

Universität Leipzig  
Fakultät für Mathematik und Informatik  
Institut für Informatik  
Abteilung Automatische Sprachverarbeitung

## Wortschatz Zeitgeist

Seminararbeit

**Autoren:** Döring, Thomas  
Kießling, Max  
Otto, Wolfgang (2885214)

**Modul:** Anwendungen Linguistische Informatik (10-202-2307)

**Abgabe:** 13. Juni 2015. (Sommersemester 2015)

**Betreuer:** Maciej Janicki

**Seminarleiter:** Prof. Dr. Uwe Quasthoff

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Aufgabenstellung . . . . .	1
1.2	Status quo . . . . .	1
1.3	Vergleichbare Ansätze . . . . .	1
<b>2</b>	<b>Das finden von Tagesaktuellen Wörter</b>	<b>2</b>
2.1	Maße zur Trend-Detection . . . . .	2
2.1.1	Relatives Vorkommen (Referenz) . . . . .	2
2.1.2	Poisson-Maß . . . . .	2
2.1.3	Termfrequenz inverse Dokumentenfrequenz (tf-idf) . . . . .	2
2.1.4	Termfrequenz inverse Dokumentenfrequenz inverse Quellen- frequenz (tf-idf-isf) . . . . .	3
2.1.5	Z-Score . . . . .	3
2.1.6	Weitere Maße . . . . .	4
2.2	Zeitreihenanalysen . . . . .	5
2.3	Cleaning . . . . .	5
<b>3</b>	<b>Implementierungen in SQL und R</b>	<b>6</b>
<b>4</b>	<b>Ein empirischer Vergleich</b>	<b>7</b>
4.1	Einleitung . . . . .	7
4.2	Qualitativer Vergleich . . . . .	7
4.3	Quantitativer Vergleich - Average Overlap als Vergleichmaß . . . . .	8
4.3.1	Einführung . . . . .	8
4.3.2	Ergebnisse . . . . .	8
<b>5</b>	<b>Bewertung und Zusammenfassung</b>	<b>10</b>
	<b>Literaturverzeichnis</b>	<b>11</b>

# 1 Einleitung

## 1.1 Aufgabenstellung

Es soll untersucht werden, welche Maße sinnvoll für die Generierung der Wörter des Tages sind. Eine Implementierung soll in SQL und für Zeitreihenuntersuchungen in R erfolgen.

Desweiteren soll ein regelbasiertes Verfahren implementiert werden um Datumsangaben und andere strukturelle Worte zu filtern.

## 1.2 Status quo

## 1.3 Vergleichbare Ansätze

Tagesaktuelle Wikiartikel  
google trends?

## 2 Das finden von Tagesaktuellen Wörter

### 2.1 Maße zur Trend-Detection

#### 2.1.1 Relatives Vorkommen (Referenz)

Das Maß wird durch die Differenz des relativen Anteils eines Tokens an allen Tokens eines Tages und dem relativen Anteil eines Tokens an allen Tokens eines Vergleichskorpus bestimmt.

$$sig_{referenz}(w) = p_{tag} - p_{jahr} \quad (2.1)$$

#### 2.1.2 Poisson-Maß

Die Formel leitet sich aus der Poissonverteilung ab und beschreibt wie Wahrscheinlich es ist, dass die gemessene Tagesfrequenz beobachtet werden kann.

$$sig_{poisson}(w) = \frac{k(\log(k) - \log(n \cdot p) - 1))}{\log(n)} \quad (2.2)$$

k:= Anzahl der Token von w in Tagesbericht

n := Anzahl der Tokens in Tagesbericht

p := relativer Anteil eines Tokens am Jahreskorpus

Es ist das gleiche Maß wie in [8, S. 338-340] beschrieben und hergeleitet. Hier aber nicht zum auffinden von signifikanten Kookurenzen, sondern zum auffinden von signifikanten Nennungen im Tageskorpus gegenüber einem Vergleichskorpus.

#### 2.1.3 Termfrequenz inverse Dokumentenfrequenz (tf-idf)

$$sig_{tfidf}(w) = \frac{k}{\max(K)} \cdot \log\left(\frac{365}{|documentdays(w)|}\right) \quad (2.3)$$

### 2.1.4 Termfrequenz inverse Dokumentenfrequenz inverse Quellenfrequenz (tf-idf-isf)

*Idee:* Wörter sind dann interessant, wenn sie an einem Tag in möglichst vielen verschiedenen Quellen genannt werden.

