Programmiertechniken in der Computerlinguistik I HS 2016

Aufgabenstellung: Simon Clematide/Martin Volk

Prüfung vom 12. Januar 2017
Institut für Computerlinguistik, Universität Zürich

Vorname _____ Matrikelnummer _____

Nachname _____

Für Studierende der folgenden Studiengänge:

☐ BA - Studiengang Computerlinguistik (Phil. Fakultät)

☐ BA - Studiengang Computerlinguistik und Sprachtechnologie (Phil. Fakultät)

☐ Anderer Studiengang: _____

Aufgabe Nr.:	1	2	3	4	5	6	7	8	9	**Summe**
Punktzahl:	16	21	9	5	3	3	3	10	20	90
Davon erreicht:										

Note SU: _____ Note SP: _____

Endnote: _____ Bestanden: ☐ Ja ☐ Nein

Wichtige Hinweise

- Maximale Punktzahl: 90
- Pro Punkt sollte ungefähr 1 Minute gebraucht werden.
- Bitte schreiben Sie in einem knappen, aber verbalen Stil (keine Stichwortsammlungen). Bei inhaltlichen Auswahlsendungen, wo einfach alles spontan hingeschrieben und Falsches wie Korrektes munter vermischt wird, behalten wir uns Abzüge vor.
- Erlaubtes Hilfsmittel 1 A4-Blatt mit eigenen Notizen.
- Die Antworten können **deutsch** (bevorzugt) oder **englisch** sein. Bei sprachlichen Verständnisfragen melden Sie sich bei der Aufsichtsperson.

Auf jedes separate Blatt mit Lösungen den Nachnamen schreiben! Viel Erfolg!

1. Auf der Kommandozeile mit dem Text+Berg-Korpus arbeiten (16 Punkte)

In einem Verzeichnis seien **eine kleine Readme-Datei** namens `Readme.txt` sowie alle **mehrsprachigen Jahrbücher** des Text+Berg-Korpus wie in der Übung 0 (z.B. `SAC-Jahrbuch_1930_mul_columns.txt`). Es gibt Tokenzeilen und Strukturzeilen. Tokenzeilen enthalten in 3 tabulator-separierten Spalten das Token, das POS-Tag und das Lemma. Strukturzeilen markieren den Anfang von Zeitschriftenartikeln und Sätzen, enthalten keine Tabulatoren und beginnen mit "`<article `" bzw. "`<s `".

```
<article n="1">
<s lang="de" n="1-1">
Vorwort           NN              Vorwort
<s lang="de" n="1-2">
Das               ART             d
Jahrbuch          NN              Jahrbuch
XXXVI             CARD            @card@
weicht            VVFIN           ab+weichen
ab                PTKVZ           ab
.                 $.              .
<s lang="de" n="1-3">
Verlustigieren    VVFIN           unk
```

Alle Dateien sind gemäss Muster "`SAC-Jahrbuch_1930_mul_columns.txt`" benannt. Die folgenden Fragen setzen voraus, dass die Befehle im Verzeichnis mit den Jahrbüchern ausgeführt werden. Bei grep können Sie die Option "-E" oder "-P" hinschreiben, um die erweiterten regulären Ausdrücke zu verwenden, welche Sie auch von Python kennen.

(a) Wie kann man den Inhalt aller Jahrbücher in eine Datei namens SAC.TEXT schreiben? [2]

.....

(In den folgenden Teilaufgaben, darf die Datei SAC.TEXT alternativ zu den einzelnen Jahrbüchern verwendet werden.)

(b) Welche wesentliche Information können wir dem Output des folgenden Befehls entnehmen? [2]

```
grep '^<s ' SAC.TEXT | wc
```

.....

.....

.....

(c) Wie kann man sich alle Zeilen anzeigen lassen, welche ein unbekanntes Vollverb (VVFIN, VVINF, VVPP, VVIMP) enthalten? (Unbekannte Wörter haben das Lemma `unk`). [3]

.....

(d) Wie kann man die Menge (!) aller Zeilen ausgeben, die potentiell eine römisch geschriebene Zahl enthalten? [3]

.....

(e) Wie kann man diejenigen Zeilen ausgeben, bei denen die Wortform und das Lemma identisch sind? [3]

.....

(f) Wie kann man alle Zeilen ausgeben, welche das Token "Jungfrau" oder "Mönch"mit dem POS-Tag "NN" aufweisen? Es sollen jeweils die beiden Zeilen vor einer Trefferzeile mitausgegeben werden. [3]

.....

