Introduction to Text Technology: Using XML in Natural Language Processing

Georg Rehm

Justus-Liebig-Universität Gießen
Institut für Germanistik
Angewandte Sprachwissenschaft und Computerlinguistik
Otto-Behagel-Straße 10 D
35394 Gießen

Telefon: +49 641 99 29052
Fax: +49 641 99 29059

Georg.Rehm@uni-giessen.de
http://www.uni-giessen.de/~g91063/
http://www.uni-giessen.de/germanistik/ascl/

September 10, 2003

Abstract

This document covers the basic structure of the course "Introduction to Text Technology: Using XML in Natural Language Processing", the topics of the individual sessions, requirements for a graded Schein, the most important websites and essential reading.
Introduction to Text Technology – Course Overview

• Week 1: 09/22/2003—09/26/2003
  1. **Introduction:**
     - Text Technology: Definitions and Related Fields.
     - A Visual Introduction to XML.
     - The History of Markup Languages.
     - Example of a Text Technology application.
  2. **Markup Languages I:**
     - HTML Crash Course.
     - Example: Rumelhart's Story Grammar as an XML Application.
     - Introduction to the Syntax of XML-DTDs – Part 1: Elements.
  3. **Markup Languages II:**
     - Introduction to the Syntax of XML-DTDs – Part 2: Attributes and Entities.
     - Linguistic Background of Document Grammars.
     - How to write Document Type Definitions.
     - Parsing XML Document Instances.
     - Basic XML Software: XML Parsers and XML Editors.
  4. **Practical Exercise I:**
     - Building and parsing DTDs and XML Document Instances.
  5. **Doing it with Style I:**
     - Cascading Style Sheets.
     - Introduction to XSL.
     - XPath as a Prerequisite for XSLT.

• Week 2: 09/29/2003—10/03/2003
  6. **Doing it with Style II:**
     - XSLT.
  7. **Practical Exercise II:**
     - Handling of a standalone XSLT processor.
     - Implementation of XSLT stylesheets
  8. **Related Standards:**
     - Beyond DTDs: XML Schema.
     - XML and Related Standards.
  9. **Glueing it All Together: Text Technology Applications**
     - Text Technology Applications.
     - Programming XML interfaces: SAX and DOM.
 10. **Coming Full Circle:**
      - Topics we didn't (and couldn't) cover.
      - Disadvantages of XML.
Requirements for a Schein

In order to receive a graded Schein for this course, you can choose one of the following options. Regular attendance and especially practical work during the two exercises is mandatory.

1. If you already happen to
   
   - be in the planning or even writing phase of the final thesis for the course you are reading 
   and if your thesis could be related to Text Technology in one way or another, or if you 
   - work in an applied NLP- or Computational Linguistics research project:

   Write a paper which introduces the topic of your thesis or the project you work on and describe its potential relations to Text Technology. Where could XML-related formalisms or languages be useful? Would it be possible to implement proprietary data structures in a Text Technology way? You should make the relations as concrete as possible, i.e., you should include, e.g., example data, XSLT stylesheets, architectures, Document Type Definitions or code. The paper should not exceed 20 pages.

   Example: If your thesis deals with natural language parsing, you should discuss the representation of parse trees as XML document instances where the hierarchical structure of the parse tree is reflected by the XML structure of the instance and you could develop XML markup languages for the representation of the lexicon and the grammar.

2. If you have a background in programming:

   Develop a piece of software that utilizes Text Technology methods and write a paper as its documentation. The function of this software is open to negotiation. The paper should not exceed 20 pages.

   Example: Since the mid-90ies, corpora have become a crucial aspect of almost any NLP- or CL-related project, especially with regard to the evaluation of natural language software. Choose a language that some NL-software should process, then choose a website containing HTML documents of this language which could be processed by the abovementioned NL-software. Implement a robot and a wrapper which (a) automatically connect to this website, (b) identify webpages of interest, (c) wrap them up in simple XML code, (d) perform basic tokenization and (e) store these XML document instances in a corpus for further processing. You could even try to embed this corpus-building-process into a larger architecture, using NL-software already available at your university or department (taggers, parsers etc.). Another aspect would include the implementation of word frequency statistics, simple text summarization methods (e.g., with graphical representations based on XSLT transformations) and so on.

