

# Wortschatz Zeitgeist

Wolfgang Otto, Thomas Döring, Max Kießling

Seminar Anwendungen der linguistischen Informatik

16. Juni 2015

Motivation

Algorithmen

Vergleich und Auswertung

# Motivation

---

# Wortschatzprojekt

---

- Bla
- Foobar
- Batz

## Definition

Definition

## Example

Beispiel

# Algorithmen

---

## Relative Häufigkeit

---

*Idee:* Tokens, deren relatives Auftreten am gewählten Tag im Verhältnis zum relativen Auftreten im Referenzzeitraum (2014) besonders groß ist, sind interessante Wörter.

*Formel:*

$$\text{sigfreqratio}(w) = \frac{\frac{k_{\text{day}}}{n_{\text{day}}}}{\frac{k_{2014}}{n_{2014}}} \quad (1)$$

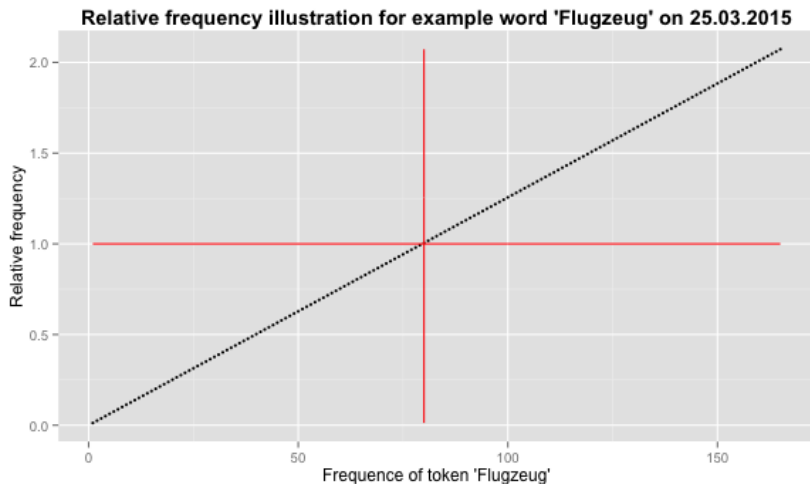
$k_{\text{day}}$ : Frequenz des Tokens an einem Tag

$n_{\text{day}}$ : Summe der Frequenzen aller Tokens eines Tages

$k_{2014}$ : Frequenz des Tokens im Referenz Zeitrahmen (2014)

$n_{\text{day}}$ : Summe der Frequenzen aller Tokens im Referenzzeitrahmen (2014)

# Relative Häufigkeit



# Relative Häufigkeiten - Bemerkungen

---

- Erster Ansatz
- Einfache Implementierung
- Selten Auftretende Wörter werden gegenüber anderen interessanten Wörtern bevorteilt
- Positiv: Hochfrequente Worte werden selten hoch gerankt



# Poisson als Maß

---

*Idee:* Modellierung der Wahrscheinlichkeit eine bestimmte Frequenz eines Tokens zu sehen. Wenn die Tagesfrequenz eines Tokens sehr unwahrscheinlich ist, ist das Token interessant.

*Annahme:* Diese Wahrscheinlichkeiten sind Poisson-Verteilt.

*Formel der Poisson-Verteilung allgemein:*

$$P_{\lambda}(k) = \frac{\lambda^k}{k!} \cdot e^{-\lambda} \quad (2)$$

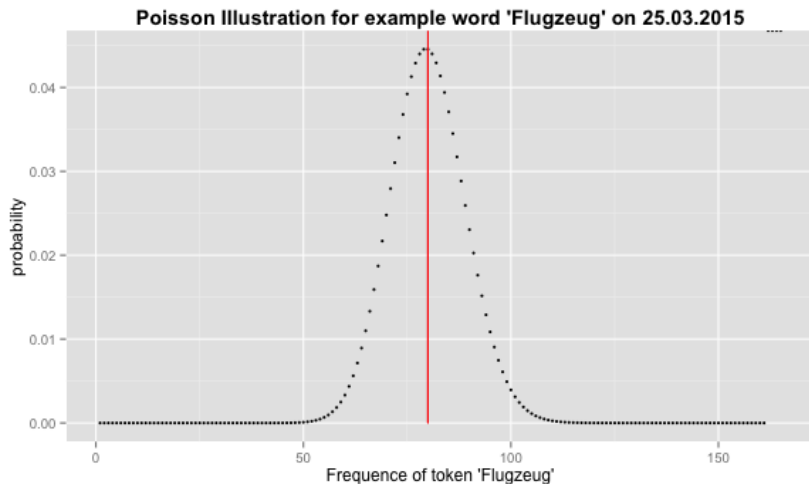
$\lambda$ : Welche Frequenz wird erwartet

