Assignment #6: Dependency parsing using machine learning techniques

## **Objectives**

The objectives of this assignment are to:

* Extract feature vectors and train a classifier
* Write a statistical dependency parser
* Understand how to design parameter sets

## **Organization and location**

The sixth lab session will take place on

* Group 1: Wednesday, October 15 from 10:15 to 12:00 in the Alpha room
* Group 2: Wednesday, October 15 from 10:15 to 12:00 in the Gamma room
* Group 3: Wednesday, October 15 from 13:15 to 15:00 in the Gamma room

You can work alone or collaborate with another student.

Each group will have to:

* Write and train a machine learning program to parse dependencies
* Use different parameter sets
* Evaluate the results on a corpus and comment them briefly

## **Programming**

This assignment is inspired by the shared task of the Tenth conference on computational natural language learning, [CONLL-X](http://ilk.uvt.nl/conll/), and uses similar data. The conference site contains a description of multilingual dependency parsing, reference papers, training and test sets for a variety of languages, as well as evaluation programs. See also [CONLL 2007](http://depparse.uvt.nl/SharedTaskWebsite.html), on the same topic.

In this session, you will implement and test a dependency parser for Swedish using machine learning techniques.

### **Choosing a training and a test sets**

1. The CONLL-X annotated corpora and annotation scheme are available [here](http://ilk.uvt.nl/conll/post_task_data.html). The Swedish corpus called *Talbanken* was originally collected and annotated in Lund and modified by Joakim Nivre. Read details on the corpus and references [here](http://w3.msi.vxu.se/~nivre/research/talbanken.html).
2. In this assignment, you will use the CONLL-X Swedish corpus. Download the tar archives containing the training and test sets for Swedish and uncompress them: [[data sets](http://ilk.uvt.nl/conll/free_data.html)]. Local copies: [[training set](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conllx/sv/swedish_talbanken05_train.conll)] [[test set](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conllx/sv/swedish_talbanken05_test_blind.conll)] [[test set with answers](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conllx/sv/swedish_talbanken05_test.conll)].

### **Training the classifiers**

If you have not done it in the previous assignment, learn the decision tree corresponding your data sets using Weka and produce the corresponding models from your training file.

* Start Weka. You must use version 3.7.2 or higher. You can increase the memory by typing java -Xmx6g -jar weka.jar, where the mx parameter corresponds to the maximum heap size.
* Load a data file by selecting the *Preprocess button*
* Choose and create a classifier by pressing the *Classify* button and then the *Choose* button. Use the **J48** decision trees.
* Save the corresponding model by right-clicking on the item in the *Result list*

### **Parsing the corpus and evaluating the results**

Once you have generated your models, you will embed them in Nivre's parser and compute their respective efficiencies. You will carry out the following steps steps:

* You will reuse and complement the Java code from the fifth assigment to run the parser. The parser is started in theParser.java file. The programming details are given in the next section.
* You will initialize the Weka models and couple them to your parser using WekaGlue.java. Read it to understand how this works.
* Once you have parsed the test set, you will measure the accuracy of your parser using the CoNLL evaluation script [[3](http://ilk.uvt.nl/conll/software.html#eval)]. Local copy: [[eval.pl](http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/conllx/eval.pl)]. You will run this script using the command: perl eval.pl -g gold\_standard\_file -s system\_output -q, where -q stands for quiet.
* You will run the parser with the three feature sets described in the fifth assignment to carry out a labelled and unlabelled dependency parsing. (Six experiments in total to carry out).

### **Programming and running the parser**

Read the code of the program and:

1. Complement the parser() method in the Parser.java program to carry out parsing actions.
2. Following the model of Guide4.java, write classes that extract the different feature sets.
3. Run the program with the different feature sets and report the results. Compare them with the best labelled and unlabelled scores obtained at CoNLL-X.

## **Complement (Optional)**

Read the text *Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines* by Joakim Nivre et al. (2006) [[pdf](http://w3.msi.vxu.se/~nivre/papers/malt_shared.pdf)]. Read the slides [here](http://ilk.uvt.nl/conll/slides/Nivre.pdf).