INF5830, H2015 – Semantic Role Labeling (SRL) (1 of 2)
Part 1: Feature extraction

Deadline: November 13th

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. 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 assignment is divided into two parts:

1. **Feature extraction**
2. Classification and evaluation

The requirement for the whole assignment (parts 1 *and* 2) is to submit a written report of 3-6 pages which provides details on your experiments and addresses the questions posed in the assignment. In order to document your work, you should also submit a sample of your input data (more on this below). The report and data should be submitted in Devilry before the deadline (November 13th, 23:59).

1 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 this article describing the shared task (`http://dl.acm.org/citation.cfm?id=1596324.1596352`) 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).

2 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 are the instances for classification?
  - which features should we use to represent the instances?
- Write code which takes a CoNLL08 data file and
  - stores the data in a multidimensional datastructure that allows you to access the different fields (e.g. PoS, dependency relation) by token index
  - locates the semantic predicate(s) in the sentence
  - extracts semantic arguments of verbs, all the while keeping track of the predicate for each argument
- Describe your program **briefly**, using either metacode or simple prose.

3 Feature extraction

- You should now have what you need in order to extract features for your classifier. It is a good idea to store the full feature vector for each instance as a dictionary in a list of dictionaries:

```
[ {feature1:value1, feature2:value2, ..., feature n:valuen},
  {feature1:value1, feature2:value2, ..., feature n:valuen} ]
```

- The classes for each instance should be stored in a separate list, where the order of the classes corresponds to the order of the instances, such that e.g. `classes[i]` contains the class label for the ith instance.
- Your feature extraction code should extract the following basic 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

4 Feature engineering

- Extract additional features for your system. 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. Write a short description of your additional features and how these were extracted. Provide examples of some feature values.
- As part of your submission for this assignment you should process the training data and print the feature vectors of the first 10 instances along with their classes to a file entitled `features.txt`.