Als Quelle definieren wir die Serveradresse einer Quelle. Diese wird mittels eines regulären Ausdrucks aus den zugeordneten Quellen in der MySQL-Datenbank ermittelt. Als Gesamtzahl der Quellen verwenden wir alle an einem Tag den Wörtern zugeordnete Quellen.

Das entstandene Signifikanzmaß wird wie folgt definiert:

$$sig_{tfidfisf}(w) = sig_{tfidf}(w) \cdot \log\left(\frac{Q_d}{q_d(w)}\right) \quad (2.4)$$

Analog zur inversen Dokumentenfrequenz wird also das tf-idf-Signifikanzmaß mit dem Logarithmus der inversen relativen Anzahl der Quellenfrequenz multipliziert.  $Q_d$  ist die Anzahl aller erwähnten Quellen an einem Tag  $d$  und  $q_d()$  bildet ein Wort auf die Anzahl der Quellen ab, in denen das Wort an Tag  $d$  erwähnt wird.

### 2.1.5 Z-Score

Benattar et al. beschreiben in [3] einen Ansatz zur Trend-Erkennung basierend auf dem Z-Score. Dabei beziehen Sie neben der relativen Worthäufigkeit noch die Anzahl der Tage ein an denen ein Wort mindestens einmal auftritt, um so das 0-Frequenz-Problem zu umgehen.

#### Berechnung

- Wortfrequenz

$f(w)_d :=$  Anzahl der Vorkommen von Wort  $w$  an Datum  $d$

- relative Worthäufigkeit

Die relative Worthäufigkeit  $p(w)_d$  berechnet sich durch:

$t_d :=$  Anzahl verschiedener Worte an Datum  $d$

$$p(w)_d = \frac{f(w)_d}{t_d}$$

- Erwartungswert

Der Erwartungswert  $\bar{w}$  berechnet sich durch:

$N :=$  Anzahl der Tage in der betrachteten Zeitspanne

$$\bar{w} = \frac{1}{N} \sum p(w)_d$$

- Standardabweichung

Die Standardabweichung  $\sigma_w$  berechnet sich durch:

$$\sigma_w = \sqrt{\frac{1}{N} \sum (p(w)_d - \bar{w})^2}$$

- ZScore

Der Zscore  $Z(w)_d$  misst die Abweichung der relativen Worthäufigkeit vom Erwartungswert in Vielfachen der Standardabweichung.

$$Z(w)_d = \frac{p(w)_d - \bar{w}}{\sigma_w}$$

- Auftrittshäufigkeit

Die Auftrittshäufigkeit  $Po(w)$  gibt an wie vielen Tagen innerhalb des betrachteten Zeitraums das Wort mindestens einmal auftritt:

$nbD(w) :=$  Anzahl der Tage an denen  $w$  vorkommt //  $c_d :=$  Anzahl der Tage innerhalb des betrachteten Zeitraums

$$Po(w) = \frac{nbD(w)}{c_d}$$

- Schwellwerte Zur besseren Unterscheidung echter Trends von statistischen Anomalien schlagen Benattar et. al. vor die Worte anhand ihrer Auftrittshäufigkeit zu clustern. Den Clustern werden dabei Z-Score-Schwellwerte zugeordnet. Überschreitet der Z-Score eines Wortes den Schwellwert seines Clusters wird dieses Wort als signifikant und somit als Trend eingestuft. Cluster mit niedriger Auftrittshäufigkeit erhalten dabei höhere Schwellwerte. Je häufiger ein Wort auftritt desto niedriger wird der Schwellwert.

## Vorgehen

1. Für jedes Wort in  $w$  in  $d$  berechne  $Z(w)$
- 2.

### 2.1.6 Weitere Maße

Einbeziehung der Anzahl von Quellen.

## 2.2 Zeitreihenanalysen

## 2.3 Cleaning

Es sollen Datumsangaben und evtl. neu auftauchende strukturelle Angaben ausgefiltert werden.

Ansatz: Regelbasiert.

Gibt es Maße, die solche Angaben strukturell ausschließen?

