Übung 05: Editierdistanz auf Wortebene und Automatische Klassifikation von Vornamen

Programmiertechniken in der Computerlinguistik II, FS 17

Abgabedatum: 9. Mai 2017, 23:59

Hinweise zur Abgabe

- Bitte gib jedes Python-Programm in einer eigenen Datei ab, die die Dateiendung `.py` hat und *ausführbaren* Python-Code enthält. Die Programme sollten importierbar sein.
- Gib all deine Antworten, welche keine Skripts sind, in einem PDF-Dokument oder einer Plain-Text-Datei (benannt nach dem Schema *vorname_nachname_uebung0X.pdf*) ab.
- Geize nicht mit Kommentaren direkt im Programm-Code, wo Erläuterungen angebracht sind. Umfangreiche Erklärungen werden hingegen besser in einer separaten README-Datei mitgeliefert (vorzugsweise Plain-Text oder PDF).
- Halte dich an die Vorgaben des Python Style Guide! Grobe Verstösse werden mit Punkteabzug geahndet.
- Um das Hochladen der Abgabe auf OLAT zu erleichtern, kannst du die Dateien mit `zip` (oder einem anderen verbreiteten Format) archivieren / komprimieren.

1 Editierdistanz auf Wortebene

Implementiere ein Programm, welches die minimale Editierdistanz zwischen zwei tokenisierten Sätzen (= Wortlisten) ermittelt. Neben der Editierdistanz soll – Analog zum Beispiel in Jurafsky und Martin auf Buchstabenebene (Abbildung 2.12 im Kapitel 2.4; s. OLAT) – die quasi-Wortalignierung und alle angewendeten Operationen (Einfügen, Löschen, Ersetzen) ausgegeben werden.
Zum Beispiel sieht der Output für die zwei Wortlisten
['This', 'is', 'nice', 'cat', 'food', '.']
und
['this', 'is', 'the', 'nice', 'cat', '.']
wie folgt aus:

```
This is ∗   nice cat food .
|    | |   |    |   |    |
this  is the nice cat ∗    .
S       I             D
```

Edit distance: 3

Dies soll dein Program für die folgenden drei Listen-Paare tun:

['This', 'is', 'nice', 'cat', 'food', '.']
['this', 'is', 'the', 'nice', 'cat', '.']

['The', 'cat', 'likes', 'tasty', 'fish', '.']
['The', 'cat', 'likes', 'fish', 'very', 'much', '.']

['I', 'have', 'adopted', 'cute', 'cats', '.']
['I', 'have', 'many', 'cats', '.']

Abzugeben: Ein importierbares Python-Skript.

2 Automatische Klassifikation von Vornamen

Im Aufgabenordner findest du das Python-Skript *aufgabe2_vorlage.py* welches einen einfachen Naive-Bays-Klassifikator (von NLTK) implementiert, welcher Vornamen ihrem Geschlecht zuordnet. Deine Aufgabe besteht darin, das Skript zu vervollständigen. Gehe dabei wie folgt vor:

- Vervollständige die Methode *get_train_and_test_data*. Diese Methode soll die Daten der Männer- und Frauennamen in einen Test- und Trainings-Teil aufteilen. Wähle dazu eine sinnvolle Verteilung. Die Trainings- und Test-Daten sollen als Liste zurückgeben werden.
- Vervollständige die Methode *gender_features*. Diese Methode soll möglichst gute Features enthalten, welche die Vornamen-Klassifikation optimieren.
- Vervollständige die Methode *get_training_and_test_labeled_features*. Dabei solltest du jedem Namen in den Trainings- und Test-Daten das ensprechende Label zuordnen ('male' oder 'female'). Diese Label-Listen kannst du dann verwenden, um eine Trainings- und Test-Feature-Liste mit Labels zu erstellen und zurückzugeben. Diese sollte in der Form [({feature1_name : feature1_value, ...}, label), ...] sein. Die beiden Listen werden für das Trainieren und die Evaluation des Klassifikators benötigt.
- Vervollständige die Methode *evaluation*. Diese Methode soll die Accuracy, Precision, Recall und F-Measure sinnvoll auf der Konsole ausgeben.

Dein Programm sollte für die Accuracy mindestens einen Wert von 0.7 erreichen.
Wir haben aus dem euch zur Verfügung gestellten Datensatz ein eigenes Test-Set extrahiert und werden alle Abgaben mit diesem zusätzlich testen. Unter allen Abgaben werden wir das Programm mit den besten Klassifikationsergebnissen mit einem kleinen Überraschungspreis belohnen. Es lohnt sich daher, nicht nur das Minimum zu erreichen :)

Hinweis: Für die Aufgabe lohnt es sich, eine Kreuzvalidierung durchzuführen. Ansonsten droht die Gefahr von Overftting auf einem einzigen Extrakt zufällig ausgewählter Daten.

Abzugeben: Das vervollständigte Python-Skript.

Reflexion/Feedback

a) Fasse deine Erkenntnisse und Lernfortschritte in zwei Sätzen zusammen.

b) Wie viel Zeit hast du in diese Übungen investiert?