Homework 5: Precision, recall and relative word frequencies

Florian Fink
Symbolische Programmiersprache

Due: Thursday, December 10, 2020, 16:00

In this exercise you will:

- Implement functions to calculate precision, recall and F1-measure.
- Implement a simple class to calculate relative frequencies.
- Use *Doctest* to validate your implementations.

Use the doctest module to test your implementation of the functions in the module `hw05_evaluation/evaluation`. Run your tests with (this assumes that you are in the src/ directory of your repository): `python3 -m doctest -v hw05_evaluation/evaluation.py`

Exercise 1: Precision and recall

Finish the implementation of the three functions that calculate precision, recall and F1-measure. Each function expects a string (the class that is used) and a list of tuples. Each tuple in the list represents a classifiction with the classification (first entry of the tuple) and its real (ground-truth) value (second entry of the tuple). [5 points]

Make shure that your implementation passes the Doctests. Do not alter any of the Doctests.

1. Implement the precision function that calculates the precision for the given class and classifications. [2 points]
2. Implement the recall function that calculates the recall for the given class and classifications. [2 points]
3. Implement the f1-measure function that calculates the f1-measure for the given class and classifications. [1 point]

Exercise 2: Relative frequencies

Implement the RelFreq class. Finish the Implementation of the constructor and the other three member functions. The constructor expects to word lists that represent the corpus of spam/ham emails and should initialize some attributes to help with the calculation of the member funcions. The other three functions all expect a word and return the relative frequency of the word. [3 points]

Make shure that your implementation passes the Doctests. Do not alter any of the Doctests.

1. Implement the constructor. The constructor expects to word lists that contain the tokens of spam/ham emails. Initialize the class attributes as you see fit and use them for the implementation of the other three functions.

2. Implement the get_rel_freq function that returns relative frequency of the given word in the whole corpus (spam + ham). [1 point]

3. Implement the get_rel_freq_ham function that returns the relative frequency of the given word in the ham corpus. [1 point]

4. Implement the get_rel_freq_spam function that returns the relative frequency of the given word in the spam corpus. [1 point]