Assignment 4

Information Retrieval and Text Mining 23/24

Publication: 2023-12-14
Submission Deadline: 2024-01-11
Discussion Session: 2024-01-16

Yarik Menchaca Resendiz, Roman Klinger

irtm-teachers@ims.uni-stuttgart.de

- **Groups:** Working in groups of up to three people is encouraged, up to four people is allowed. More people are not allowed. Copying results from one group to another (or from elsewhere) is not allowed. Changing groups during the term is allowed.
- **Grading:** Passing the assignments is a requirement for participation in the exam in all modules IRTM can be part of. Altogether 80 points need to be reached. There are five assignments with 20 pen & paper points and 10 programming points each. That means, altogether, 150 points can be reached.
- **Submission:** First make a group in Ilias, then submit the PDF. Write all group members on the first page of the PDF. Only submit *one* PDF file. If you are technically not able to make a group (it seems that happens on Ilias from time to time), do not submit a PDF multiple times by multiple people – only submit it once. Submission for the programming tasks should also be in the same PDF.
- **Make it understandable:** Do the best you can such that we can understand what you mean. Explain your solutions, comment your code. Print the code in a readable format, write your solutions in a way we can read them.
- **Handwriting:** We typically receive some submissions which are handwritten. That is fine, but if you submit handwritten solutions, make sure that they are well organized, easy to read and to understand, and that there is not doubt about the interpretation of letters. If you think that this might be hard, please typeset the solutions with a computer. We might reduce points if it's really tough for us and cannot read your submission properly.
- **Language:** Please submit your solutions in English. We have limited capacity of correcting German submissions.

Task 1 (Naïve Bayes) 10 points

Train a Naïve Bayes (the version of the model which we discussed in class) given the following documents annotated with classes c_1 and c_2. Use Add-One-Smoothing. Provide all parameters for a full model specification.

c_1 “happy new year”

c_1 “happy holiday”

c_1 “new year”

c_2 “term starts”

c_2 “work starts”

Given the document

- “happy new year celebrations”

Which class is assigned by the model?

Task 2 (Maximum Entropy Classification) 10 points

Given the following features (without making a difference between upper and lower case) and documents:

weight	feature
λ_1 = 0.2	$f_1(y,x)$ = 1 if “$” in x and y = SPAM
λ_2 = -0.1	$f_2(y,x)$ = 1 if “$” in x and y = HAM
λ_3 = 0.5	$f_3(y,x)$ = 1 if “Nigerian” in x and y = SPAM
λ_4 = -0.2	$f_4(y,x)$ = 1 if “Nigerian” in x and y = HAM
λ_5 = -0.1	$f_5(y,x)$ = 1 if “you” in x and y = SPAM
λ_6 = 0.4	$f_6(y,x)$ = 1 if “you” in x and y = HAM
λ_7 = 0.1	$f_7(y,x)$ = 1 if y = SPAM
λ_8 = 0.0	$f_7(y,x)$ = 1 if y = HAM

Class y	document
SPAM	x_1 = $1 million from Nigerian defense minister
SPAM	x_2 = Please contact Nigerian finance minister
SPAM	x_3 = You won $30,000!
SPAM	x_4 = Buy these Ginsu knifes now.
HAM	x_5 = You should send the Nigerian wildlife report.
HAM	x_6 = Thanks for great dinner. I owe you $20.

Subtask 2.1, 4 points

Calculate $p(\text{SPAM}|x_1)$ with this given configuration of a maximum entropy classifier with the specified features and weights?

Subtask 2.2, 4 points

Calculate the partial derivative of the log-likelihood of all documents with respect to λ_6!

Programming Task 4 (10 points)

The assignment data contains two files `games-train.csv` and `games-test.csv`. These are German app reviews for games (a subset of the data described in `http://www.lrec-conf.org/proceedings/lrec2016/pdf/59_Paper.pdf`).

The files are formatted as follows:

- Column 1: Title of game
- Column 2: Class of review (good or bad)
- Column 3: Title of review
- Column 4: Review text

Title and review texts can be empty.

Subtask 1: 10 points

Implement a text classifier from scratch. You can chose the method from the approaches that we discuss in class. You can use existing code from the other assignments or libraries for tokenization and preprocessing, but the learning and prediction algorithms need to be implemented by you. The easiest approach to implement is probably the Naive Bayes classifier.

The classifier's task is to predict the class (good, bad) stated in Column 2. You can use all information from the training file to build your classifier. You are free in chosing hyperparameters like smoothing, stop-word deletion, stemming, or preprocessing or optimizing those on validation data/via cross validation.

You could implement this as follows (or differently, whatever you prefer):

- Construct a term-frequency data structure using a hashmap or dictionary, where the key is a string representing a term, and the value is an integer. Make such hashmap for every class that you have in the data. Iterate through all tokens and fill this count data structure. Do something similar for the class counts.
- Once done, calculate the prior probabilities and likelihoods by iterating through all terms and normalizing the counts into probabilities.
- When you do calculate c_{map} you only need to step through the tokens of the test instance and retrieve the probabilities from these data structure. Avoid stepping through the test data twice.

As usual, submit your code, well-commented and with an explanation. Which terms have the highest importance, according to the model? List the 100 terms with highest probability in each class together with the term probabilities.

Bonus: 10 points

Implement an evaluation system (from scratch) to Subtask 1 and apply it on `games-test.csv` What is your precision, recall, and F to predict the class good and what is your precision, recall, and F to predict the class bad? Also report the numbers of TP, FP, FN. Discuss your results. Is your result a good result?

Look at some wrongly classified instances and try to understand why they have been wrongly classified (and show them in the submission). Could you come up with ideas how to improve your model, based on these issues?