Teaching "Unstructured Information Management: Theory and Applications" to Computational Linguistics Students

Iryna Gurevych, Christof Müller, Torsten Zesch
Ubiquitous Knowledge Processing Group, Telecooperation Division
Darmstadt University of Technology
http://www.ukp.tu-darmstadt.de

February 9, 2007

Motivation

Students in Computational Linguistics often lack experience in building robust and scalable software components. Thus, student projects tend to be unstable and to work only under very special preconditions (e.g., a project has to be installed in a certain directory, or handles only single files instead of whole directories). Furthermore, if students have to build a system from scratch, they have to concentrate on input and output issues, as well as connecting numerous preprocessing components that were not designed to work together. This limits the scope of feasible course tasks to relatively simple ones, e.g., implementing a tokenizer.

When offering the course "Unstructured Information Management: Theory and Applications"[1] as part of the B.A./M.A. program of International Studies in Computational Linguistics at the University of Tübingen, our motivation was to familiarize the students with fundamental concepts in unstructured information management and Natural Language Processing (NLP) middleware. This should enable students of computational linguistics to work on more challenging tasks, and to gain first experiences with building complex software systems.

The course goals were supported by providing basic preprocessing components, such as a tokenizer, or a PoS tagger, on the basis of the Unstructured Information Management Architecture (UIMA) (Ferrucci and Lally, 2004). Thus, students of computational linguistics can concentrate on their core competence and work on more challenging tasks both in terms of theoretical complexity and industrial relevance. As a side effect, components developed in the course are robust and scalable, which enables their re-use by the research community.[2] UIMA let us shift the focus from software engineering to research relevant tasks, e.g., thorough evaluation of the projects.

Course description

The course was organized as a compact seminar (6 sessions, 4 hours each) representing a mix of theoretical classes and practical work. We started with a lecture explaining the theoretical concepts underlying unstructured information management, followed by practical classes and exercises introducing various components of the UIMA architecture. By the end of the course, students had to implement a project on the basis of UIMA and write a course paper. Appropriate tasks related to unstructured information management were defined in collaboration with lecturers. Bachelor projects focused on a certain UIMA based component, e.g., writing an annotator or a consumer, while Master students had to combine several annotators, or develop a collection processing engine. Among the projects targeted by Master students were the following ones:

Annotating Wikipedia articles Each structural element of a Wikipedia article (e.g., sections, paragraphs, links, lists, or bold terms) is annotated and visualized. Access to Wikipedia articles is provided using the

[1] http://www.ukp.tu-darmstadt.de/teaching/ws0607/UIMseminar

[2] UIMA based components developed in this course will be made available as part of the Darmstadt Knowledge Processing Repository http://www.ukp.tu-darmstadt.de/software.

Wikipedia API (Zesch et al., 2007).

Extracting lexical semantic information from blogs The aim of this project is to use the increasing number of publicly available weblogs (called *blogs*) to create a continuously updated lexical semantic network. UIMA is used to integrate the components for compiling the underlying corpus of blog posts, as well as for analyzing the posts to find keywords and detecting strong semantic relations between keywords.

Named entity recognition (NER) This project develops a hybrid NER system for German combining rules with several gazetteers extracted from GermaNet (Kunze, 2004) and Wikipedia[3]. UIMA is used for preprocessing (tokenization, PoS tagging) and the annotation of named entities. GermaNet and Wikipedia are accessed as UIMA resources.

Sentiment detection This projects aims at detecting sentiment expressions in English texts and linking them with the entity that is judged. UIMA is used for preprocessing, but the project additionally requires a robust NER component that is not yet available as a UIMA component. Thus, the GATE-UIMA interoperability layer is used to integrate the named entity recognition tool delivered with GATE (Cunningham et al., 2002).

Word Sense Disambiguation (WSD) The WSD approach introduced by Patwardhan and Pedersen (2006) is implemented. Necessary word glosses are generated using GermaNet as described by Gurevych (2005). GermaNet is integrated as a UIMA resource, and the necessary preprocessing steps, like tokenization and lemmatization, are provided as UIMA analysis engines.

Lessons learned

Advantages of UIMA (i) Necessary preprocessing tools can be provided as UIMA components, which enables students to work on more advanced NLP tasks. (ii) Students can concentrate on their linguistic task and do not have to think about software engineering tasks, like robustness and scalability. (iii) Course results are more likely to be re-used by the research community or the industry - another motivation boost for students.

Challenges related to UIMA A large number of UIMA concepts have to be learned, before students can start using it. Students with little programming experience found it hard to understand the connections between various UIMA components. For future courses, we suggest to better adapt the level of technical details of UIMA covered in the course to the target group. Students of computational linguistics should be provided with a preconfigured working environment, while students of computer science can be exposed to the full level of technical complexity.

Acknowledgments

This work was supported by the grant "Semantic Information Retrieval from Texts in the Example Domain Electronic Career Guidance" (SIR), GU 798/1-2. We would like to thank the Director of B.A./M.A. program of International Studies in Computational Linguistics Prof. Hinrichs for his idea to offer the seminar. We acknowledge the work and valuable contributions of the students, whose projects were described: Jonathan Khoo, Niels Ott, Sladjana Pavlovic, Maria Tchalakova, Bela Usabaev, Desislava Zhekova, and Ramon Ziai.

References

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of ACL'02*.

Ferrucci, D. and Lally, A. (2004). UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

Gurevych, I. (2005). Using the Structure of a Conceptual Network in Computing Semantic Relatedness. In *Proceedings of IJCNLP'05*, pages 767–778.

Kunze, C. (2004). *Lexikalisch-semantische Wortnetze*, chapter Computerlinguistik und Sprachtechnologie, pages 423–431. Spektrum Akademischer Verlag.

Patwardhan, S. and Pedersen, T. (2006). Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense*.

Zesch, T., Gurevych, I., and Mühlhäuser, M. (2007). Analyzing and Accessing Wikipedia as a Lexical Semantic Resource. In *Biannual Conference of the Society for Computational Linguistics and Language Technology*, pages 213–221, Tuebingen, Germany.

[3] `http://www.wikipedia.org`