2. Python-Theorie und -Praxis (21 Punkte)

Kreuzen Sie **alle korrekten Aussagen** an. Jede richtige Antwort ergibt 0.75 Punkte, jede **falsche** Antwort ergibt 0.75 Punkte **Abzug**. Man kann minimal 0 Punkte machen pro Teilaufgabe. Es können beliebig viele Antworten richtig oder falsch sein pro Teilaufgabe.

(a) Zeichen und Zeichenketten [3]

- ☐ Die Zeichenkette `r"\\t"` hat eine Länge von 3 Buchstaben.
- ☐ Texte von europäischen Sprachen sollten besser in ISO-LATIN-1-Kodierung abgespeichert werden als in UTF-8.
- ☐ Eine UTF-8-Bytefolge `z = b'b\xc3\xa4'` wird mittels `z.decode('utf-8')` in einen normalen Unicode-String konvertiert.
- ☐ Zeichenketten, die mit doppelten Anführungszeichen `"` begrenzt sind, dürfen sich maximal über 1 Zeile im Quellkode erstrecken.

(b) Reguläre Ausdrücke in Python (mit Modul re importiert) [3]

- ☐ Der reguläre Ausdruck `r'\w+'` matcht in Python 3 auch Umlaute.
- ☐ Das Flag `(?x)` erlaubt Zeilenkommentare und Leerzeichen, welche ignoriert werden.
- ☐ Der Ausdruck `re.sub(r'e','ie','bittere')` evaluiert zu `'bittierie'`.
- ☐ Der Ausdruck `re.sub(r'(ere|e)$','','bittere')` evaluiert zu `'bitt'`.

(c) Funktionen [3]

- ☐ Funktionen mit einem yield-Statement funktionieren wie Generatoren.
- ☐ Doc-Strings in Funktionen können direkt nach dem Funktionskopf oder ganz am Schluss der Funktion stehen.
- ☐ Eine Funktion kann höchstens *ein* return-Statement enthalten.
- ☐ Alle Funktionen, die innerhalb einer Funktionsdefinition f verwendet werden, müssen im Quellkode vor der Definition von f stehen.

(d) Veränderliche Datenstrukturen und Dictionaries [3]

- ☐ Die beiden Ausdrücke `dict('a'=1,'b'=2)` und `{'a':1,'b':2}` ergeben dasselbe Dictionary.
- ☐ Die Schlüssel von Dictionaries müssen aus Zeichenketten bestehen.
- ☐ Die Werte von Dictionaries müssen unveränderliche Datenstrukturen sein.
- ☐ Mengen (Datentyp `set`) sind veränderlich.

(e) Objekte [3]

- ☐ Listen sind in Python keine Objekte, weil sie veränderbar sind.
- ☐ Funktionen sind so etwas wie aufrufbare (*callable*) Objekte.
- ☐ Jedes Objekt ist eine Instanz einer Klasse.
- ☐ Generatoren sind ebenfalls Objekte.

(f) Klassen [3]

- ☐ Vordefinierte Klassenbezeichner wie `dict` sind selbst auch Objekte.
- ☐ Eine Oberklasse hat grundsätzlich alle Methoden von ihrer Unterklasse.
- ☐ Von der Klasse `object` kann man beliebig viele Instanzen erzeugen.
- ☐ Die Klasse `object` hat keine Oberklasse.

(g) Identität und Wertgleichheit [3]

- ☐ Der Ausdruck `list("abc") is ['a','b','c']` evaluiert zu `True`.
- ☐ Der Ausdruck `list("abc") == ['a','b','c']` evaluiert zu `True`.
- ☐ Der folgende Kode schreibt `True` heraus.

```
a = ['Ein','Wort']
b = a
print(b is a)
del b[0]
```

- ☐ Nach Ausführen der obigen 4 Zeilen hat `a` genau 1 Element.

3. Manipulation von Listen und Dictionaries (9 Punkte) 9

Das Folgende sind interaktive Eingaben im Python-Interpreter, welche nacheinander ausgeführt werden. Schreiben Sie die Ausgabe der Print-Funktionen auf die entsprechenden Linien oder FEHLER und eine kurze Begründung, falls ein Ausdruck gar nicht evaluiert werden kann.

