Machine Learning for NLP Assignment 1

Deadline: November 6th 2020

This first assignment consists of a small theoretical component and the first preparations for building and evaluating systems for Named Entity Recognition. The recommended time division is that you complete the first two sections in the first week and the third section in the second week.

All code needed for this assignment can be found under code/assignment1 in the github repository: https://github.com/cltl/ma-ml4nlp-labs

Please indicate how much time you spent on each component in your submission. This is meant to ensure the workload of the course is appropriate (you get as much out of it as you can without becoming overworked). You can also provide a complete overview at the end of your submission.

1 Theoretical Component

1.1 What is machine learning?

Provide an explanation of your own view in a short paragraph. If you take a definition from someone else, explain why you think it is a good definition or how you would alter it to make it a good definition. You do not need to do research to answer this question: your current understanding of machine learning is sufficient.

Submit this together with an indication of the time you spent on the course during the first week as a pdf in 'presubmission theoretical Assignment 1, week 1'.

2 Preparation Experimental Setup

Each NLP experiment or NLP system development starts by getting a good understanding of the task and ensuring you have the right materials for developing your system (or for doing your research). This means you need to get an understanding of the goal of the task and the data and you need to make sure you have the means to evaluate your outcome. This exercise will force you to explore named entity recognition as a task, to carry out some (basic) preprocessing steps and setup a standard basic evaluation.

2.1 Understanding the task and the data

In Natural Language Processing, it is often worthwhile to explore the data, i.e. look at some examples and snippets. We generally use data structures that are intuitive to understand: you can either figure out how they work by looking at them or by finding information online. We will look at the training and evaluation data provided through the conll shared task of 2003 (Sang and De Meulder, 2003) and the output of Stanford CoreNLP and Spacy on this data.

In this case, there are two things you need to figure out:

1. the exact task: which types of named entities are annotated in the gold data and what distinctions does each of the systems make?
2. the representation format: how is the output of the tools presented in conll and how does the BIO schema work (as used in the gold standard)?

You can probably figure both of these questions out by carefully studying the data (gold_stripped.conll, spacy_out_matched_tokens.conll and stanford_out_matched_tokens.conll): by looking at the files and using basic scripts to extract all classes that are annotated or assigned. You can also search for information on the systems and representation format online (information is easy to find with a search engine using relevant keywords).

There are differences between the conll shared task data and the output of Spacy and Stanford. First, Spacy and Stanford each have their own tokenization and it differs from the conll one. Fixing differences in tokenization is not complicated, but it can be a time consuming and burdensome task. Therefore, I have fixed this first issue for you. Second, Spacy and Stanford each have their own set of classes (different kinds of named entities) that do not correspond to those in the conll gold data. You will need to study the data (or descriptions of the systems) to figure out what the differences are.

You will need to provide a short description of the task of named entity recognition and the classes that are distinguished in your final report.

2.2 Preprocessing

When we talk about 'preprocessing' we mean any kind of steps carried out to prepare the data. This can be cleaning data (e.g. removing special characters), but also token alignment as mentioned above. In this case, we will carry out a step in which we convert named entity labels, so that the classes of the systems and the gold data correspond.

1. Walk through the code in preprocessing_conll.ipynb and figure out what this code does (post questions on the Discussion Board if you get stuck!)
2. Decide on what conversions you would like to carry out to make sure the output of individual systems and the gold data aligns. You will need to describe your decisions in your report.
3. Complete the conversions.tsv file according to your decisions.

4. Convert the files

NB: you may also write your own code to do this, if you prefer or change the code in any way you see fit. It does not make a difference in your grading whether you use the code that is provided, an adapted version of it or our your own code.

2.3 Basic Evaluation

Now that you have the output of two systems and gold that are mapped to corresponding classes, we can complete our experimental setup with basic evaluations (for now). You will program functions that can provide precision, recall and f-score as well as a confusion matrix. For this specific assignment, you should provide these metrics and the confusion matrix **without making use of external modules**: i.e. your code should include the actual calculations.

1. Go through the notebook basic_evaluation.ipynb. The notebook provides examples of how to read data in using Pandas, how to output tables and how to test code using assert statements. It also contains tests using the mini-datafiles provided on github.
2. Complete this code adding functions that can compare system and gold results and provide precision, recall, f-score and a confusion matrix.
3. Run the tests to see if your code works properly.
4. Run the evaluation on the Stanford and Spacy output.

You can change the structure of the code or the provided functions in any way you see fit. Just remember that you must program the code that carries out the calculations yourself and you should not rely on external modules for this (e.g. by scikit-learn). The results will be included in your final report.

Make a copy of your local 'code' directory with solutions with your student number as its name. Create a .zip or .tar.gz file from this folder and submit this as part of 'presubmission code Assignment 1, week 1'. Do not include large data files in your submission.

3 A Basic System (first results)

In this final component, you will train a basic named entity recognition system and start writing your report.

3.1 A basic system

We will start with a simple logistic regression system that makes use of one feature only.

1. Examine the code provided in basic_system.ipynb and check out the documentation mentioned in the notebook.

2. Provide documentation to the code. Make sure to ask questions on the discussion forum if there are things you do not understand: you will need to extend this code next week.

3. Provide the correct arguments and train your system. You can then evaluate the output of this system using your code from Component 2.3.

3.2 Writing your report

Describe the work you did these two weeks as part of your report:

1. Provide a short description of named entity recognition as a task and the data that you are using.

2. Describe the preprocessing steps you carried out, including an overview of the conversions you applied to the class labels with a motivation.

3. Present the results of Spacy, Stanford and your system in a table accompanied by a brief description and comment.

4 Submission Assignment 1

By the end of Week 2, submit the following (this submission is **obligatory**):

1. Theoretical component. A pdf file including:

    - Your definition of machine learning
    - The first components of your report (as outlined above)
    - An overview of the time spent on individual components of the course during these first two weeks

2. Practical component. A .zip or .tar.gz file of a folder with your student id as its name. This folder should contain any code you have worked with and written for this first assignment, with a readme explaining how to run things, if necessary (this is currently not necessarily if you simply completed the notebooks provided).

References

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.