## 3 Implementierungen in SQL und R



## 4 Ein empirischer Vergleich

Kriterien: Anteil niederfrequenter Wörter in der Top-Liste

### 4.1 Einleitung

Die Messung der Güte der Ergebnisse stellt eine Herausforderung dar, da es keine geeignete Referenz, beispielsweise in Form eines Goldstandards der wichtigsten Worte eines Tages gibt. Um die Güte trotzdem einschätzen zu können bieten sich zwei herangehensweisen an. Zum einen die eigenständige manuelle Prüfung der Ergebnisse unter selbst formulierten Kriterien, zum anderen der quantitative Vergleich mittels eines geeigneten Abstandsmaßes. Letzterer Ansatz bietet aber nur die Möglichkeit eines Vergleiches der Ähnlichkeiten der Ergebnisse und hilft abzuschätzen wie sich die Ergebnisse gegeneinander verhalten. Über die Güte gibt diese Methode keine Auskunft. Allerdings lassen sich Ausreißer gut erkennen und der Prämisse, dass gleiche Ergebnisse, die aus verschiedenen Mäss ungen stammen eine höhere Wahrscheinlichkeit besitzen gute Ergebnisse zu sein lässt sich auch die Qualität beurteilen.

### 4.2 Qualitativer Vergleich

Um sich einen Eindruck der Ergebnisse anhand der resultierenden sortierten Wortlisten zu verschaffen wurden die Listen ausgewählter Tage verglichen. Da die Untersuchenden keine ausgewiesene Expertise ausweist, die wichtigsten Wörter eines täglichen Nachrichtenstroms zu indentifizieren, die über der eines Zeitungslesers liegt kann die Analyse nicht in die Tiefe gehen. Aber durch die Wahl der Tage lässt sich das Überblicken der Ergebnisse vereinfachen. Deshalb wählten wir den 1.1.2015. Das funktiuniert so noch nicht!!! Umschreiben ist nur blabla!!

## 4.3 Quantitativer Vergleich - Average Overlap als Vergleichmaß

### 4.3.1 Einführung

Der Vergleich zweier mit einer Rangfolge versehenen Listen ist ein bekanntes Problem. In unserem Fall handelt es sich um den Spezialfall von Listen gleicher und fester Länge, aber einer potentiell unendlichen Zahl verschiedener Wörter. Desweiteren sind die Listen nicht *Conjoint*, was bedeutet, dass nicht nur gemeinsame Wörter in den verschiedenen Listen auftauchen. In [14] werden als Einleitung für ein Maß, dass in der Lage ist auch unendliche Listen und Listen verschiedener Länge vergleichen zu können geeignete Verfahren vorgestellt um solche Listen zu vergleichen. Das gewählte Verfahren *Average Overlap* wird von den Autoren als *top-k ranking* identifiziert. Also ein Ranking bis zu einer definierten Tiefe von  $k$ .

Der Vorteil des genutzten Verfahrens für unseren Anwendungsfall ist, dass der Rang der Wörter einen Einfluss auf das Maß haben. Ähnlichkeiten an der Spitze der Liste werden stärker gewichtet.

Das Verfahren ist ein Mengenbasierter Ansatz. Listen sind sich dann ähnlich, wenn sie die relative Anzahl gemeinsamer Wörter hoch ist. Um nun aufsteigende Gewichtungen zu erhalten wird nun nicht nur die gesamte Überlappung zweier Listen gemessen, sondern die Listen in  $K$  Listen unterteilt, wobei  $K$  die Länge der Listen ist und jede einzelne Liste jeweils alle Elemente bis zu dem Rang des Laufindex  $k$  von 1 bis  $K$  enthält. Also eine Liste der Form:  $[[ \text{Wort 1}], [\text{Wort 1}, \text{Wort 2}], \dots]$ . Nun wird bei den einzelnen Listen gleicher Länge die relative Überlappung gemessen. Um nun das Vergleichsmaß zu erhalten wird der Durchschnitt aller errechneten Werte gemessen. Formalisiert ergibt dies:

$$AO(S, T) = \frac{\sum_{k=1}^K \frac{|M(S_k) \cap M(T_k)|}{k}}{K} \quad (4.1)$$

Wobei  $S$  und  $T$  zwei Listen sind, der tiefgestellte Index  $k$  die Teilliste bis zum Rang  $k$  angibt und  $K$  die Länge der beiden Listen definiert.  $M$  ist hierbei die Abbildung einer Liste auf die Menge der enthaltenen Elemente.

