Language Technology - EDAN20
Assignment 6 - Fall 2020

Dependency Parsing Using Machine Learning Techniques

Hicham Mohamad, hi8826mo-s
hsmo@kth.se

August 30, 2020

1 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
- Write a short report on your results

2 Organization and location

The sixth lab session will take place on October 15th and 16th. There can be last minute changes. Please always check the official times here:

`https://cloud.timeedit.net/lu/web/lth1/ri1Q5006.html`

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

3 Programming

This assignment is inspired by the shared task of the Tenth conference on computational natural language learning, **CONLL-X**, and uses a subset of their 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**, on the same topic.

Please note that the original CoNLL-X site is down. To access the pages, use the Archive.org site:

`https://web.archive.org/web/20161105025307/http://ilk.uvt.nl/conll/` and to download the data sets, use the local copies.

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**. The Swedish corpus called *Talbanken* was originally collected and annotated in Lund and modified by Joakim Nivre. You can read details on the corpus and references **here**.

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**]. Local copies: [**training set**] [**test set**] [**test set with answers**].

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.

Your parser will proceed, sentence by sentence, and word by word. For a certain state, it will predict the next action using your classifier. You will then execute the corresponding action: la, ra, re, or sh. If an action is not possible, you will carry out a shift.

You are free to implement it the way you want. Here are some suggestions:

- The loop will basically have this structure:

```
while queue:
    features.extract()
    trans_nr = classifier.predict()
    stack, queue, graph, trans = parse_ml(stack, queue, graph, trans)
```

- The parsing function, `parse_ml()`, takes the the stack, queue, graph, and the transition predicted by the classifier, and carries out the transition. You can use this model and complete it:

```
def parse_ml(stack, queue, graph, trans):
    if stack and trans[:2] == 'ra':
        stack, queue, graph = transition.right_arc(stack, queue, graph,
            trans[3:])
        return stack, queue, graph, 'ra'
    ...
```

  where `trans` is either `ra.deprel`, `la.deprel`, `re`, or `sh`.

- You will then use the partial graph to write the values of the heads and functions to the words. item Finally, you will save the sentences in an output file.

- Once you have parsed the test set, you will measure the accuracy of your parser using the **CoNLL evaluation** script [**3**]. Local copy: [`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 dependency parsing.

- You need to reach a labelled attachment score of 75 to pass this lab.

4 Reading

Read the article: *Globally Normalized Transition-Based Neural Networks*, by Andor and al. (2016) [**pdf**] and write in a few sentences how it relates to your work in this assignment.

5 Appendix

References

[1] Pierre Nugues, *Language processing with Perl and Prolog*, 2nd edition, 2014, Springer.

[2] P. Nugues, Language Technology - EDAN20, Lectures notes: `https://cs.lth.se/edan20/`

[3] D. Jurafsky, J. Martin, *Speech and Language Processing*, 3rd edition. Online: `https://web.stanford.edu/~jurafsky/slp3/`

[4] VanderPlas, A Whirlwind Tour of Python. O'Reilly 2016. Online reading: `https://jakevdp.github.io/WhirlwindTourOfPython/`

[5] Pierre Nugues, Assignment 6: `https://github.com/pnugues/edan20/blob/master/notebooks/6-dependency_parsing.ipynb`

[6] Universal Dependencies: `https://universaldependencies.org/`

[7] CoNLL 2018 Shared Task, Multilingual Parsing from Raw Text to Universal Dependencies: `http://universaldependencies.org/conll18/`

[8] D Andor et al., Globally Normalized Transition-Based Neural Networks, 2016. Google Inc New York, NY

[9] Joakim Nivre, An Efficient Algorithm for Projective Dependency Parsing, (2003) `https://cl.lingfil.uu.se/~nivre/docs/iwpt03.pdf`

[10] Geron, Jupyter notebook on linear algebra from Machine Learning and Deep Learning in Python using Scikit-Learn, Keras and TensorFlow: `https://github.com/ageron/handson-ml2`