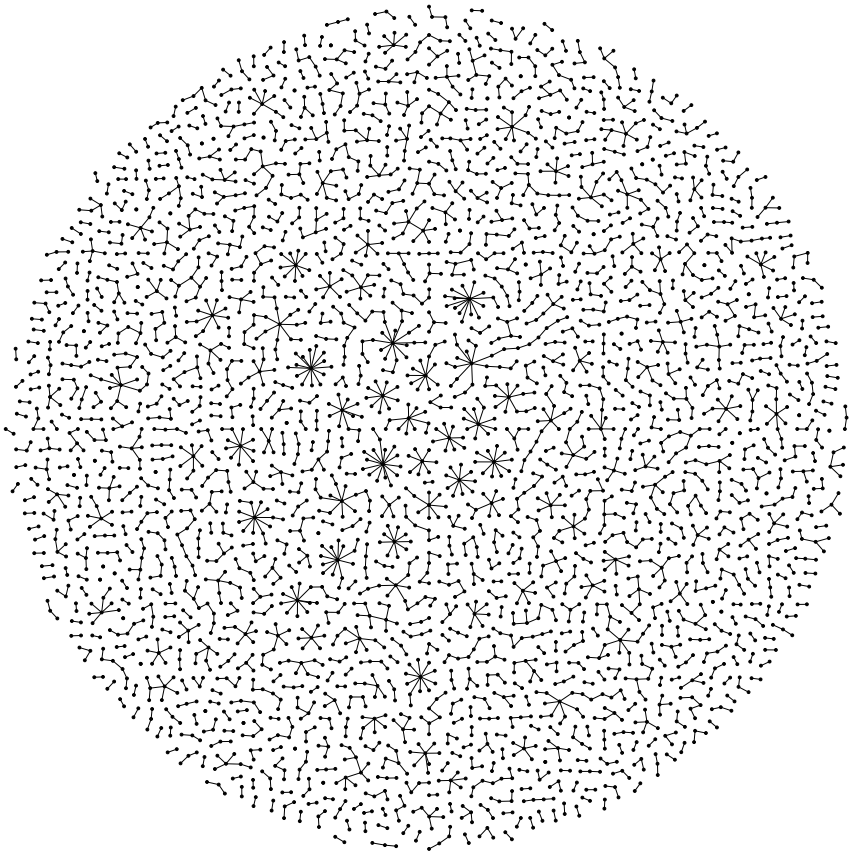


INDEPENDENT COURSEWORK 2

VISUALISIERUNG VON WEBFORUMSDATEN

HTW BERLIN / SEPTEMBER 22, 2017

Inhaltsverzeichnis

Öffentliche Diskussion im Netz

Internetkommentare werden mit zunehmender Bedeutung nicht nur Ort von Debatten, sondern auch öfter Diskursobjekt. Die Verbreitung von Textbeiträgen über das Internet in Form von Postings, Kommentaren und Kurznachrichten bildet heute den Grundpfeiler aller **"Social Media"** Plattformen. Plattformen wie Facebook, Twitter und Reddit stellen Möglichkeiten für öffentliche Diskurse bereit. Diese Plattformen haben die Grenzen zwischen Sender und Empfänger aufgeweicht. Aber eine textbasierte Kommunikation beeinflusst auch wie wir Kommunikation verstehen, "the media is the message". Eine Kommunikationsform, die vielen Sprechern eine Stimme gibt, ist demokratisch, aber nicht ohne Probleme. Unternehmen, Gruppierungen und Einzelpersonen ringen um Aufmerksamkeit für ein größeres Sprachrohr.

Ziel der Arbeit

In dieser Arbeit soll es um Social Media Diskussionen gehen. Aktuell hat diese Kommunikationsart einen Umfang erreicht, der nicht mehr leicht verständlich ist. Die Arbeit versucht die Untergruppe der **öffentlichen, asynchronen und textbasierten Kommunikation zu visualisieren**. (Also zum Beispiel Foren, Kommentare auf Nachrichtenseiten oder Blogs). Die Technologie dieser Kommunikation wird im ersten Teil untersucht. Eine Kategorisierung unterschiedlicher Kommunikationmodi erfolgt im zweiten Teil. Für praktischen Teil der Arbeit wurde eine Visualisierung entwickelt basierend auf einer Kategorisierung (Thread-Diskussion) aus dem zweiten Teil.

Entwicklung der Diskussionsplattformen

Internetnutzer in den 80er und 90er Jahren hatten meist das nötige "know-how" um Programme wie Bulletin Boards, Usenet und IRC zu nutzen und teilweise auch zu hosten. Mit dem Aufkommen von HTML wurde das Internet immer einfacher zugänglich. Software wie phpBB[17] erlaubt es jedem mit einem Browser an einer Diskussion teilzunehmen.