Eingaben 1:

```
>>> x1 = 'eins zwei drei'
>>> print(x1[-1])                    Ausgabe: _____
>>> x1 = x1.split()
>>> print(x1[1:2])                   Ausgabe: _____

>>> print(sorted(x1,reverse=True))   Ausgabe: _____
>>> del x1[0]
>>> print(x1[0])                     Ausgabe: _____
>>> x1.append('eins')
>>> print(x1)                        Ausgabe: _____
>>> del x1
>>> print(x1)                        Ausgabe: _____
```

Eingaben 2:

```
>>> y = {'a':12,'b':13}
>>> print(y[a])                      Ausgabe: _____
>>> y['c'] = 'cacao'.count('c')
>>> print(max(y))                    Ausgabe: _____
>>> del y['a']
>>> print(sorted(y.items()))         Ausgabe: _____
```

4. `global` (5 Punkte) Weshalb kann das Schlüsselwort `global` nützlich sein? Geben Sie ein Beispiel! 5

.....

.....

.....

.....

.....

.....

.....

5. ASCII vs UTF-8 (3 Punkte) Welche Konzepte stecken hinter den Akronymen? Was unterscheidet/-verbindet sie? 3

.....

.....

.....

.....

6. Python-3-Zeichenketten vs. Bytefolgen (3 Punkte) Wie kann man beide Dinge als Literale notieren? Wozu kann man sie gebrauchen? Was unterscheidet sie? [3]

.....

.....

.....

.....

7. Die eingebaute Funktion `len()` (3 Punkte) [3]
Erklären Sie für drei verschiedene Datentypen kurz, was die Funktion `len()` berechnet.

.....

.....

.....

.....

8. Objektorientierten Python-Kode verstehen (10 Punkte) Hans Misteli war etwas abgelenkt und im Stress, als er in der Vorlesung den folgenden Kode zum KWIC-Konkordanzer abgetippt hat.

(a) Versuche mindestens 5 verschiedenartige Fehler im Kode zu korrigieren. [5]

(b) Markiere und beschrifte jeweils die verlangte Anzahl Exemplare von folgenden Konstrukten im Kode mit den korrekten Buchstaben in der folgenden Weise: 1 Modul ⓐ, 1 Package ⓑ, 1 Methodendefinition ⓒ, 3 Klassennamen ⓓ, 1 Methodenaufruf ⓔ, 1 Instanzvariable ⓕ, 1 lokale Variable ⓖ (keine Instanzvariable), 1 Formatierungsausdruck ⓗ, 2 Instanzen von selbstdefinierten Klassen ⓘ. [5]

```
from * import nltk, re

class RegexStemmer(object):
   def __init(r=r'^.*?(ing|ly|ed|ious|ies|ive|es|s|ment)?$'):
      return self._r = r

   def stem(word):
      m = re.macht(self._r, word)
      m.group(1)


class IndexedText(object):

   def __init(self, stemmer, text):
      self._text = text
      self._stemmer = stemmer
      self._index = nltk.Index((self._stem(word), i)
                        for (j, word) in enumerate(text))

   def _stem(self, word):
      return self._stemmer.stem(word).lower


   def concordance(self, word, width:40)
      key = self._stem(word) # stemmed keyword
      wc = width//4        # words of context
      for i not in self._index[key]:
         lcontext = ' '.join(self._text[i-wc:i])
         rcontext = ' '.join(self._text[i:i+wc])
         ldisplay = '%*s' % (width, lcontext[-width:])
         rdisplay = '%-*s' % (width, rcontext[:width])
         print(ldisplay, rdisplay)
```

```
regex_stemmer = RegexStemmer
text == nltk.corpus.webtext.words('grail.txt')

regex_index = IndexedText(regex_stemmer, text)

print("\nKWIC mit Regex-Stemmer\n")
regex_index.concordance['seem']
```

9. SAC-Spaltenformat transformieren (20 Punkte) Gegeben seien die Text+Berg-Dateien wie in Aufgabe 1. Schreiben Sie ein vollständiges Pythonscript `sac2detext.py`. Wenn es aufgerufen wird auf der Kommandozeile mit einem multilingualen Jahrbuch, soll es jeden **deutschsprachigen Satz** auf einer separaten Zeile ausgeben, der mindestens 3 Token enthält. Jeder Satz besteht nur aus seinen Tokens, welche durch Leerzeichen getrennt sind. Leicht fiktionaler Beispielaufruf: **20**

```
$ sac2detext.py SAC-Jahrbuch_1930_mul_columns.txt
Das Jahrbuch XXXVI weicht ab .
Verlustigieren sich die Clubisten am Tödi ?
```

Hinweis: Wenn Ihre Lösung unnötig viele Zeichenketten im Speicher für den "Garbage-Collector" erzeugt, gibt es Abzug. Das letzte Token eines Satzes darf von einem Leerzeichen gefolgt sein.