

Spamerkennung mit der Naïve-Bayes-Methode

Laden Sie das *Enron-Spammail-Dataset* (Enron1-6) von der Adresse http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html herunter und packen Sie die Dateien aus. Die Verzeichnisse enthalten Unterordner mit den Namen *Spam* und *Ham*, die jeweils Emails mit Spam- und nicht-Spam-Nachrichten enthalten. Jede Datei enthält genau eine Mail, die auf einfache Art tokenisiert wurde. Benutzen Sie Enron1-4 als Trainingsdaten, Enron5 als Development-Daten und Enron6 als Testdaten. (Die Developmentdaten werden Sie erst bei der nächsten Aufgabe benötigen.) Erzeugen Sie drei Ordner *train*, *dev* und *test* mit je zwei Unterordnern *spam* und *ham* und verschieben Sie die Dateien aus den Enron-Verzeichnissen dorthin.

Implementieren Sie dann einen Naïve-Bayes-Klassifikator, den Sie auf den Trainingsdaten trainieren. Der Klassifikator besteht aus einem Trainingsprogramm, welches die Modell-Parameter in einer Datei speichert, und einem Anwendungsprogramm, welches die Parameterdatei einliest und die Testdateien (bzw. Developmentdateien) annotiert.

Die Programme sollen folgendermaßen aufgerufen werden:

```
python3 train.py train-dir paramfile
```

```
python3 test.py paramfile mail-dir
```

train-dir ist ein Verzeichnis mit einem Unterverzeichnis für jede Klasse, das Dokumente enthält. *mail-dir* ist ein Verzeichnis mit Email-Dokumenten.

Ihr Programm sollte so allgemein implementiert werden, dass es auch für Text-Klassifikations-Probleme mit mehr/anderen Klassen anwendbar ist.

Vorüberlegungen

- Welche Teilaufgaben umfasst das Training?
- Was speichern Sie in der Parameterdatei?
- Welche Teilaufgaben umfasst das Anwendungsprogramm?
- Wie verhindern Sie Underflow bei der Berechnung der besten Klasse?

Schicken Sie das fertige Programm, die Liste der ausgegebenen Klassen und die erzielte Genauigkeit an schmid@cis.lmu.de.