INF5830, H2013 – Project part B: Semantic Role Labeling (SRL)

Deadline: November 20th

In this assignment you will be working to solve the task of **argument classification**, an integral subtask within the larger task of SRL. We will assume that predicates and arguments have been identified and will focus on the task of labeling these arguments with semantic roles. We will be working with the original data set from the CoNLL08 shared task on syntactic and semantic parsing for English. The task will be solved as a supervised classification task using the Weka machine learning software. In doing so, you will need to process the data to extract relevant features, format these appropriately and experiment with different machine learning algorithms, in order to arrive at your final solution.

The requirement for this assignment is to submit a written report of 3-6 pages which provides details on your experiments and addresses the questions posed in sections 3-8 below. Everyone should answer sections 3-5 and 8, and you should **choose** between answering either section 6 or 7. In order to document your work, you should also submit a sample of your data-files in the .arff-format required by Weka (more on this below). The report and data should be submitted in Devilry before the deadline (November 20th, 23:59).

1. **Obtain and run the Weka software** Weka is freely available for download from `http://www.cs.waikato.ac.nz/ml/weka/`

   - Download the software following the Download-instructions available from this page. Make sure you obtain the latest stable book version of the software. On a Linux system, "installing" the software is nothing but unzipping the zip archive.
   - Browse through the documentation in the Weka Primer (`http://weka.wikispaces.com/Primer`) and try running the software as described there and using some of the data sets that come with the software. You may skip the part on filters and move on to the section on classifiers.
   - You should focus on running the software from the command line, executing a command like the following:

     ```
     java weka.classifiers.trees.J48 -t data/weather.arff
     ```

     If you get output starting with "J48 pruned tree", then congratulations, you have just successfully trained a decision tree classifier! (If not, check out the WEKA FAQ, or see the requirements, or the WEKA documentation.) Note that you may have to set your CLASSPATH variable in order to get things running properly.

2. **Obtain the CoNLL08 data sets**

   - The data sets from the CoNLL08 shared task are available through the Linguistic Data Consortium, of which the University of Oslo is a member. In order to obtain the data, please log in to a Ifi Linux server and copy the data to your home directory:

     ```
     cp /ifi/asgard/e00/liljao/CoNLL08_5830.tgz .
     ```

     You are now free to use these data as long as you do **not** distribute them to anyone outside the University of Oslo.

- Examine the official web page for the shared task (`http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=conll2008:start`) and in particular the description of the data format. Note that the CoNLL-format used in the previous assignment has been extended for this task to include information on semantic roles (from PropBank and NomBank).

3. **Data processing**

   - Start out by making sure you understand the format of the CoNLL08 data sets. In particular, figure out why the number of columns in the data varies. Also make sure you understand the treatment of hyphenated words and how this affects the representation of predicate-argument structure.
   - The classification task is as follows: given a semantic argument, provide a semantic role for the argument. In order to perform this task you will need to consider the following:
     - what constitutes the instances for classification, i.e., what is it that we want to classify?
     - which features should we use to represent the instances?
   - You will need to familiarize yourself with the .arff-format required by Weka. Please take a look at some of the data sets that are made available with the installation (in the `data`-folder). Note that the format has a few quirks that are worth taking note of (common sources of errors):
     - Weka seems to want nominal attributes (i.e. listing all the possible values for each feature) rather than "string" attributes.
     - It also does not like attribute values like "u.s.a." that contain punctuation.
     - Note also that .arff train and test files must contain the exact same lists of attributes in order to be compatible.
   - Write a program which takes a CoNLL08 data file and
     - extracts semantic arguments of verbs
     - extracts features of these arguments, e.g., their part-of-speech (PoS) and dependency relation (deprel), or the PoS or deprel of their predicate.
   - Describe your program **briefly**, using either metacode or simple prose.

4. **Baseline system**
   We will start out by training a baseline system, using a small number of simple features and a decision tree classifier. Please train on data taken from the training data set (`train.closed`), develop on the development set (`devel.closed`) and do your final testing on the held-out test set (`test.wsj.closed.GOLD`). This means that you should refrain from testing on the final test set until you have optimized your system wrt features and algorithms at the end of the assignment (section 8).

   - Train a decision tree classifier that uses the following features (taken from the Johansson & Nugues article). You may restrict yourself to verbal predicates:

     PREDLEMMASENSE The lemma and sense number of the predicate, e.g., *give.01*

     ARGPOS The (predicted) PoS-tag of the argument

     PREDPOS The (predicted) PoS-tag of the predicate

     FUNCTION The grammatical function of the argument
   - Evaluate your classifier:
     - What is the accuracy of your classifier?
     - Which five semantic roles obtain the highest F-Measures?
     - Which five semantic roles obtain the lowest F-Measures?

5. **Feature engineering** Extract additional features for your system and evaluate their performance. You should introduce at least 4 new features. In order to do so you may glance at the literature (Gildea & Jurafsky, 2002 or Johansson & Nugues, 2008) for inspiration. **NB!** You should avoid features that use word forms or lemmas of the arguments, as Weka tends to run into memory problems when employed with a large number of attribute values.

    - present your additional features and provide some examples of feature values
    - evaluate the effect of the features in terms of classifier performance

6. **Machine learning algorithm** Experiment with different machine learning algorithms and evaluate their performance. You should try out at least 2 different machine learning algorithms available in Weka, in addition to the decision tree baseline.

    - briefly present the machine learning algorithm
    - evaluate the effect in terms of classifier performance
    - can you say anything about which algorithm is best suited for this task?

7. **Nominal predicates** Train a separate classifier for nominal predicates, using the set of features obtained in section 5. You are free to add or remove features, but this is not a requirement.

    - Evaluate your classifier:
        - What is the accuracy of your classifier?
        - Which five semantic roles obtain the highest F-Measures?
        - Which five semantic roles obtain the lowest F-Measures?
        - Do you observe any differences compared to the verb classifier?

8. **Final testing on held-out data** Choose a final configuration of your system and test it on the held-out test data.

    - Evaluate your classifier:
        - What is the accuracy of your classifier?
        - Which five semantic roles obtain the highest F-Measures?
        - Which five semantic roles obtain the lowest F-Measures?