Heute wird das Internet von Social Media Plattformen dominiert. In den USA sind 67% der Erwachsenen Facebook-Nutzer und 44% nutzen die Seite auch um sich über Nachrichten zu informeren[GS16]. Zusammen mit anderen Plattformen (Twitter, reddit, Tumblr) sind es für Millionen Menschen möglich an öffentlichen Debatten teilzunehmen.

Nachrichten im Sinne von *news*, nicht *messages*

Motivation

Warum wird dieses Feature überhaupt genutzt? Was sind die Gründe im auf Nachrichtenseiten zu kommentieren? Eine Umfrage über die Motivation von Nutzer, die auf Newsseiten kommentieren[Str+] zeigt, dass Nutzer sehr unterschiedliche Motivationen haben:

- Ausdruck von Emotionen und Meinungen 44%
- Korrigieren von Fehlern 42%
- Teilnahme an der Debatte 39%
- Information bereitstellen 38%
- Ausgleichen der Debatte 35%

Es bleibt noch die Frage, welche Form diese Kommunikation hat.

Diskursformen

Grundelement einer textbasierten Kommunikation ist immer ein Text. Alleine betrachtet ist der Text nur eine einfache Zeichenkette. Eine Kommunikationsform entsteht erst in Verbindung mit anderen Elementen. Die Abbildung 1 soll das verdeutlichen:

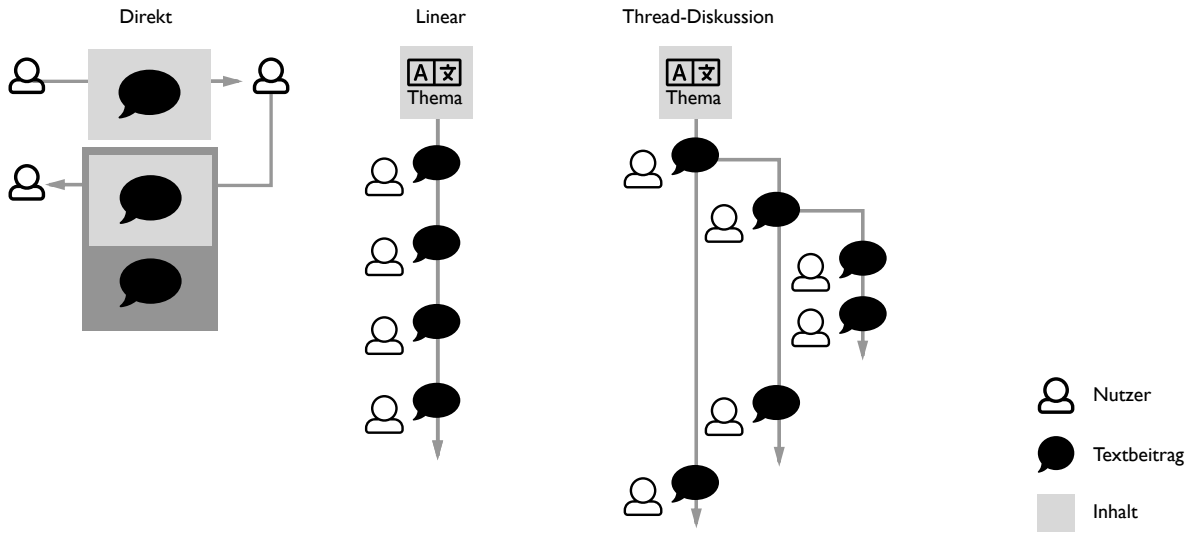


Abbildung 1: Schematische Darstellung der Diskursformen

Direkt

In der direkten Kommunikation senden Nutzer 👤 Nachrichten 💬 direkt. Der Kontext einer Nachricht besteht dann zwischen 👤 ↦ 💬 ↦ 👤. Chat und Email sind Beispiele für diese Form.

Linear

Auf Online-News-Webseiten, wie zum Beispiel `spiegel.de`, können Nutzer Kommentare zu Artikeln schreiben (Relation 🈂 ↦ [💬 1, 💬 2, 💬 3 . . .]). Die Kommentare können dann chronologisch sortiert werden.

Bei vielen Nutzer wird in dieser Form ein Problem deutlich: Ein Text hat immer auch eine semantische Komponente (z.B.

unterschiedliche Aspekte eines Themas). Gibt es ein Thema A und eine Thema B, dann müssen Nutzer dies im Text schreiben. 🈂 ↦ [💬$_1$, 💬$_2$, 💬$_3$, . . .]