(relativer Anteil im Referenzkorpus  $\cdot$  Umfang des Tageskorpus)

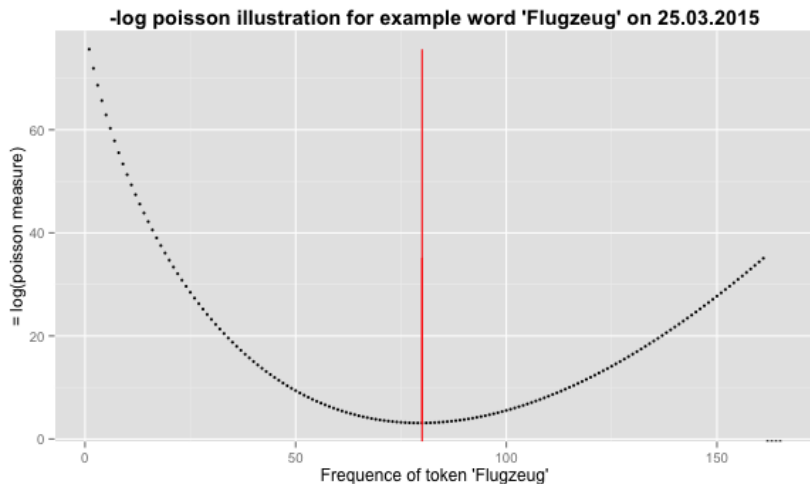
$k$ : tatsächliches Auftreten von einem Wort  $k$

$P_{\lambda}(k)$ : Erwartete Wahrscheinlichkeit meine Beobachtung  $k$

# Poisson Verteilung



# Poisson Verteilung II



# Poisson als Maß: Implementierung

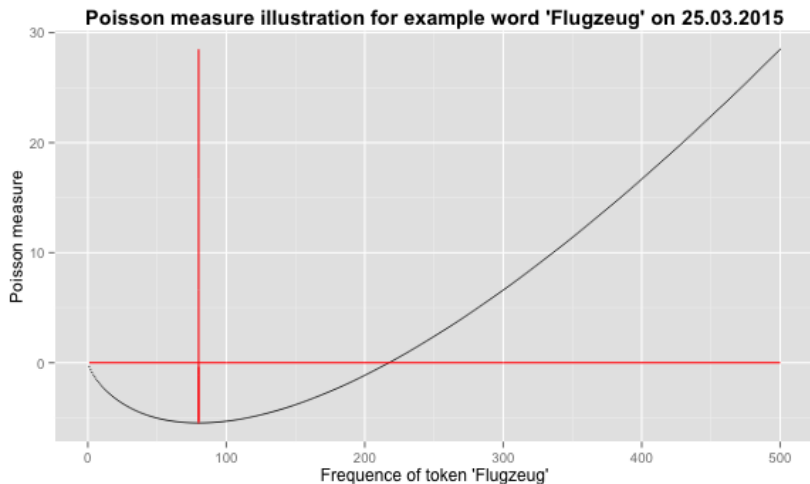
---

- Problem: Berechnung der Fakultät
- Vergleichbarkeit der Werte einzelner Tage untereinander
- Ziel: Hoher Rang soll einen hohen Wert haben ( $-\log$ -Methode)
- Wenn die Frequenz unterdurchschnittlich ist, soll kein hoher Wert erzeugt werden

Formel:

$$\text{sig}_{\text{poisson}}(w) = \frac{k(\log(k) - \log(n \cdot p) - 1))}{\log(n)} \quad (3)$$

# Poisson-Verteilung



*Einschub:*

## Wortzahl vs. Satzzahl zur Berechnung relativer Verhältnisse

---

- Bei der Referenz wird mit Satzzahlen gearbeitet
- Jeder Satz hat im Schnitt gleiche Anzahl von Wörtern ( $\approx 10$ )

$$\frac{Satz_{heute}}{Satz_{jahr}} \approx \frac{Token_{heute}}{Token_{jahr}} \quad (4)$$

- Zur Überprüfung später mehr

# TF/IDF

---

*Idee:* Wir gewichten die Auftretensfrequenz eines Token an einem Tag mit dem Inversen einer Maßzahl, die Angibt an wie vielen Tagen im Referenzjahr das Wort erwähnt wurde.

