Python für Linguisten

Dozentin: Wiebke Petersen & Co-Dozentin: Esther Seyffarth

6. Foliensatz
NLTK

NLTK: Natural Language ToolKit

- Entwickler: Steven Bird, Ewan Klein und Edward Loper
- Website: `http://www.nltk.org/`
- NLTK Buch: `http://www.nltk.org/book/`

Installationshinweise

- Installation von NLTK: `http://www.nltk.org/install.html`
- zusätzlich benötigte Pakete:
  - NumPy (für viele Berechnungen benötigt): `http://sourceforge.net/projects/numpy/files/NumPy/`
  - matplotlib (Plotten von Funktionen): `http://matplotlib.org/`

NLTK benutzen

- Zunächst NLTK importieren: `>>> import nltk`
- Korpora downloaden mit `>>> nltk.download()`
- einen Text nutzen, zB: `>>> from nltk.book import text1`
- Beispiel: Kookkurrenzen mit Kontext: `>>> text1.concordance('whale')`
- Beispiel: Wörter, die in einem ähnlichen Kontext erscheinen:
  `>>> text1.similar('whale')`

Aufgabe:
Vergleichen Sie für Text 1 und Text 2 mit
`text1.similar('monstrous')` *`#bzw. text2`* die Wörter, die in ähnlichen Kontexten wie `monstrous` erscheinen. Was fällt auf? Wie erklären Sie sich das?

Textstatistik mit Gutenberg

Die Gutenberg-Texte wurden bereits tokenisiert:

```
>>> from nltk.corpus import gutenberg
>>> nltk.corpus.gutenberg.fileids() # Liste aller Gutenbergtexte
>>> r = gutenberg.raw('carroll-alice.txt')
>>> w = gutenberg.words('carroll-alice.txt')
>>> s = gutenberg.sents('carroll-alice.txt')
```

Schauen Sie sich an, wie die Variablen `r`, `w`, `s` belegt sind (Vorsicht, schauen Sie sich nur einen Ausschnitt an). Ermitteln Sie die durchschnittliche Satzlänge, die durchschnittliche Wortlänge und das durchschnittliche Vorkommen eines Types im Text.

Benutzung des CMU-Korpus

Zunächst muss das Korpus heruntergeladen werden.

```
>>> from nltk.corpus import cmudict
>>> cmu_dict = dict(cmudict.entries())
```

Suchen Sie alle Wörter heraus, die orthografisch auf ‘ough’ enden.
Was fällt bei der Aussprache auf?

Verteilungen plotten

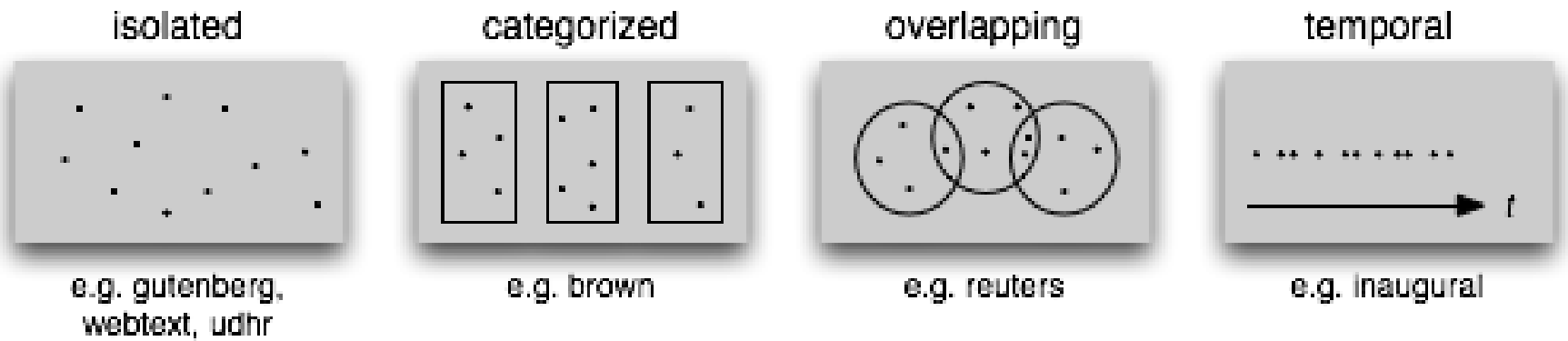
Mit FreqDist:

```
# jede Funktion muss zunächst importiert werden
>>> from nltk import FreqDist
>>> fdist1 = FreqDist(text1)
>>> fdist1.most_common(50)
>>> fdist1.plot(50)
>>> fdist1.plot(50,cumulative=True)
```

Mit dispersion_plot:

```
>>> from nltk.book import * # importiert alles aus dem Buch
>>> text1.dispersion_plot(['Moby', 'whale', 'sea', 'ship'])
>>> text4.dispersion_plot(['freedom', 'citizen','safety','democracy','sec
```

Korpustypen



Quelle: `http://nltk.org/book/ch02.html#fig-text-corpus-structure`

Korpora

- Beispiele: Brown Corpus, Gutenberg Corpus, Reuters Corpus
- Treebanks: Dependency Treebank, Penn Treebank (selections), Floresta Treebank
- multilingual: Genesis Corpus, Univ Decl of Human Rights (UDHR)
- monitoring/zeitlich: Inaugural Address Corpus
- gesprochene Sprache: Switchboard Corpus, TIMIT Corpus (selections)
- informelle Sprache: Chat-80-Corpus (Chatlogs), NPS Chat Corpus

Beispiel: Brown Korpus

- Nutzung: `>>> from nltk.corpus import brown`
- Dateien: `>>> brown.fileids()`
- Kategorien: `>>> brown.categories()`
- Raw: `>>> rawtext = brown.raw()`
- Worte: `>>> words = brown.words()`
- Sätze: `>>> sentences = brown.sents()`
- Gesamtzahl aller Typen:
  ```
  >>> len(set([w.lower() for w in brown.words()]))
  ```
- Eingeschränkt auf Kategorien:
  ```
  len(set([w.lower() for w in brown.words(categories='mystery')]))
  ```