20-00-0546-iv Foundations of Language Technology

Homework 8
Categorizing and tagging words (part 2)

17. December 2020

In case your submission consists of several files, compress these to a `zip`-file. Indicate clearly which submission corresponds to which question. Include comments in your program code to make it easier readable. It is very important that you submit your solution as a Jupyter Notebook file (`.ipnyb`).The deadline for the homework is **Friday, 01.01.21 23:59 CET**.

8.1 Homework

In natural language processing, machine learning algorithms are widely used to automatically classify texts according to a given task. Usually, a training and a development corpus are given in advance to prepare the classification system. The training corpus is used to train a system, the development corpus is used to fine-tune the parameters. The testing corpus is usually kept secret and is used only for the **final** evaluation of the systems. It must not be used for testing during the development phase and for fine-tuning.

In the next task, you will create a part-of-speech tagger. The training, the development and the test set are given to you in advance. Your tagger is supposed to be only trained on the training corpus and maybe on the development corpus. You must not use the test corpus for training or fine-tuning your (hyper)parameters.

Listing 1: Datasets

```
import nltk
from nltk.corpus import nps_chat, brown

def get_train_dev_test(tagged_parts):
  nr_parts = len(tagged_parts)
  #first 80% of the corpus
  train_parts = tagged_parts[:(nr_parts*8)//10]
  # 80-90% of the corpus
  development_parts = tagged_parts[(nr_parts*8)//10:(nr_parts*9)//10]
  # last 10% of the corpus
  test_parts = tagged_parts[((nr_parts*9)//10):]
  print(nr_parts, ":", len(train_parts), len(development_parts), len(test_parts))
  return train_parts, development_parts, test_parts

train_posts, development_posts, test_posts = get_train_dev_test(
    nps_chat.tagged_posts(tagset='universal'))

train_sents, development_sents, test_sents = get_train_dev_test(
    brown.tagged_sents(tagset='universal'))
```

Homework 8.1 *(2 points, no programming needed)*

Please explain why POS taggers are important for natural language processing and what kinds of downstream tasks are enabled with POS tagging (give one example). What different strategies for implementing a POS tagger do you already know and what are the pros and cons of these approaches (name at least two). Do you have suggestions how they can be improved?

Answer each question in two or three sentences.

Homework 8.2 *(8 points) Develop a part-of-speech tagger for texts in the chat domain and the brown corpus. You may explore several directions such as handling rare tokens, handling special chat related phenomena like smilies, or using better and more training data. You can try different things to improve the performance of the tagger on the training data, but be careful not to overfit on the training data. Document your ideas and process well.*

Hints:

- *You do not need to implement a tagger from scratch.*
- *You may train your tagger on more than one corpus.*

- *The corpora provided by NLTK offer a README in the corresponding folder, i.e. within the sub folders of* `nltk\_data/corpora` *in your home folder (*`Anwendungsdaten` *for Windows) you will find additional information.*
- *Use the universal tag set.*
- *Typical chat tokens like 'lol' or ':-)' should be tagged as 'X', i.e. other.*
- *Read the paragraph "Tagging Unknown Words"*[1].
- *use* $'importmatplotlib.pyplot\ as\ plt'$ *for the plotting*

Collect your top 3 results on the development set for the brown and the chat corpus with the diffrerent hyperparameters and plot them. Evaluate your best tagger for brown and the chat corpus on the testset. Upload the code and report the accuracy as part of your submission. Use the text field in the submission module to submit the final accuracy (just put a single number there for each corpus).

[1] See NLTK-book page chapter 5.5, page 206