/filtering* in retrieval scenarios.

Acknowledgement

The Mymory project is funded by the Bundesministerium für Bildung und Forschung (Federal Ministry of Education and Research) under grant 01 IW F01. For this work, many ideas and components from the NEPOMUK project have been used. NEPOMUK is funded by the IST Programme of the European Union under grant FP6-027705.

References

1. Georg Buscher. Attention-based information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval (doctoral consortium)*, 2007.
2. L. Sauermann, L. van Elst, and A. Dengel. Pimo - a framework for representing personal information models. In T. Pellegrini and S. Schaffert, editors, *Proceedings of I-MEDIA '07 and I-SEMANTICS '07 International Conferences on New Media Technology and Semantic Systems as part of TRIPLE-I 2007*, J.UCS, pages 270–277. Know-Center, Austria, 2007.