*Modifikationen:*

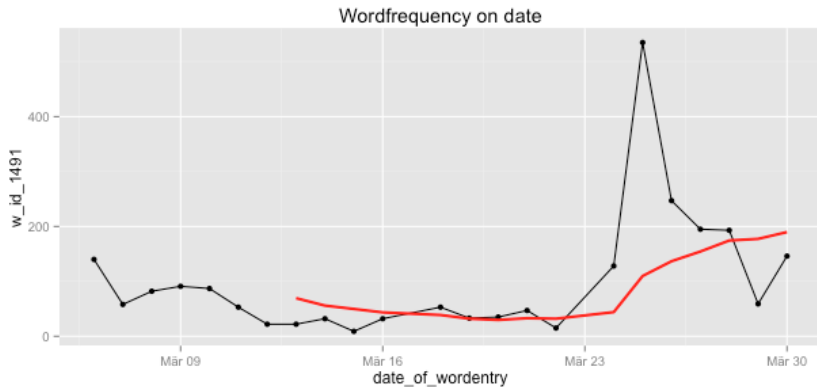
- Relativierung der Frequenz auf Frequenz des häufigsten Tokens am Tag (Vergleichbarkeit)
- Logarithmieren des IDF-Wertes

*Formel:*

$$sig_{tfidf}(w) = \frac{k}{\max(K)} \cdot \log\left(\frac{365}{documentdays(w)}\right) \quad (5)$$

$k$ : Frequenz eines Tokens an einem Tag  $K$ : Alle Frequenzen an einem Tag

# TF-IDF Beispiel





# Z-Score

---

Something about Z-score

# Zeitreihenanalyse

---

## Definition (Zeitreihenanalyse)

Unter einer Zeitreihe versteht man die Entwicklung einer bestimmten Größe, deren Werte im Zeitablauf zu bestimmten Zeitpunkten oder für bestimmte Zeitintervalle erfasst und dargestellt werden

## Maß: gleitender Mittelwert

---

- Glättet Zeit oder Datenreihen
- Erfolgt durch glätten hoher Frequenzanteile
- Es gibt ein Raster der Größe  $n$
- Es werden  $n$  Tage zusammenaddiert und dann durch  $n$  geteilt

Wie hilft uns das weiter?

- Tritt ein Wort häufiger als sein Durchschnittswert an dem Tag auf kann das interessant sein.

