Hochschule RheinMain
Prof. Dr. Adrian Ulges

11.11.2021

Anwendungen der KI (WS 21/22)

Aufgabenblatt 4

zu bearbeiten bis: 18.11.2021

Aufgabe 4.1 (Logistische Regression in sklearn)
Die Python-Bibliothek bietet zahlreiche Machine Learning - Modelle, unter anderem **logistische Regression**. Schauen Sie sich die Klasse `LogisticRegression` an: Mit `fit()` trainieren Sie das Modell, mit `predict()` erhalten Sie für neue Daten Vorhersagen. Dieses Tutorial bietet Ihnen kleine Beispiele.

Aufgabe 4.2 (Datensatz vorbereiten)
Das Ziel dieser Woche ist es einen Klassifikator für die Fragen zu implementieren. Bereits Sie dafür zunächst den Datensatz vor. Schreiben Sie ein Programm, das die Fragensammlung (`train_all.csv`) in eine Trainings- (80%), eine Validierungs- (10%) und eine Testmenge (10%) unterteilt. Ihr Programm sollte dafür drei neue `.csv` Dateien (`train.csv, valid.csv, test.csv` erzeugen.
Beachten Sie dabei,

- dass Fragen mit einer ungültigen Antwortkategorie verworfen werden. Beschränken Sie sich auf die 11 Kategorien von Übungsblatt 01.
- dass die Zuordnung der Fragen zu den jeweiligen Mengen zufällig, aber reproduzierbar ist. Wiederholte Ausführungen ihres Programms mit identischer Parametrierung führen zu der gleichen Unterteilung.

Aufgabe 4.3 (Fragenklassifikation)
Implementieren Sie ein ein Programm für die Klassifikation von Fragen. Der Klassifikator (`LogisticRegression`) soll automatisch Fragen die Kategorie ihrer Antwort zuordnen (wie `HUM:ind, LOC:other, ...`).

- Lesen ihre in Aufgabe 4.2 erstellten die Trainings-Fragen ein.
- Erstellen Sie einen Vokabular V aus allen Termen, die in den Trainingsfragen vorkommen. Die Art der Tokenization ist hierbei Ihnen überlassen. Es bietet sich an, die Terme zu stemmen.
- Überführen Sie die Trainingsfragen in boolesche Bag-of-Words-Vektoren der Länge $\#V$: Kommt ein Term nicht in einem Dokument vor, ist das entsprechende Merkmal 0, ansonsten 1 (d.h. die *Häufigkeit* des Vorkommens ist irrelevant).
- Erstellen und trainieren Sie einen `LogisticRegression`-Klassifikator, mit sklearn.

- Schreiben Sie außerdem etwas Code, um den trainierten Klassifikator zu speichern und später wieder zu verwenden. Hier bietet sich das Python-Modul `pickle` an.
- Trainieren Sie auf der Trainingsmenge.

Aufgabe 4.4 (Fragenklassifikation: Analyse)
Erweitern Sie Ihr Programm, so dass die **Fehlerrate** gemessen wird. Vergleichen Sie hierzu das Resultat Ihres Klassifikators mit den echten Kategorien.
Analysieren Sie außerdem die gelernten Gewichte, und geben Sie die 20 wichtigsten Terme/Features jeder Klasse aus (z.B. `hum:ind` $\rightarrow$ ["who", ...]). Hierfür besitzt sklearn's `Logistic-Regression`-Objekt ein Attribut `classifier.coef_`, in dem die Gewichte der einzelnen Features enthalten sind. Geben Sie für jede Klasse die höchsten 20 Gewichte zusammen mit dem jeweiligen Term aus.

Aufgabe 4.5 (Fragen-Klassifikator: Anwendung)
Speichern Sie nach dem Training (siehe oben) Ihren Klassifikator ab. Schreiben Sie dann ein neues Programm, das...

- ihren trainierten Fragen-Klassifikator lädt.
- dem User erlaubt, in der Shell Fragen einzugeben.
- diese Fragen automatisch klassifiziert und die zugehörige Kategorie ausgibt.