Bulletin Boards versuchen dieses Problem zu lösen in dem Nutzer andere Nachrichten als Zitat in neue Nachrichten einbauen können.

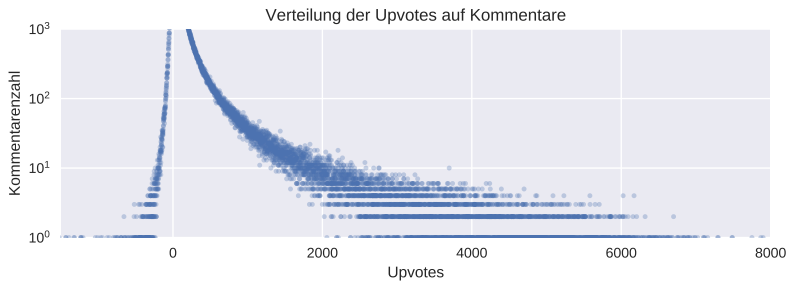
Thread-Diskussion

Ein Thread-basiertes Komentarsystem erlaubt es Nutzern nicht nur Inhalte zu kommentiern (🈂 ↦ 💬$_i$) ,sondern auch ein anderen Kommentar 💬 ↦ [💬$_1$, 💬$_2$, 💬$_3$ . . .]. Dadurch entsteht eine Baumstruktur. Wie diese Struktur in der Diskussion genutzt wird, hängt von den Konventionen der Plattform ab. Diese werden auch stark durch das Nutzerinterface einer Website geprägt. Generell kann in Thread-Diskussion auf zwei Arten geantwortet werden. Sequentiell 💬 ↦ [💬, 💬] oder direkt 💬 ↦ [💬 ↦ [💬]]

Sichtbar und Interaktionen

Thread-Diskussionen haben aber auch Probleme. Eine sehr große Menge an Subkommentaren kann Kommentare, die direkt auf das Thema antworten, außerhalb des Sichtbereichs drängen (Abbildung 2).

Kommentare die zuerst gesehen werden erhalten mehr Aufmerksamkeit. Dies kann wiederum die Sortierung der Kommentare beeinflussen (siehe Abschnitt reddit.com).



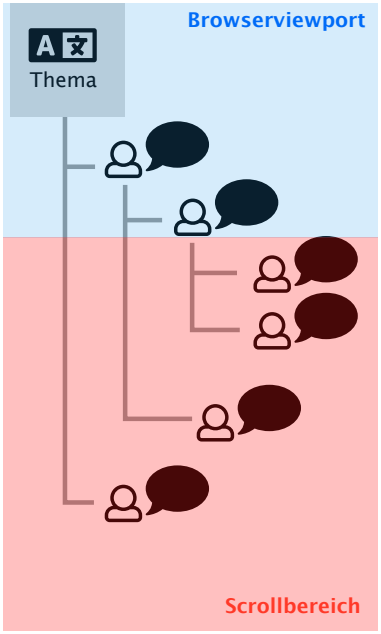


Abbildung 3: Je mehr Unterkommentare das erste Kommentar hat, desto weniger sichtbar sind andere Root-Kommentare

Visualisierung

Die Umsetzung einer Visualisierung umfasst nicht nur die Bereiche Statistik und Grafikdesign, inzwischen spielt Informatik eine Hauptrolle im Informationsdesign. Fry [Fry04] bezeichnet diese Mischung von Fachbereichen als **"Computional Information Design"**. Er definiert auch einen Prozess (Abbildung 4) um die Vorgehensweise im Informationsdesign zu verdeutlichen. Die Beschreibung der Umsetzung dieser Arbeit orientiert sich an diesen Schritten.

Abbildung 4: Arbeitschritte [Fry04, S. 164]

Datenbeschaffung (acquire)

Es sind viele aufbereitete Datensets verfügbar die Postings von Usern enthalten und es gibt noch mehr potentielle Datenquellen. Social Media Plattformen stellen Programmierschnittstellen (APIs) bereit auf denen Daten direkt, meist im JSON-Format, abgerufen werden können. Jedoch sind diese für die Entwicklung von Endanwendungen gedacht. Das Herunterladen größer Mengen an Beiträgen ist unerwünscht, limitiert und unterliegt rechtlichen Auflagen. Meistens ist nur eine nichtkommerziele Nutzung erlaubt.

