Text Annotation with OpenNLP and UIMA

Graham Wilcock
University of Helsinki
graham.wilcock@helsinki.fi

Abstract

The tutorial presents a practical overview of automatic linguistic annotation of texts using freely available open source tools.

1 OpenNLP

Text annotation typically involves tasks at several linguistic levels, such as sentence boundary detection, tokenization, part-of-speech tagging, phrase chunking, syntactic parsing, named entity recognition, coreference resolution, and semantic role labelling. Most of these tasks can be done with appropriate combinations of OpenNLP tools (http://opennlp.sourceforge.net).

Practical examples will show annotations of a short English text. OpenNLP outputs annotations in a simple plain text format.

The OpenNLP tools do a good job of creating annotations automatically, but a number of issues arise. Although the OpenNLP tools themselves are open source Java and platform-independent, the annotation pipelines (where the output of one component is input to the next component) are created by Linux shell scripts and Windows .bat files that are platform-dependent and error-prone. Apache Ant can be used to gain platform-independence, but Ant requires technical skills.

2 WordFreak

OpenNLP tools can also be used in WordFreak (http://wordfreak.sourceforge.net) as plugins. WordFreak provides an attractive, easy-to-use GUI for linguistic annotations. It is open source Java and platform-independent, and is convenient for manually correcting annotations made by the OpenNLP tools. However, WordFreak creates annotations in its own specific XML stand-off annotation format.

This raises the issue of interoperability. How can annotations be interchanged between tools that use different annotation formats? This can be done by XSLT transformations, for example WordFreak XML format can be transformed by XSLT to OpenNLP plain text annotation format. However, writing such XSLT stylesheets requires specific technical skills.

3 UIMA

UIMA (Unstructured Information Management Architecture) provides solutions to many of the above issues. UIMA is open-source Java (http://incubator.apache.org/uima). It aims to support interoperability and scalability.

In UIMA, annotators run in analysis engines. New annotators are written in Java, and existing annotation tools such as the OpenNLP tools are converted to UIMA annotators by Java wrappers. Pipelines of annotators run in aggregate analysis engines. Pipelines can be configured by writing XML descriptors (similar in some ways to Ant tasks), or by means of an easy-to-use graphical configuration tool in the Eclipse GUI (Figure 1).

UIMA supports interoperability at the level of annotation formats by adopting XML Metadata Interchange (XMI), which has been proposed as an interchange standard. Instead of having its own specific XML annotation format, the UIMA annotation format is XMI.

UIMA also supports interoperability at the level of annotation tools by means of a type system that defines annotation types and their features. Types are used to check that output from one component is the right type for input to the next component.

Practical examples will show how to configure and use pipelines of OpenNLP tools in UIMA, and how to view the annotations in UIMA (Figure 2).

References

Graham Wilcock. 2009. *Introduction to Linguistic Annotation and Text Analytics*. Morgan and Claypool.

Kristiina Jokinen and Eckhard Bick (Eds.)
NODALIDA 2009 Conference Proceedings, pp. 7–8

