TOKENIZATION, STEMMING, STOPPING

VORGESTELLT VON STEFANIE TIRKOVA

INHALT

- Tokenisierung
  - Probleme der Tokenisierung
  - Tokenisierung in Python - Beispiele
- Stemming
  - Porter-Stemmer-Algorithmus
- Zipfs Gesetz
- Stoppwörter
- Tokenization und Stemming Bibliotheken für Python
- Diskussion

TOKENISIERUNG

- Computerlinguistik: die Segmentierung eines Textes in Einheiten der Wortebene (manchmal auch Sätze, Absätze o. Ä.)
- Zerlegung in Wörter: White-Space-Tokenisierung

```
So funktioniert die Tokenisierung.    >> So
                                         Funktioniert
                                         die
                                         Tokenisierug
```

PROBLEME DER TOKENISIERUNG

- 多少字在这里 ???
- Was ist ein Wort?
  - Mehrwortlexemen (San Francisco)
  - speziell Eigennamen (Klaus-Rüdiger)
  - Währungsangaben ($5)
  - nicht-ganze Zahlen (3,14)
  - Abkürzungen (z. B., d. h., Prof.in)

TOKENISIERUNG IN PYTHON - BEISPIELE

- mit *.split()*

```
satz = "Das ist eine einfache Tokenisierung."
woerter_1 = satz.split(" ")

print woerter_1
```

```
['Das', 'ist', 'eine', 'einfache', 'Tokenisierung.']
```

- *mit* Regulären Ausdrücken

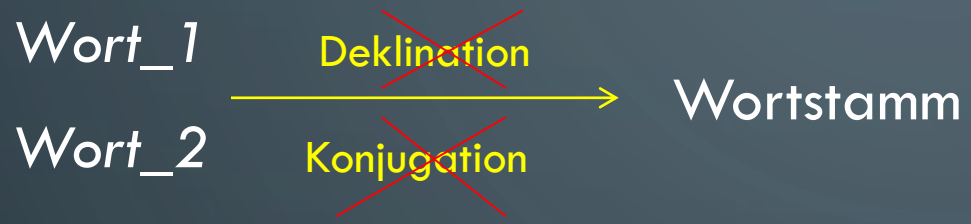
```
import re
satz = "Das ist eine einfache Tokenisierung."
einfach_regex = re.compile("\w+")
woerter_2 = einfach_regex.findall(satz)

print woerter_2
```

```
['Das', 'ist', 'eine', 'einfache', 'Tokenisierung']
```

STEMMING

- Stammformreduktion, Normalformenreduktion



Beispiel:

Wortes, Wörter ⟶ *Wort*

aß, gegessen ⟶ *ess*

PORTER-STEMMER-ALGORITHMUS

- Funktionsweise:

  Wort → [C](VC)^m[V]

  C = ein oder mehrere Konsonanten

  V = ein oder mehrere Vokale

Beispiele:

tr-ee, t-o (m=0)

w-eb (m=1)

b-etw-een (m=2)

t-ok-en-iz-at-ion (m=5)

- Verkürzungsregeln:

  "sses" → "s"

  "ies" → "i"

  "s" → " "

  "y" → "i"

Beispiele:

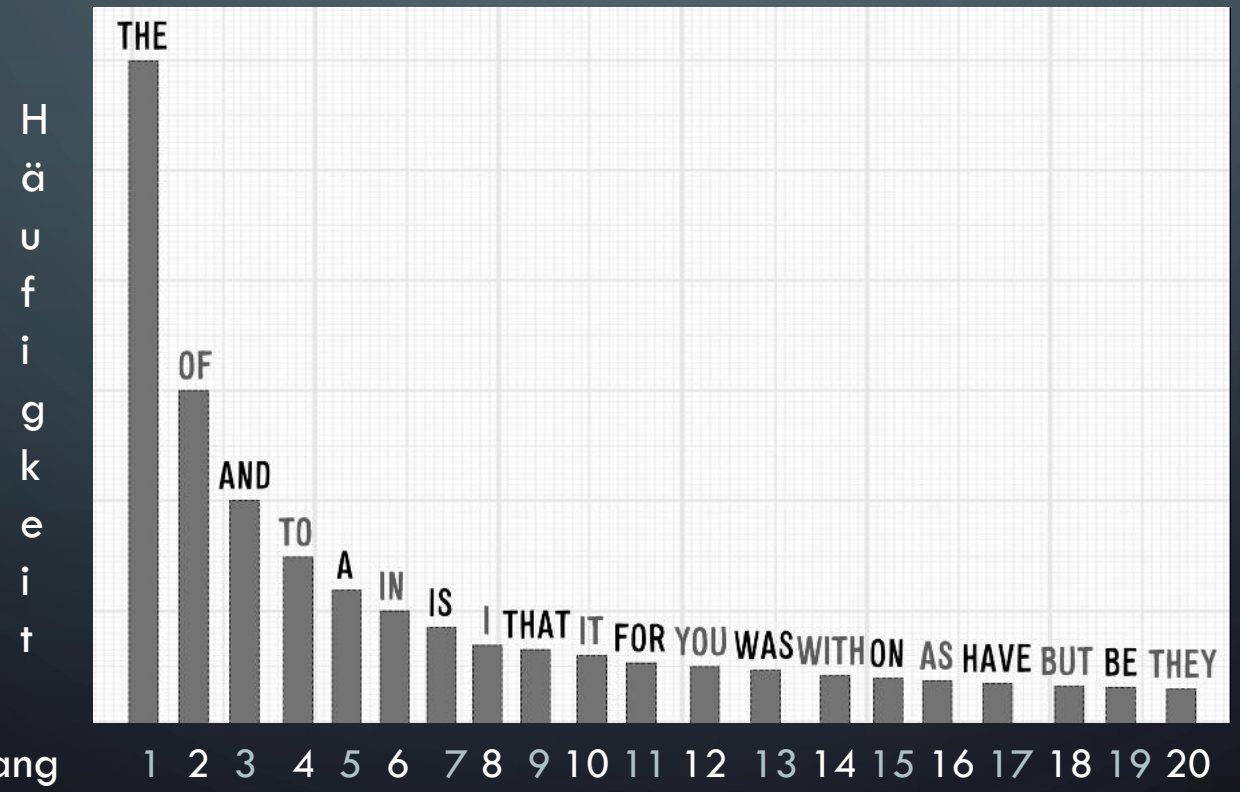
*librar**ies** → librar**i***

*librar**y** → librar**i***

ZIPFS GESETZ

- $p(n) \sim \frac{1}{n}$

$$Worthäufigkeit \sim \frac{1}{Rang}$$



STOPPWÖRTER

- 'der', 'die', 'das', 'einer', 'eine', 'ein' 'und', 'oder', 'doch', 'an', 'in', 'von'
- keine Relevanz → kein Index
- Stopword Removal – Stoppwortlisten (NLTK)

TOKENIZATION UND STEMMING BIBLIOTHEKEN FÜR PYTHON

- NLTK - Natural Language Toolkit
- SpaCy

```
import nltk
satz = "Das ist eine bessere Tokenisierung."
tokens = nltk.word_tokenize(satz)
print tokens
```

```
['Das', 'ist', 'eine', 'bessere', 'Tokenisierung', '.']
```

WAS HABT IHR VERSTANDEN?

- Tokenisierung = ?
- Welche sind die möglichen Weisen, die Tokenisierung in Python durchzuführen?
- Stemming = ?
- Was für ein Verhältnis stellt das Zipfs Gesetz dar? Zwischen was?
- Beispiele für Stoppwörter?
- Warum werden diese bei einer Suche nicht beachtet?

QUELLEN

https://de.wikipedia.org/wiki/Tokenisierung (08.07.16)

https://en.wikipedia.org/wiki/Lexical_analysis (08.07.16)

http://www.delph-in.net/courses/07/nlp/Gre:Tap:94.pdf (08.07.16)

https://de.wikipedia.org/wiki/Stemming (08.07.16)

https://www.youtube.com/watch?v=fCn8zs912OE (09.07.16)

https://de.wikipedia.org/wiki/Zipfsches_Gesetz (09.07.16)

https://de.wikipedia.org/wiki/Stoppwort (10.07.16)

https://de.wikipedia.org/wiki/Porter-Stemmer-Algorithmus (10.07.16)

http://www.nltk.org/ (11.07.16)

https://spacy.io/ (11.07.16)

https://repl.it/

Danke für die Aufmerksamkeit!