Im Abschnitt Sichtbar und Interaktionen wurden die Probleme genannt, die bei einer Diskussion mit sehr vielen beteiligten und vielen Kommentaren auftreten. Das Datenset für die Visualisierung sollte deshalb auch sehr umfangreiche Diskussion enthalten. Beiträge sollten auch im Thread-Format vorliegen (siehe Thread-Diskussion).

Ein Datenset, dass diesen Anforderungen entspricht, ist das von Baumgartner erstellte *reddit*-Datenset [Bau17].

reddit.com

Die Webseite *reddit.com* wird von 6% der U.S. Amerikanischen Internetnutzern genutzt[DS13] . Reddit ist ein "link aggregator" (dt.

Linksammler). Diese Art von Social Media Plattformen bietet Nutzern zwei Hauptfunktionen: Nutzer können URLs “posten” und diese bewerten. Jede URL kann kommentiert werden. Zusätzlich ist es möglich auf Posts und Kommentare positiv oder negativ zu bewerten. Diese Interaktion wird Voting genannt und ermöglicht Kommentare nach einer zusätzlichen Dimension zu ordnen. In den Verhaltensregeln für reddit (die “reddiquette” [red]) wird *voting* wie folgt beschrieben:

> **Vote.** If you think something contributes to conversation, upvote it. If you think it does not contribute to the subreddit it is posted in or is off-topic in a particular community, downvote it.

Standardmäßig wird die Beziehung 💬 $\mapsto$ [💬$_1$, 💬$_2$, 💬$_3$ . . .] nicht chronologisch sortiert, sondern nach einem Maß, dass aus den “up” und “down” Wahlen erechnet wird [Mil09].

Reddit bietet eine offizielle Programmierschnittstelle (**API**) an, aber diese unterliegt bestimmten **Begrenzungen** ähnlich denen anderer Plattform. Die exakte Menge an up/down Wahlen ist geheim und es wird nur ein Wert (“ups”) ausgeben. Die API ist auf ein bestimmtes Featureset begrenzt. Eine Suche nach Einreichungen mit einer hohen Kommentaranzahl ist nicht möglich.

Deswegen verwende ich ein “offline” Datensatz der Reddit-Kommentare[Bau17]. Der Datensatz beinhaltet alle Reddit-Kommentare und ist unterteilt Monate. Abbildung 5 zeigt, dass der Datensatz aktuell mehr als 3500 Millionen Kommentare enthält.

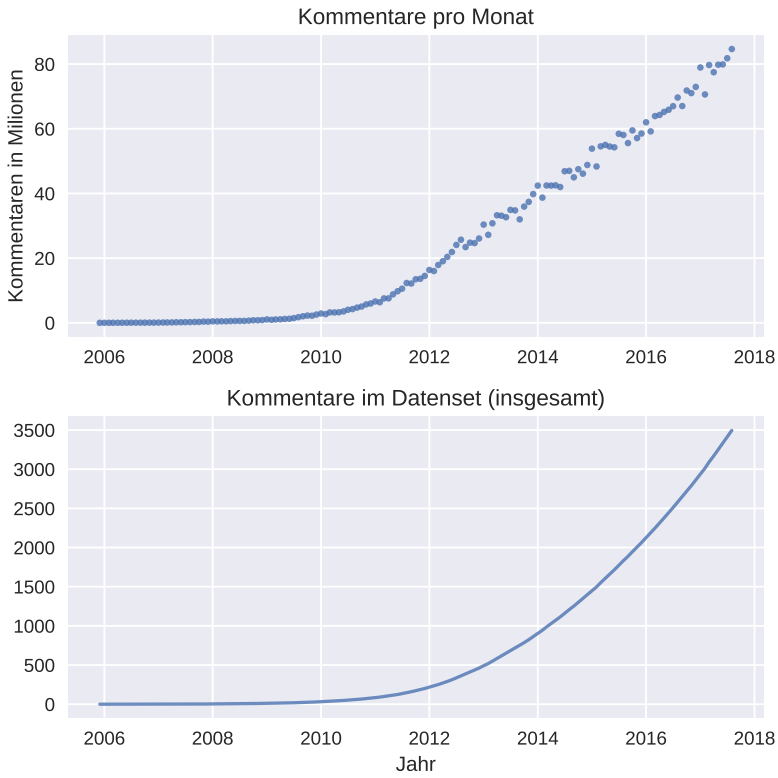


Abbildung 5: Kommentaranzahl in [Bau17]

Datenformat (parse)

