Exercise 4 – Report

Authors

Group C

- Florian Eckerstofer (0725781)
- Haichao Miao (0726810)

Documentation

CFPExtractor.java

This class contains the entire code. In general a controller is initilized and the Batch Learning PR creole plugin is added into it. The Batch Learning PR is running twice. First in Training mode, where classes are created for Machine Learning model. The Second time is in Application mode, where we apply our Machine Learning model. The resulting Annotations will be saved into an XML-file.

XmlWriter.java

The only purpose of this class is to write a list of `GateAnalysisResult` objects into a XML file.

Pre-processing with ANNIE and JAPE

Since the named annotations in the corpus do not have features we had to write a JAPE file in order to transform the documents into a format that works well with the *Batch Learning PR*.

We did this by importing the corpus into GATE and using a JAPE transducer to bring the documents in a format that was suitable for our task. The JAPE transducer we wrote can be found on `config/cfp.jape`.

Our JAPE transducer walked through all required named annotations (like `workshopname`) and created a new annotations called `IE` with the type of the previous annotation as a feature `type`.

The last thing we did in the GATE application was to export all documents as XML.

Batch Learning PR

For this part we embedded GATE into a Java application. This Creole Plugin offers different runmodes, that is sufficient for our purpose. It uses the machine-learning approach to learn patterns from our already annotated training corpus. First we have to configure through an configuration file (see `ml-config.xml`).

Training

The training corpus will be presented to the machine learning algorithm. The data must be well defined for the machine learning algorithm.

1. *instances*: These are objects, that are considered and learned by the learning algorithm. Every instance influences what the algorithms learns. Tokens are very convenient here.
2. *attributes*: The fraction of information of the instance. In our case the features of a Token. e.g. String, Length.
3. *class*: What we want to learn.

Example:

The “International School on Information Extraction” (Token-Feature: String) is a Instance of the Class conferencename.

Application:

When Batch Learning PR is applied under the Application Runmode, it creates new Annotations on our test corpus. The new Annotations are saved in the AnnotationSet “machine_learned”. They can now compared and evaluated against the original markups, to see how well the machine learner worked.

Problems during Implementation

1. Although the Gate-API is very well documented only few information could be found the the configuration on the Creole Plugin *Batch Learning PR*.
2. This was especially true for on how to write the configuration file. There are often not all possible values for a specific configuration option given in the documentation.
3. There were also only very little information given what to do when the given annotations are not a format required by the *Batch Learning PR*.
4. The GATE application was really slow when processing large amounts of documents. In the end we had to split the corpus into mutliple sets of documents (about 100 documents each) and import, transduce and export them separately. This was maybe a memory problem, however, GATE only used about 1 GB of the 16 GB available in my notebook.

Resources

We mainly used the GATE documentation to develop Information Extraction application, specifically:

- GATE Embedded
- Batch Learning PR
- Module 11: Machine Learning and resources