Tutorial Information Retrieval

Universität Innsbruck - Department of Computer Science

E. Zangerle

2019-04-10

Exercise Sheet 4

Exercise 1 (Text Classification) [4 Points]

In this tutorial, we will deal with a further text retrieval task: text classification. Particularly, we will look into classifying the author of a given text—the so-called *authorship attribution*-task. To get a first understanding of authorship attribution, please read the following paper on authorship attribution:

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009. doi: 10.1002/asi.21001. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001

Please answer the following questions:

a) *1.5 Points* What are the main feature groups and most widely used features for authorship attribution?

b) *1.5 Points* What are the main approaches for instance-based authorship attribution? Please also discuss the pros and cons of these approaches.

c) *1 Point* Stamatatos also discusses the evaluation of authorship attribution approaches. How can such an evaluation be performed and what are crucial aspects to ensure a fair evaluation?

Exercise 2 (Authorship Attribution) [6 Points]

In this exercise, we will implement an instance-based authorship attribution system. Hence, we aim to classify (attribute) the correct original author for the given test documents. For training and evaluating our system, we will rely on the established Reuter_50_50 dataset[1]. This dataset contains a set of training documents (50 documents for each of the 50 authors) and analogously, a set of test documents (again, 50 documents for each of the 50 authors).
For the preprocessing, classification and evaluation steps, we encourage you to get accustomed with the scikit-learn[2] Python library.

a) *1 Point* For training and testing classification models (supervised machine learning), we firstly have to read the dataset and create a labeled training and test dataset. I.e., each document in either the test or training dataset is assigned a label (the name of the original author). Note that scikit-learn often refers to the documents as `X` and the labels as y[3]. For now it suffices to store training and test documents and labels in four separate lists.

[1] https://archive.ics.uci.edu/ml/datasets/Reuter_50_50

[2] https://scikit-learn.org/stable/

[3] https://scikit-learn.org/stable/tutorial/basic/tutorial.html

b) *1 Point* In a next step, we have perform some preprocessing on the input documents to compute a sensible representation of documents. We will rely on three types of representations: tf/idf vectors, word and character n-grams with $n = 1...3$. For this step, scikit's vectorizers can be quite helpful.

Based on the different representations of documents (and hence, transitively also authors), we aim to perform an authorship attribution task by training an author classifier based on the given training data. For now, we will experiment with three different classifier approaches:

c) *1 Point* Support Vector Machines (SVM)

d) *1 Point* k-Nearest-Neighor classification

e) *1 Point* Decision trees

Train a suitable classifier for each of the three classifier approaches. Shortly describe the basic principles of each of the three classification approaches. What's the best way to find suitable parameters?

f) *1 Point* The scikit-library also provides us with means for evaluating and comparing the implemented classifiers by utilizing the test datasets. Evaluate your models by computing accuracy and the confusion matrix to look into which authors are particularly hard to classify correct or may be mixed up easily by the model. Which configuration (document representation, classifier, parameters) works best? Why?

Important: Submit your solution to OLAT and mark your solved exercises with the provided checkboxes. The deadline ends at 23:59 on the day before the discussion.