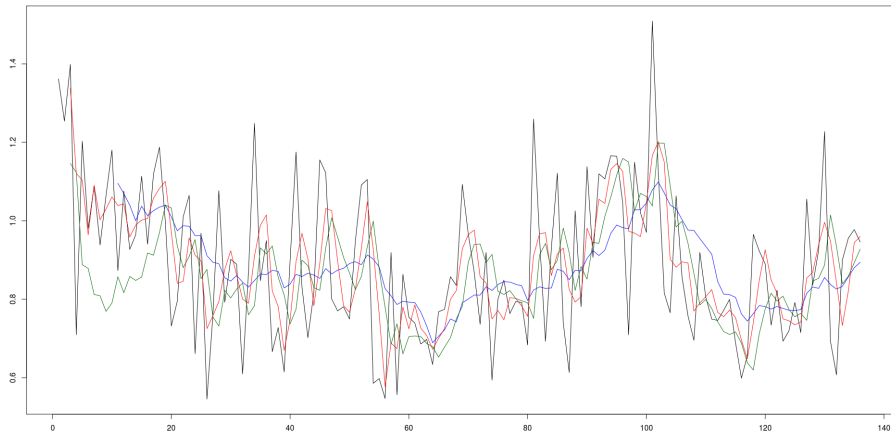
## Erster Ansatz: R

---

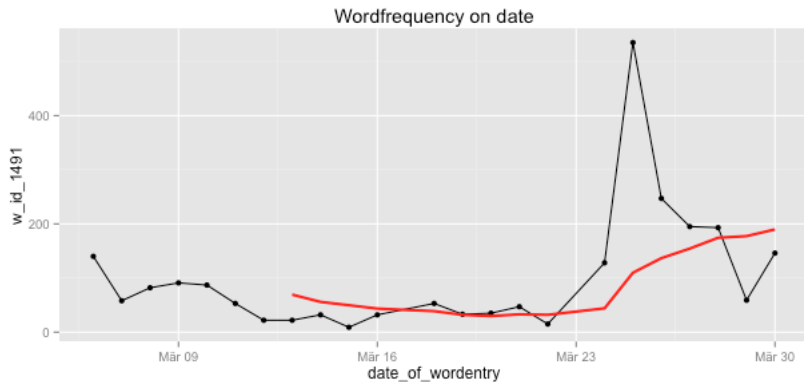
- Der erste Ansatz war ein R Programm welches den gleitenden Mittelwert ausrechnen sollte
- Problem: R verarbeitet Wörter einzeln
- 3 Mio. Wörter  $\rightarrow$  3 Mio. Transaktionen = MySQL Overkill
- Ausführungszeit würde mehrere Tage beanspruchen

# Beispiel: Haus

---



# Beispiel: Flugzeug



## Zweiter Ansatz: MySQL

---

- Der Zweite Ansatz ist es direkt in MySQL zu berechnen
- Problem: Inner Join auf selbe Tabelle (ca. 20 Mio Zeilen)
- Jeder Eintrag muss geprüft werden ob die Join Tabelle den Eintrag in der Größe des Rasters hat
- Eine Datums Differenz Tabelle kann das ganze jedoch beschleunigen

## Finaler Ansatz: R BigTable

---

- Diesmal reshape der Tabelle
- Spalten = Wörter, Zeilen = Datumfelder, Wert = freq
- Darüber kann man das effizient einzeln berechnen
- Danach überführung in alte Struktur und Speicherung



# Vergleich und Auswertung

---

# Qualitative vs. Quantitative Auswertung

---

- Schwierigkeit einer quantifizierbaren qualitativen Evaluierung
- Quantitative vergleiche möglich, aber keine Aussage über Qualität
- Im Rahmen des Projektes möglich:
  - *“Evaluierung durch draufschaun”*
  - Geeignetes Maß zum quantitativen Vergleich nutzen

# Qualitative Auswertung

---

Hier Beispiellisten für den 25.3.2015 einfügen.

# Quantitative Auswertung

---

*Problemstellung:* Vergleich von sortierten Listen mit potentiell unterschiedlichem Inhalt.

- Der Vergleich von Wortpaaren nicht sauber möglich.
- Schwierigkeit eines Mengenbasierter Ansatzes:  
Reihenfolge wird nicht beachtet

## Quantitative Auswertung: Maximum Overlap

---

*Idee:* Es wird ein Mengenbasierter Ansatz für Teillisten genutzt und dann gemittelt.

Für jeden Rang der Listen wird eine Teilliste (Rang 1 bis betrachteter Rang) verglichen.

*Beispiel:* Tafelbild

# Quantitative Auswertung: Maximum Overlap Ergebnisse

---

Tabelle

# Quantitative Auswertung: Maximum Overlap Ergebnisse

---

Graph

## *Einschub II: Wortzahl vs. Satzzahl zur Berechnung relativer Verhältnisse*

---

Graph

- Übereinstimmung von numerisch 100 Prozent



# Zusammenfassung

---

- Es wurden bestehende Verfahren untersucht
- Es wurden weitere Verfahren ausprobiert
- Es wurden die Ergebnisse quantitativ und qualitativ verglichen
- Es wurden MySQL und R Implementierungen umgesetzt.
- Es werden noch Musterbasierte Verfahren zum Cleaning der Listen implementiert
- Es wird noch ein weiteres Vergleichsmaß mit Berücksichtigung der Anzahl der Quellen in denen ein Token erwähnt wird untersucht.

## Quellen (1)

---