UNIVERSITÄT DES SAARLANDES

Statistical Natural Language Processing

Summer Semester 2020
Prof. Dr. Dietrich Klakow

Exercise Sheet VIII

Submission Deadline: July 3rd, 23:59

1 Feature Selection

1.1 Pointwise Mutual Iinformation (3 points)

In this exercise you will perform feature selection using Pointwise Mutual Information (PMI).

Extract the archive `movie_review_data.zip` provided together with the excercise sheet. It contains 2 directories with positive and negative movie reviews. You should extract the text and remove all non-alphanumeric characters including punctuation and stopwords defined in nltk (use `stopwords from nltk.corpus`).

Extract all unigrams and bigrams for positive and negative reviews. Note that you should also filter out all bigrams when at least one of the words appears in the stopword list.

Compute PMI for unigrams which occur at least 100 times in the corpus and report the top 20 values with the corresponding words for the positive class and top 20 values for the negative class. Do the same for bigrams but use 50 occurrences as the frequency threshold and compare the results. Can you predict whether the unigrams and bigrams with the highest PMI values come from the positive or negative reviews judging from the words alone? Why do we need to filter out low frequency words? What would happen if we considered all words in the corpus for the PMI calculation?

Also compare PMI and frequency values of word *good* versus *bad* for both positive and negative classes. Which difference do you observe?

1.2 Chi-square Test (1.5 points)

Assume that you have the following counts of words {*good*, *interesting*} and {*nice*, *entertaining*} from the movie reviews as shown in Table 1.

	positive	negative
good	7397	7159
interesting	1449	1603

	positive	negative
nice	1105	854
entertaining	817	558

Table 1: Observed counts

Use the Chi-square test to find out whether the distributions of two different sets of words: {*good* and *interesting*} and {*nice* and *entertaining*} depend on

the type of movie reviews (positive vs negative). In other words, you will use test statistics to check whether the distribution depends on the category.

Your solution should include the computation of the Chi-square value and the corresponding p-value for alpha=0.05.

2 Classification

2.1 Support Vector Machine (4 points)

In this exercise you will implement a sentiment classifier using Support Vector Machine (SVM). Use the dataset from `movie_review_data.zip` which you explored in the previous exercise. Using the same pre-processing steps as specified in part 1, extract unigrams and bigrams and apply CountVectorizer from the `sklearn` library[1]. This will allow you to generate features for the classifier.

Divide the data into training (80%) and test (20%) sets. You can use `train_test_split from sklearn.model_selection`.

Train SVM classifier with linear kernel using only unigrams as features and report the accuracy scores on the test set. Create another classifier using bigrams as features and compare its performance to the unigram-based classifier. For each case provide a confusion matrix which shows number of correctly and incorrectly classified instances of each class.

You should also experiment with different values of the regularization parameter C (e.g. using a range of values between 0.5 and 1 for the parameter tuning). Briefly explain the impact of the parameter choice on the classification.

Apply tf-idf[2] to the counts and compare SVM performance with simple counts to the results obtained with tf-idf counts.

Note: In case the computation is very slow on your computer, reduce the corpus size by removing the equal proportion of positive and negative reviews and mention any modifications in the report.

2.2 Different Classifiers (1.5 points)

Use one classifier of your choice: Decision Tree, Maximum Entropy (Logistic Regression) or Multinomial Bayes to perform the classification[3]. Briefly describe how the classifier works and which parameters it needs (if any) and compare the results to the SVM performance. In which situation it is optimal to use the selected classifier and in which situation it is not a good choice?

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html`

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer`

[3] See more details about the classifiers: `https://scikit-learn.org/stable/supervised_learning.html`

Submission Instructions

The following instructions are mandatory. Please read them carefully. If you do not follow these instructions, the tutors can decide not to correct your exercise solutions.

- You have to submit the solutions of this exercise sheet as a team of 2 students.
- If you submit source code along with your assignment, please use Python unless otherwise agreed upon with your tutor.
- NLTK modules are not allowed, and not necessary, for the assignments unless otherwise specified.
- Make a single `ZIP` archive file of your solution with the following structure
  - A `source_code` directory that contains your well-documented source code and a `README` file with instructions to run the code and reproduce the results.
  - A `PDF` report with your solutions, figures, and discussions on the questions that you would like to include. You may also upload scans or photos of high quality.
  - A `README` file with group member names, matriculation numbers and emails.
- Rename your `ZIP` submission file in the format

  `exercise08_id#1_id#2.zip`

  where `id#n` is the matriculation number of every member in the team.
- Your exercise solution must be uploaded by only one of your team members under *Assignments* in the *General* channel on Microsoft Teams.
- If you have any problems with the submission, contact your tutor before the deadline.