### 4.3.2 Ergebnisse

Hier die Ergebnisse für den 1.5.2015 mit der Listenlänge  $K = 1000$

	List	List_to_compare	average_overlap
1	tf_idf	poisson	0.66
2	tf_idf	z-score	0.18
3	tf_idf	freqratio	0.31
4	tf_idf	freqratio_old	0.31
5	tf_idf	poisson_old	0.66
6	poisson	z-score	0.15
7	poisson	freqratio	0.16
8	poisson	freqratio_old	0.16
9	poisson	poisson_old	1.00
10	z-score	freqratio	0.16
11	z-score	freqratio_old	0.16
12	z-score	poisson_old	0.15
13	freqratio	freqratio_old	1.00
14	freqratio	poisson_old	0.16
15	freqratio_old	poisson_old	0.16

Tabelle 4.1: Avarage Overlap Comparison

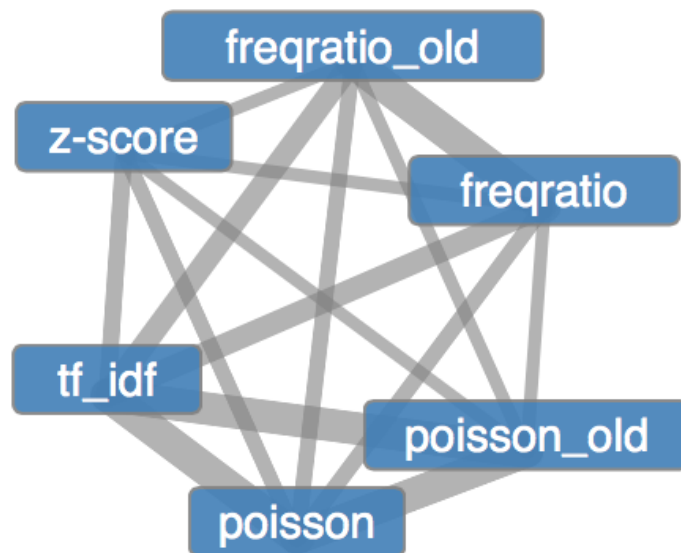


Abbildung 4.1: Graph of Average Overlap

## 5 Bewertung und Zusammenfassung

# Literaturverzeichnis

- [1] AGGARWAL, Charu C.: Mining text streams. In: *Mining Text Data*. Springer, 2012, S. 297–321
- [2] AGGARWAL, Charu C.: Mining text and social streams: a review. In: *ACM SIGKDD Explorations Newsletter* 15 (2014), Nr. 2, S. 9–19
- [3] BENATTAR, Gary ; TRÉBUCHET, Philippe u. a.: Trend Analysis in Polls, Topics, Opinions and Answers. (2011)
- [4] BENHARDUS, James ; KALITA, Jugal: Streaming trend detection in twitter. In: *International Journal of Web Based Communities* 9 (2013), Nr. 1, S. 122–139
- [5] CHEN, Chaomei: CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. In: *Journal of the American Society for information Science and Technology* 57 (2006), Nr. 3, S. 359–377
- [6] GAO, Yan ; LIU, Jin ; MA, PeiXun: The hot keyphrase extraction based on tf\* pdf. In: *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on IEEE*, 2011, S. 1524–1528
- [7] GUPTA, Manish ; GAO, Jing ; AGGARWAL, Charu ; HAN, Jiawei: Outlier detection for temporal data. In: *Synthesis Lectures on Data Mining and Knowledge Discovery* 5 (2014), Nr. 1, S. 1–129
- [8] HEYER, Gerhard ; QUASTHOFF, Uwe ; WITTIG, Thomas: *Text Mining: Wissensrohstoff Text. Konzepte, Algorithmen, Ergebnisse*. W3L, 2006
- [9] JAHNAVI, Y ; RADHIKA, Y: Hot topic extraction based on frequency, position, scattering and topical weight for time sliced news documents. In: *Advanced Computing Technologies (ICACT), 2013 15th International Conference on IEEE*, 2013, S. 1–6
- [10] KONTOSTATHIS, April ; GALITSKY, Leon M. ; POTTENGER, William M. ; ROY, Soma ; PHELPS, Daniel J.: A survey of emerging trend detection in textual data mining. In: *Survey of Text Mining*. Springer, 2004, S. 185–224
- [11] LONG, William J.: *Real-Time Trend Detection Using Segmental Linear Regression*.  
<http://groups.csail.mit.edu/medg/projects/maita/documents/trends/segmental-trend>
- [12] NIKOLOV, Stannislav: *Early detection of Twitter trends explained*.  
<https://snikolov.wordpress.com/2012/11/14/early-detection-of-twitter-trends/>.  
Version: November 2012

- [13] VIERMETZ, Maximilian ; SKUBACZ, Michal ; ZIEGLER, Cai-Nicolas ; SEIPEL, Dietmar: Tracking topic evolution in news environments. In: *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on IEEE*, 2008, S. 215–220
- [14] WEBBER, William ; MOFFAT, Alistair ; ZOBEL, Justin: A similarity measure for indefinite rankings. In: *ACM Transactions on Information Systems (TOIS)* 28 (2010), Nr. 4, S. 20