Programmierkurs Python I – WS 09/10

Übung 5

---

1 Wörter und POS-Tags (4 + 2 Punkte)

Auf der Homepage (unter „Code-Beispiele“) findest du die Datei `A00-pos-lines`, ein Ausschnitt aus dem British National Corpus. In der Datei sind alle Wörter mit POS-tags versehen, in der Form `wort$tag`. Pro Zeile gibt es genau ein Wort-POS-Paar. „Wörter“ können aus mehreren Strings bestehen (Mehrwortausdrücke), in dem Fall sind die mit Leerzeichen getrennt. POS-Tags sind immer nur ein Wort. Satzzeichen zählen als ein Wort (und sind getagt). Schreibe ein Programm, das die Datei einliest (entweder von der Homepage oder von Eurem Dateisystem), und zählt, welches Wort wie oft vorkommt. Als Ausgabe solltet Ihr eine Datei schreiben, das für jedes Wort seine Häufigkeit auflistet und wie oft es mit welchem POS-tag vorkommt, etwa so (hypothetische Zahlen):

```
name   17      NN1:12  VVT:5
```

Bonusaufgabe 1: Richte Dein Programm so ein, dass man auf der Kommandozeile entweder eine URL oder einen Dateinamen angeben kann. Dein Programm soll dann diese Eingabe erkennen, mit dem jeweils richtigen Konstrukt öffnen und wie oben bearbeiten.

Bonusaufgabe 2: Auf der Homepage findest Du außerdem die Datei `A00-pos`, die im Original-BNC-Format gedruckt ist. Sie unterscheidet sich von der anderen Datei nur darin, dass mehrere Wort-Pos-Paare auf einer Zeile stehen können. Erweitere Dein Programm so, dass es auch diese Datei wie oben erklärt verarbeiten kann.

2 Der Goldkäfer (6 Punkte)

„Der Goldkäfer“ ist ursprünglich eine Geschichte von E. A. Poe. Es geht unter anderem darum, wie jemand einen Text anhand von Buchstabenhäufigkeiten entschlüsselt. Man geht davon aus, dass im verschlüsselten Text jedes Zeichen einfach durch ein anderes ersetzt wurde. Die Taktik dabei ist, große Textmengen von beliebigen Texten zu betrachten und zu ermitteln, welches Zeichen wie häufig vorkommt. Dann bestimmt man die Häufigkeitsverteilung der Buchstaben im gegebenen Text,

und übersetzt das häufigste Zeichen im Text mit dem häufigsten Zeichen im Korpus gleich, das zweithäufigste Zeichen mit dem zweithäufigsten aus dem Korpus, usw.

Implementiere einen Algorithmus, der diese Art der Dechiffrierung benutzt. Zum Entschlüsseln haben wir einen Text encodiert, (brown-sample-enc.txt), den ihr von der Homepage laden könnt. Satzzeichen sind nicht verändert worden. Um die Buchstabenhäufigkeiten zu bestimmen, könnt ihr das Korpus brown.txt benutzen. Lest zuerst das Referenzkorpus (brown.txt) aus, speichert die Buchstabenhäufigkeiten, und versucht dann, die encodierte Datei auszulesen und eine neue Datei mit dem decodierten Text zu schreiben. Denk daran, dass hier relative Häufigkeiten gefragt sind - also die absolute Anzahl eines Zeichens im Text, geteilt durch die Anzahl der Zeichen im Text.

3 Listen und Ausnahmen (4 Punkte)

Auf den Folien wurde eine Ausnahmen-Klasse `MyIndexError` definiert, die dafür gedacht ist, detailliertere Informationen zu liefern, wenn ein Index einer Liste aufgerufen wurde, der in der Liste nicht belegt ist.

Implementiere eine Klasse `MyList`, die von `list` erbt und anstatt eines `IndexError` als Ausnahme eine Instanz von `MyIndexError` wirft. Dafür müsst ihr die Methode `__getitem__(self,index)` implementieren (bzw. überschreiben); diese wird aufgerufen, wenn man auf ein Listenelement mit `liste[i]` zugreift. Damit man die Listenobjekte sonst verwenden kann wie normale Listen, muss bei Zugriff auf einen gültigen Index die `__getitem__`-Methode von `list` benutzt werden; falls dabei ein Fehler passiert, soll dem Benutzer ein `MyIndexError` angezeigt werden

---

Abgabe bis Donnerstag, 3.12.09, 14:00 per Mail an
`regneri@coli.uni-sb.de`
`lcarolyn@coli.uni-sb.de`