3. If you do not have a background in programming:

   Write a paper which discusses Text Technology methods and approaches in concrete research projects and software prototypes, which process natural language input on one of the following levels of language description or fields of application respectively: (a) morphology, (b) syntax/grammar, (c) semantics, (d) pragmatics, (e) natural language generation, (f) summarization, (g) web document analysis, or (h) information retrieval. Limit yourself to at most three different approaches and describe their individual advantages and disadvantages. Suggest potential improvements of the discussed approaches. The paper should not exceed 20 pages.
Important Standards, Books and Papers

Standards: World Wide Web Consortium

ADLER, SHARON; BERGLUND, ANDERS; CARUSO, JEFF; DEACH, STEPHEN; GRAHAM, TONY; GROSSO, PAUL; GUTENTAG, EDUARDO; MILOWSKI, ALEX; PARNELL, SCOTT; RICHMAN, JEREMY AND ZILLES, STEVE (2001): “Extensible Stylesheet Language (XSL) 1.0”. Technische Spezifikation, W3C (World Wide Web Consortium). Online verfügbar: http://www.w3.org/TR/xsl/.


AYARS, JEFF; BULLERTMAN, DICK; COHEN, AARON; DAY, KEN; HODGE, ERIK; HOSCHKA, PHILIPP; HYCHE, ERIC; JORDAN, MURIEL; KIM, MICHELLE; KUBOTA, KENICHI; LANNPHIER, ROB; LAYDA, NABIL; MICHEL, THIERRY; NEWMAN, DEBBIE; VAN OSENSBRUGGEN, JACCO; RUTLEDGE, LLOYD; SACOCOIO, BRIDIE; SCHMITZ, PATRICK and TEN KATE, WERNER (2001): “Synchronized Multimedia Integration Language (SMIL 2.0)”. Technische Spezifikation, W3C (World Wide Web Consortium). Online verfügbar: http://www.w3.org/TR/smil20/.


Standards: ISO


Standards: Requests for Comments – RFCs


Text Technological Applications


Corpora and the Web


Language Identification and the Web


Crawling the Web


The Semantic Web


Web Document Analysis


Wrapping and Information Extraction

Abiteboul, Serge; Buneman, Peter and Suciu, Dan (2000): Data on the Web – From Relations to Semistructured Data and XML. San Francisco: Morgan Kaufmann.


Searching the Web


Automatic Summarization in the Web


Natural language processing helps computer to understand human language as it is spoken. Real world use of natural languages such as English, Hindi, German, French etc doesn't have a formulated. Over the years there have been many advancements in Natural language processing. NLP Terminologies: Lets understand the Basic Terminologies used in NLP: Tokenization, Corpus or Corpora, Stemming, Bag of Words, Stop Words, TF-IDF, Disambiguation, Topic Models, Word Boundaries.

Tokenization: Tokenization is a process to split longer strings into smaller pieces. Large documents can be tokenized into paragraphs, paragraphs can be tokenized into sentences and sentences can be tokenized into phrases, words or letters.


"Instructors looking for a good introductory text to use in a course devoted to computational linguistics should consider this book as a strong candidate if they wish to emphasize a linguistics approach. The book is interestingly written with many insightful discussions, and it is the only (introductory) computational linguistics textbook that looks at the field from a linguist's point of view." NLTK (Natural Language Toolkit) is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to many corpora and lexical resources. Also, it contains a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. Best of all, NLTK is a free, open source, community-driven project. We'll use this toolkit to show some basics of the natural language processing field. For the examples below, I'll assume that we have imported the NLTK toolkit. We can do this like this: import nltk.