SNLP 2016
Exercise 7

Submission date: 17.06.2016, 23:59

You can use NLTK for text normalization (incl. punctuation removal, stemming/lemmatizatio, stopword removal and tokenization) you need in this exercise sheet. As for lemmatization, we suggest you to use *Pattern*[1] and specially not to use *WordNetLemmatizer* from *nltk* as it performs really bad!

Feature Selection

1) (2 point) **Chi-Square**

Imagine we have a predefined "Physics" class and we want to see whether the terms "apple" and "measure" are good features for predicting this class or not. The following table is given for this purpose:

	apple	measure	rest
Class = Physic	5	100	400
Class = ~Physics	100	50	9450

Do this task by evaluating $\chi^2(apple, Physics)$ and $\chi^2(measure, Physics)$.

2) (4 point) **Mutual Information**

- Use the documents provided in "Materials/train" to construct the vocabulary. You need this vocabulary for the next exercise as well. Remember to do the text preprocessing:
  - stopword removal with the stopwords.txt given
  - lowercasing
  - lemmatization + stemming
  - tokenization
- (1 point) Find the mutual information between each term and each class (topic). Compute $pmi(t)$ in the case we want each term to discriminate well for a single category.
- (1 point) Use the $pmi(t)$ s to do the feature selection such that it results in 10 features and report them. How much has the your problem's dimension decreased?
- (1 point) Do the feature selection this time by MI [2] and select the 10 terms with greatest MI. How do these features differ from the previous section? Report these features and their difference with previous section.
- (1 point) Use the features obtained from each case separately to classify each test file by Naiive Bayes Classifier like previous exercise sheet:
  - Compute the likelihoods for each word (after feature selection) in each class (topic)
  - Assume uniform prior probability for classes

[1] `http://www.clips.ua.ac.be/pattern`

[2] `https://en.wikipedia.org/wiki/Mutual_information` in which each term is a random variable

  - Classify by posterior probability

3) (4 point) **tf-idf**

- (2 points) Familiarize yourself with *tf-idf* via `https://en.wikipedia.org/wiki/Tf` and compute it for each word in the vocabulary.
  Do the feature selection by keeping the 400 greatest *tf-idf* words. Breifly comment on the advantage of using *tf-idf* rather than *df*.

- (0.5 point) Again use the features obtained from above to classify the "test_b.txt" file by Naiive Bayes Classifier and report the result.

- (1.5 point) Use KNN as your classifer with K=1. Put the frequency of the words as the values of each feature in the feature vector of each document. Use *Euclidean Distance* as your distance measurement between the two vectors. How do you explain the result.

1 Submission Instructions: Read carefully

- You can form groups of maximum 3 people.

- Submit only 1 archive file in the ZIP format with name containing the MN of all the team members, e.g.:

Exercise_01_MatriculationNumber1_MatriculationNumber2_MatriculationNumber3.zip

- Provide in the archive:

  - your code, accompanied with sufficient comments,
  - a PDF report with answers, solutions, plots and brief instructions on executing your code,
  - a README file with the group member names, matriculation numbers and emails,
  - Data necessary to reproduce your results [3]

- The subject of your submission mail must contain the string [SNLP] (including the braces) and explicitly denoting that it is an exercise submission, e.g:

[SNLP] Exercise Submission 01

- Depending on your tutorial group, send your assignment to the corresponding tutor:

  - Sedigheh Eslami: *eslami@mpi-inf.mpg.de*
  - Naszdi Kata: *b.naszadi@gmail.com*
  - Stephanie Lund: *stflund@gmail.com*

2 General Information

- In your mails to us regarding the tutorial please add the tag [SNLP] in the subject accompanied by an appropriate subject briefly describing the contents.

- Feel free to use any programming language of your liking. However we strongly advise in favor of Python, due to the abundance of available tools (also note that Python3 comes with an excellent native support of UTF8 strings).

[3] If you feel that these files are beyond reasonable size for an email submission and also reasonably convenient, please provide a means for us to access them online

- Avoid using libraries that solve what we ask you to do (unless otherwise noted).
- Avoid building complex systems. The exercises are simple enough.
- Do not include any executable files in your submission, as this may cause the e-mail server to reject it.
- **In case of copying, all the participants (including the original solution) will get 0 points for the whole assignment. Note: it is rather easy to identify a copied solution. Plagiarism is also not tolerated.**
- **Missing the deadline even for a few minutes, will result in 50% point reduction. Submission past the next tutorial, is not corrected, as the solutions will already be discussed.**
- **Please submit in your solutions necessary to support your claims. Failure to do so, might results in reduction of points in the relevant questions.**
- Each assignment has 10 points and perhaps some bonus points (usually 2 or 3). In order to qualify for the exams, you need to have 2/3 of the total points. For example, in case there are 12 assignments, you need to collect at least 80 out of the 120 points to be eligible for the exams. A person that gets 10 plus 2 bonus points in every exercise, needs to deliver only 7 assignments in order to be eligible for the exams, since 7*12=84.
- Attending the tutorial gives 2 points increase for the corresponding assignment.
- Exercise points (including any bonuses) guarantee only the admittance to the exam, however have no further effect on the final exam grade.