Machine Learning for NLP Assignment 2

Deadline: November 20th 2020

The theoretical part of this assignment covers more theoretical groundwork on task design and features. In the practical part, you will extend your system so that it covers more machine learning methods and makes use of more features.

You will need to make use of the documentation on the basic system you created as part of Module 1. You can find a script with the code, a new function for working with embedings and some tips under code/assignment2 in the github repository (use git pull for obtaining the last version): 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 Goal and Task: System Design NER and one other NLP task

During the workgroup of November 11th, you will start joint work on finding information about a specific NLP task: what is the task and how are researchers tackling it? If you missed that workgroup, please contact me and I will assign one of the tasks to you (you may not select your own task).

1. Provide an overview of related work and the proposed task design (which features and how will you represent them) on the task you worked on in class (or the task assigned to you) in a 1-2 page report. You may work together to find information and discuss options with your workgroup, but you need to hand in an individual report. This 1-2 page report will be included in the theoretical component with your other theoretical questions
2. Include the overview and proposed task design for Named Entity Recognition in your system description report.

Please note that it is sufficient to explore related work for the purpose of this exercise (i.e. find a handful of relevant papers is sufficient in this case). For a research paper or thesis, a more elaborate study would of course be required, where you make sure to cover the latest insights and state-of-the-art.

2 Extending your system

As part of the practical component, you will extend your system so that it covers more machine learning algorithms and more features. You will first extend the system to cover more 'traditional' features, represented as one-hot encoding (as part of your presubmission) and then investigate word embeddings. An example of how to use word embeddings as a feature is provided in your script. The setup is specifically designed so that you can combine word embeddings with other features later on.

2.1 Including alternative machine learning methods

The code provided for Assignment 2 provides a setup that supports running experiments for multiple systems. It holds here as well that you are allowed to use everything that is provided, but you can also restructure the code or reimplement parts as you see fit (this does not affect your grade in either way).

Please extend your code so that you can also train models using:

- Naive Bayes: scikit learn provides multiple variations. Try to select the best one based on their descriptions.
- SVM.

Train and evaluate your systems. You do not need to tune your systems for specific settings for now, but if you do, please use 10-fold cross validation on the training set to do so (i.e. do not tune on the gold data).

2.2 Introducing more features

The second extension during this module is adding more features to your system. During the first week, you will add 'traditional features' represented as one-hot encodings (just like the tokens in the basic system). In the second week, you will investigate embeddings (which we will cover in class). You will need to carry out the following steps:

- Identify which features you would like to use and what values these features can have (taking into account that they will be presented as one hot representation).
- Add those features to your system. Tip: if you plan to include some more advanced features, it can be helpful to first create a "feature extraction" script that identifies feature values and outputs them in the conll format. Adding these features to your system then becomes quite straight-forward.
- Integrate the option of using word embeddings to your system. The script provided as part of assignment2/ includes a function that obtains the feature representations already. The (commented out) example assumes you are using the Google word embeddings, which can be obtained here: https://code.google.com/archive/p/word2vec/ (search for "GoogleNews-vectors-negative300.bin.gz").

2.3 Updating Your System Description

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

1. Update the short task description (from Assignment 1) if necessary.
2. Include the (short) related work section (created as part of your theoretical component during this module, as mentioned above).
3. Describe the dataset including the preprocessing steps you carried out (from Assignment 1, updated if necessary).
4. Provide an overview of the features (with motivation) and systems you are using.
5. Provide an overview of your results (those from Assignment 1 and those from this module).

3 Submission Assignment 2

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

1. Theoretical component. A pdf file including:
    - The two page description of the system design and features for the other NLP task assigned to you
    - Your updated system 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 the first assignment and second assignment, with a readme explaining how to run things, if necessary. **The language model you used for obtaining the embeddings should NOT be included in your submission.**