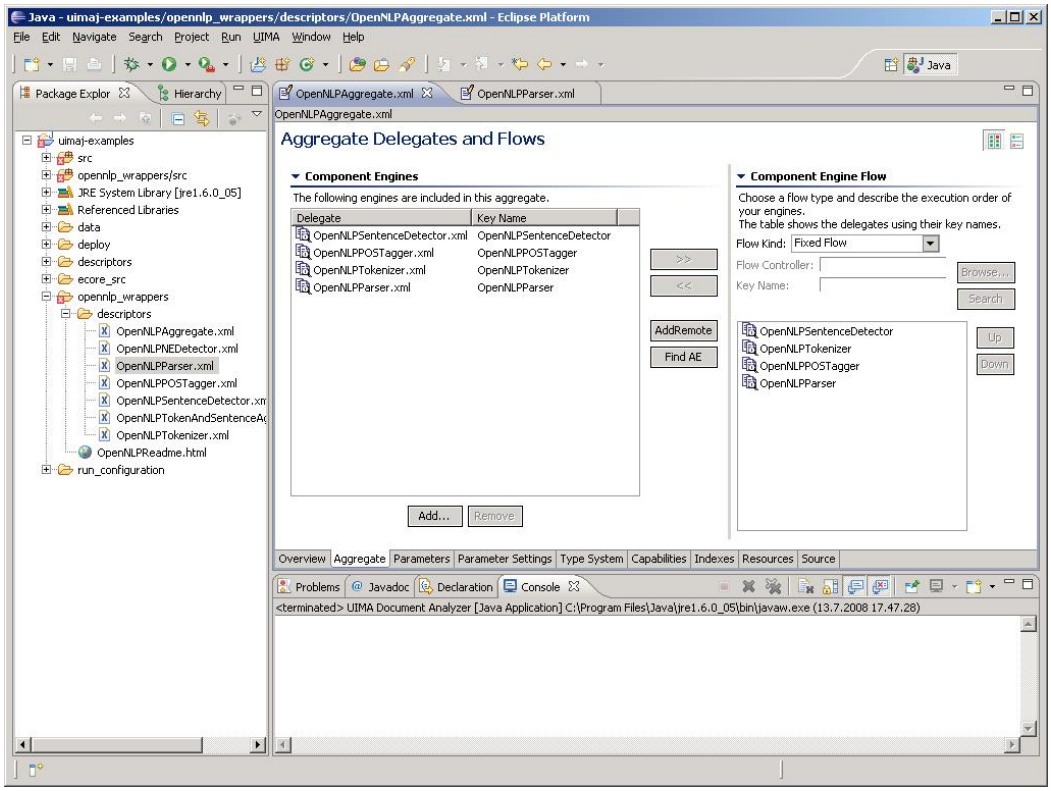

Figure 1: Configuring an OpenNLP annotation pipeline in UIMA

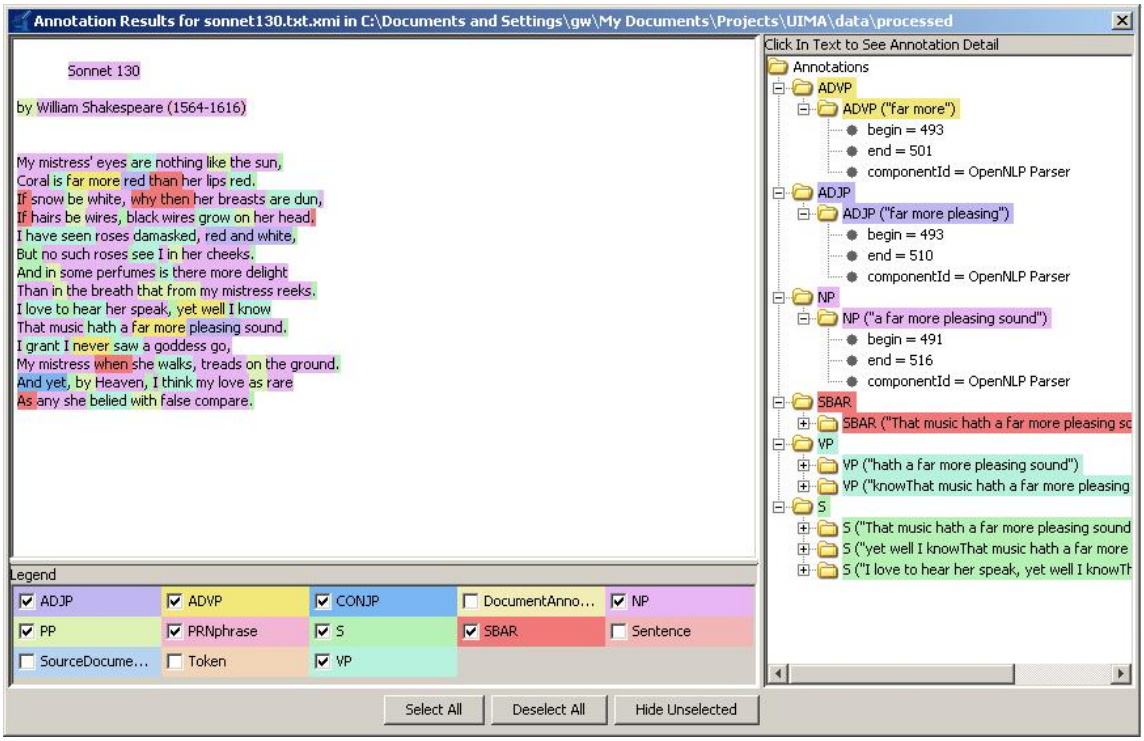

Figure 2: Viewing annotations by OpenNLP Parser in UIMA

ISSN 1736-6305 Vol. 4
http://hdl.handle.net/10062/9206