KAT: an Annotation Tool for STEM Documents

Sourabh Lal, Michael Kohlhase, Tom Wiesing

May 26, 2015

Abstract

Contemporary natural language processing (NLP) systems are based on corpora of annotated documents for training and evaluation. To extend NLP to documents from Science, Technology, Engineering, and Mathematics (STEM) we need annotation systems that can deal with structured elements like mathematical formulae, tables, and possibly even diagrams. Current linguistic annotation systems treat documents as word sequences and disregard the structure of complex document elements, and are therefore unsuited for STEM annotation as this very structure carries important syntactic and semantic information.

We present the KAT system, a browser-based annotation tool for linguistic/semantic annotations in structured (XHTML5, i.e. HTML + MathML + SVG in XML serialization) documents. As KAT is parametric in the annotation ontology and represents annotations as RDF, it can easily be integrated into RDF-based corpus management systems; we present an integration into the CorTeX system.

Contents

1 Introduction

[1] EdN:1

An annotation tool is a system that is used to manage annotations on a document. It provides features such as adding, modifying or removing annotations. Annotations are metadata about a document, and do not actually alter the content of the document. They can be thought of as a layer on top of a document which contains information about the text in the document. This can be done by creating annotations either inline or stand-off. Annotations can serve a variety of purposes including:

- Posting comments on the content.
- Marking out parts of the document.
- Demarcating relationships between information fragments.
- Discussing the contents of the document (using a comment thread linked to each annotation).

Annotation tools can be categorized according to the type of annotations they make. Generally state of the art anotation tools create one of the following types of annotations:

1. *Dynamic Annotations* - These create annotations that are anchored to the text of the document.
2. *Static Annotations* - These create annotations that are anchored to a particular position in the page of the document.

Annotation tools are of particular interest to the KWARC research group. Digitized, mathematical text lies in the focus of KWARC's research direction[2], and they need an annotation tool that could be used to annotate mathematical documents. The most appropriate technique for this would be the use of a dynamic annotation tool. However, dynamic annotations are fundamentally flawed when handling structured documents as they are equipped to only handle plain-text documents. This meant that KWARChad to build a new tool, which unlike other state of the art annotation tools could create annotations in structured (XHTML) documents. This new system created structured annotations; annotations that are anchored to a node in the document tree. [3] EdN:2 EdN:3

A key component of this project involved development of the frontend for KAT. We aimed to develop a user interface that optimizes system usability and improves the user experience. This involved first identifying which aspects of the user interface maximize usability by conducting design research. Using the results from this design research, we developed each of the main features that an annotation tool should support: creating annotations, modifying annotations and appropriately displaying annotations.

[1] EdNote: Have a better introductory paragraph
[2] EdNote: Why are they important in general? See abstract
[3] EdNote: Copy stuff from abstract in the paragraph

2 State Of The Art

Brat, Hypothesis Sourabh

3 Kat Overview

As KAT is based on XHTML5, we can employ the XML tool chain and rely on standard libraries for the implementation. In particular, we can use uniform resource identifiers (URI) to identify text fragments and represent annotations in RDF – subject/predicate/object triples where the components are URI references to web resources. The subjects are usually text fragments, the objects are as well (for relational annotations) or alternatively are concepts from an annotation ontology called KAnnSpec (for classificational annotations). The predicates are always properties and relations defined in the annotation ontology.

The KAT system itself is realized as a JavaScript library which instruments an XHTML5 document in a browser.

To simplify the URI-based referencing of text ranges (node-sets in the HTML document object model) KAT assumes that the document has been word- and sentence-tokenized; the tokens are wrapped in HTML span elements that carry unique id attributes corresponding to the TEI guidelines. A text range in KAT thus consists of all elements between a start and end span, referenced via ids.

The annotation workflow itself is form-based as shown in Figure 1: the annotator selects a text range, and is then given a modal form to fill classifications and relations as required by the annotation ontology. The annotations are stored as RDF triples in the browser's local storage and can be visualized by special pop-ups and arrows (see Figure 1).

Figure 1: Annotating in KAT: Selection and Form-Filling

4 Kat UI

TBD

5 Discussion

Sourabh

6 Future Work

Tom

7 Conclusion

References