Machine Learning for NLP Assignment 2

Deadline: November 25th 2022

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 function for working with embedings and some tips under code/assignment2 in the github repository.

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 other NLP tasks: one basic and one advanced

During the workgroup of November 16th, we will discuss other NLP tasks. We start with relative intuitive tasks. During the workgroup of November 23rd, you will start joint work on finding information about a more complex NLP task: what is the task and how are researchers tackling it? Please note that we will assign the task to you (you are not allowed to pick your own task). For this component, you will carry out a basic literature study on both tasks.

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.

1. Provide a task description and a proposed task design for one of the basic tasks we discussed in class. The overview should contain the most commonly used features (in systems that use explicit features) and refer to the literature where you found them.

2. 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 Extending your system

For our Named Entity experiments, you will extend your system so that it covers more machine learning algorithms and more features. Carry out a basic literature review for named entity recognition. This review need not be extensive, but is meant to make sure you include features that are easy to obtain and known to work. 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 Introducing more features

You will expand the features you use further, starting with one-hot features and then moving on to word embeddings. You 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").

2.2 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, unless your code becomes significantly worse).

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 with standard settings.

2.3 Hyper-parameter tuning

Read the scikit-learn.org documentation on hyper-parameter tuning: `https://scikit-learn.org/stable/modules/grid_search.html`. Carry out hyper-parameter tuning for your SVM. You need not explore all possible hyper-parameters and can choose which form of search you apply (exhaustive, randomized, halving). You can also choose whether you want to do cross-fold or tune on your development data. You'll need to provide a motivation of your choices.

2.4 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 a short description of the related work you found.
3. Update the description of the dataset including the preprocessing steps you carried out (from Assignment 1, update 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 Recommended Partial Submission Assignment 2

A the end of Week 1 of Module 2 (Week 3 of the course), we recommend you submit:

1. Theoretical component: the description of a (basic) NLP task that is not NERC
2. Report:
   - Overview of literature (first draft)
   - Overview of features NERC system and a motivation
   - Start with overview of results (need not be complete)
3. Code/Practical:
   - Complete all one-hot features
   - Run Naive Bayes and SVM (one set of settings)
   - Read up on hyper-parameter tuning (and plan for carrying this out)

4 Full Submission Assignment 2

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

1. Theoretical component. A pdf file including:

    - The description of one of the basic tasks (0.5 - 1 page)
    - The 1-2 page description of the system design and features for the other NLP task assigned to you (first draft)

2. An update of your NERC research report:
    - Your updated system report (as outlined above)
    - An overview of the time spent on individual components of the course during these first two weeks

3. 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.**