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: 27. Oktober 2015. (Sommersemester 2015)
Betreuer: Maciej Janicki
Seminarleiter: Prof. Dr. Uwe Quasthoff

Inhaltsverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Das Portal Ẅörter des Tages (wortschatz.uni-leipzig.de/wort-des-tages) stellt eine Übersicht von Wörtern, die an einem ausgewählten Tag besonders relevant erschienen dar. Die Wörter sind in neun Kategorien eingeordnet. Nach der Beschreibung auf der Website werden die Wörter ermittelt in dem die tagesaktuelle Häufigkeit eines Wortes mit seiner durchschnittlichen Häufigkeit über längere Zeit hinweg gemessen wird.
Die Aufgabe dieser Arbeit ist es verschiedene Möglichkeiten der Bestimmung von Wörtern, die an einem gewählten Tag von besonderer Relevanz sind zu beschreiben, zu vergleichen und zu evaluieren. Die Datengrundlage zur erstellung der Wörter des Tages ist ein Corpus, das durch tägliches crawlen von Newsseiten generiert wird. Die Quellen der Newseiten sind eine definierte Menge an für relevant erachtete Seiten mit Nachrichten wie zum Beispiel Spiegel.de.
Bei allen Ansätzen, die auf das Vorkommen in einem Referenzzeitraum rekurrieren besteht das Korpus aus allen gecrawlten Newsseiten des voragngegangenen Jahres (2014). Als Zusatzaufgabe soll ein musterbasiertes Verfahren in SQL entworfen werden, das es ermöglicht aufgrund eines gewählten Verfahrens falsch identifizierte Wörter zu filtern. Ein Beispiel hierfür sind Datumsangaben, die als relevant erscheinen, da sie Tagesaktuell oft auftauchen, aber im Vergleichszeitraum selten.

1.2 Status quo

1.3 Vergleichbare Ansätze

Tagesaktuelle Wikiartikel
google trends?

2 Methoden zum Finden tagesaktueller Wörter

Im folgenden Abschnitt werden vier Methoden vorgestellt, die füer jedes Wort eines Tageskorpus eine Maßzahl bestimmen, die die Relevanz des Wortes an diesem Tag ausdrücken soll.

2.1 Maße zur Trend-Detection

2.1.1 Relatives Vorkommen (Referenz)

Ein einfacher Ansatz, der als Referenz zum Finden relevanter Wörter eines Tages dient ist der, das Auftreten jedes Tokens im Tageskorpus mit dem Auftreten im Referenzkorpus ins Verhältnis zu setzen.
Hierbei werden um eine Vergleichbarkeit zwischen verschiednen Tagen zu gewährleisten die Frequenzen der Wörter über die Anzahl aller Tokens im Tages bzw. Referenzkorpus normiert.

Formel:

$$sig_{freqratio}(w) = \frac{\frac{k_{day}}{n_{day}}}{\frac{k_{2014}}{n_{2014}}} \tag{2.1}$$

k_{day}: Frequenz des Tokens an einem Tag
n_{day}: Summe der Frequenzen aller Tokens eines Tages
k_{2014}: Frequenz des Tokens im Referenz Zeitrahmen (2014)
n_{day}: Summe der Frequenzen aller Tokens im Referenzzeitraum (2014)

In Abbildung 2.1 stellt die schwarze Gerade dar, wie sich der Wert der relativen Frequenz verhält, wenn die Anzahl des Auftretens eines Tokens variiert. Die senkrechte roten Linien markiert die Anzahl der Tokens, bei denen das relative Auftreten dem relativen Auftreten im Referenzkorpus entspricht. Der Wert der relativen Frequenz steigt also linear bei der Steigerung der Anzahl der Tokens eines Wortes.Dies führt zu der Problematik der Überschätzung von niederfrequenten Wörter im Referenzkorpus selbst bei relativ seltenem Auftreten im Tageskorpus. Bei niederfrequenten Wörtern ist der Anstieg der Gerade sehr viel steiler.
Um diesem Problem gerecht zu werden hilft es ein Maßfür die Relevanz eines Wortes finden, welches ein geringe Überschreitung des relativen Anteils im Referenzkorpus

weniger goutiert als eine höhere. Der Ansatz des Poisson Maßes (2.1.2) versucht dem gerecht zu werden.

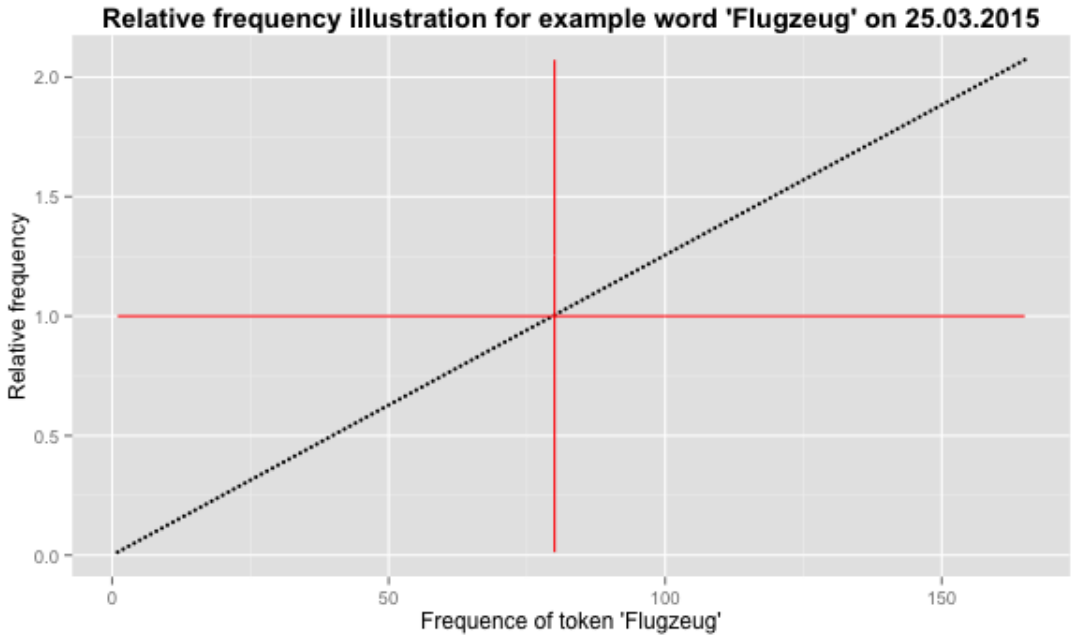

Abbildung 2.1: Illustration der relativen Frequenz des Tokens "Flugzeug" am 25.03.2015

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.

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

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)

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

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

Idee: Wörter sind dann interessant, wenn sie an einem Tag in m;glichst vielen verschiedenen Quellen gennant 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 Signifikanzmaßwird wie folgt definiert:

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

Analog zur inversen Dokumentenfrequenz wird also das tf-idf-Signifikanzmaßmit 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

Das bei diesem, von Benattar et al. [3] beschriebenen, Ansatz zu Grunde liegende statistische Mittel ist der Z-Score. Dieser misst die Abweichung einer Zufallsvariable vom Erwartungswert in Vielfachen der Standartabweichung. Als Zufallsvariable dient die relative Worthäufigkeit am jeweiligen Tag. Erwartungswert und Standartabweichung werden mittels des Referenzkorpus berechnet.

- Wortfrequenz

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

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

$$t_d := \text{Anzahl verschiedener Worte an Datum } d$$
$$p(w)_d = \frac{f(w)_d}{t_d}$$

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

$$N := \text{Anzahl der Tage im Referenzkorpus}$$
$$\bar{w} = \frac{1}{N} \sum p(w)_d$$

- Standartabweichung
  Die Standartabweichung σ_w berechnet sich durch:

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

- Z-Score
  Der Z-Score $Z(w)_d$ misst die Abweichung der relativen Worthäufigkeit vom Erwartungswert in Vielfachen der Standartabweichung.

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

Ein Problem bei der Verwendung des Z-Scores sind Worte mit sehr kleinem Erwartungswert. Dies sind häufig Worte, welche an nur sehr wenigen Tagen und im schlimmsten Fall gar nicht im Referenzkorpus auftreten. Bei diesem Worten bedeutet bereits eine sehr geringe Wortfrequenz von ein oder zwei Vorkommen einen enormen Ausschlag im Z-Score. Um dieses sogenannte Zero-Frequency-Problem abzuschwächen 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.

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

$$\begin{aligned} nbD(w) &:= \text{Anzahl der Tage an denen w vorkommt} \\ c_d &:= \text{Anzahl der Tage innerhalb des betrachteten Zeitraums} \\ Po(w) &= \frac{nbD(w)}{c_d} \end{aligned}$$

- Schwellwerte
  Die Tabelle zeigt den Z-Score-Schwellwert gecluster nach Auftrittshäufigkeit $Po(w)$:

[0-5[	[5-10[	[10-20[	[20-30[	[30-50[	[50-60[	[60-70[	[70-80[	[80-100]
20	25	15	12	10	9	8	6	5

2.1.6 Weitere Maße

Einbeziehung der Anzahl von Quelle

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, beispielswiese in Form eines Goldstandards der wichtigsten Worte eines Tages gibt. Um die Gúte 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 Verlgeiches 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 soriterten Wortlisten zu verschaffen wurden die Listen ausgewählter Tage verglichen. Da die Untersuchenden keine ausgewiesene Expertiese 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 funktiuoniert 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 [**?**] 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 Maßhaben. 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 Laufindexes k von 1 bis K enthält. Also eine Liste der Form: [[Wort 1],[Wort 1, Wort 2], ...]. Nun wird bei den einzelnen Listen gleicher Länge die relative Überlappung gemessen. Um nun das Verlgleichsmaßzu erhalten wird der Durchschnitt aller errechneten Werte gemessen. Formalisiert ergibt dies:

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

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

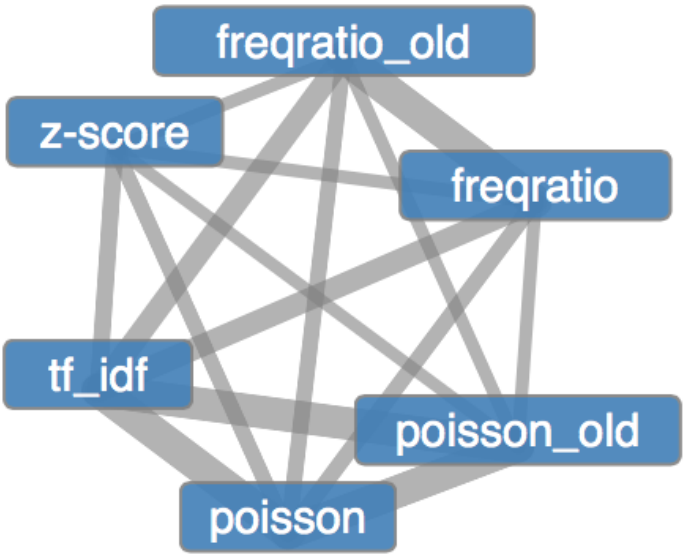

Abbildung 4.1: Graph of Average Overlap

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] 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