Der Datensatz enthält Kommentare im JSON-Format, die jeweils durch Zeilenumbrüche getrennt sind. Ein Beispiel-Datensatz ist hier aufgelistet:

```
{'author': 'RoboChrist',
'author_flair_css_class': None,
'author_flair_text': None,
'body': 'There\'s "s ...',
'controversiality': 0,
'created_utc': 1452097977,
'distinguished': None,
'edited': False,
'gilded': 0,
'id': 'cyo5nvp',
'link_id': 't3_3zpp3w',
'parent_id': 't1_cyo4in3',
'retrieved_on': 1454317397,
'score': 158,
'stickied': False,
'subreddit': 'dataisbeautiful',
'subreddit_id': 't5_2tk95',
'ups': 158}
```

Aufgrund der hohen Speicherplatzanforderung habe ich mich zuerst nur auf Daten aus dem Jahr 2016 beschränkt. Die Kommentare für 2016 sind allein 80 GB groß. Für die Verarbeitung der Dateien habe ich das Pythonframework Dask [Roc17] verwendet. Die Dask-API erstellt programmatisch ein Ausführungsbaum ("compute graph"), der dann parallel auf den Zeilen des Datensatzes ausgeführt werden kann. Mit dem folgenden Code-Beispiel wurde die durschnittliche Länge der Kommentare gemmessen:

```
jsons  = lines.map(json.loads)
length = jsons.map(lambda data: len(data['body']))
freq   = length.frequencies()
result = client.compute(freq).gather(future)
```

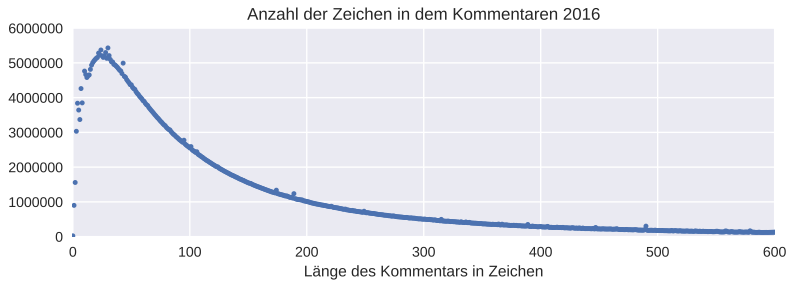


Abbildung 6: Ein reddit-Kommentar hat eine maximale Länge von 10000 Zeichen. Für entfernte und gelöschte Kommentare werden über die Zeichenketten `[removed]` und `[deleted]` dargestellt.

Dask ist ideal für die Stapelverarbeitung großer Datenmengen, aber für eine Visualisierung von Thread-Diskussionen ist nicht immer

nötig den gesamten Datensatz zu filtern.

Datenbank

Die Datenbank von *reddit* selbst speichert alle Daten auch als key/value-Paare ab [Edb13]. Reddit-Entwickler haben es dadurch leichter neue Features zu implementieren. Normalerweise müsste das Datenbankschema angepasst werden und das ist nicht trivial. Deswegen habe ich die gleiche Schemaart für die Datenbank der Visualisierung verwendet. Als Datenbank verwende ich die relationale Datenbank Postgres. Die Tabelle der Kommentare hat nur zwei Reihen: `id` und `data`. Trotzdem kann man mit dem Datentyp `jsonb` direkt auf die JSON-Elemente zugreifen.

Die folgende Suchanfrage findet Kommentare die einem Thread angehören und braucht auf dem Datensatz von 62 Millionen Kommentaren 1:22 Minuten.

```
select data from comments
where
  data->>'link_id' = 't3_42n6x9'
```

Für Anfrage wie diese wird deswegen ein Index auf das Feld `link_id` erstellt. Die gleiche Anfrage braucht mit Index nur noch 2.8s.

filter

Jeder Kommentar in der Datenbank hat ein Feld `parent_id`, mit der die Relation zum Elternknoten hergestellt wird. So einsteht am Ende eine Baumstruktur, deren Wurzel immer eine reddit-“submissions” steht.

reddit-“submissions” sind Elemente aus einer Überschrift+URL oder einer Überschrift+Text

Es ist aber auch möglich die Relation 💬 ↦ 💬 als Relation zwischen Usern zu verstehen 👤 (💬) ↦ 👤 (💬)

Abbildung 7 zeigt einen Graphen, der aus den diesen Relationen gebiltet wurde. Ich habe den Graphen nur Testweise erstellt, um zu sehen, ob sich die Visualisierung lohnt. Jedoch haben die getesteten Graphlayout-Algorithmen kein zufriedenstellendes Ergebnis geliefert.

mine

Populäre Reddit-Diskussionen erhalten bis zu 30000 Kommentare oder mehr, aber nur wenige haben ereichen eine solches Ausmaß, wie die Abbildung 8 zeigt.

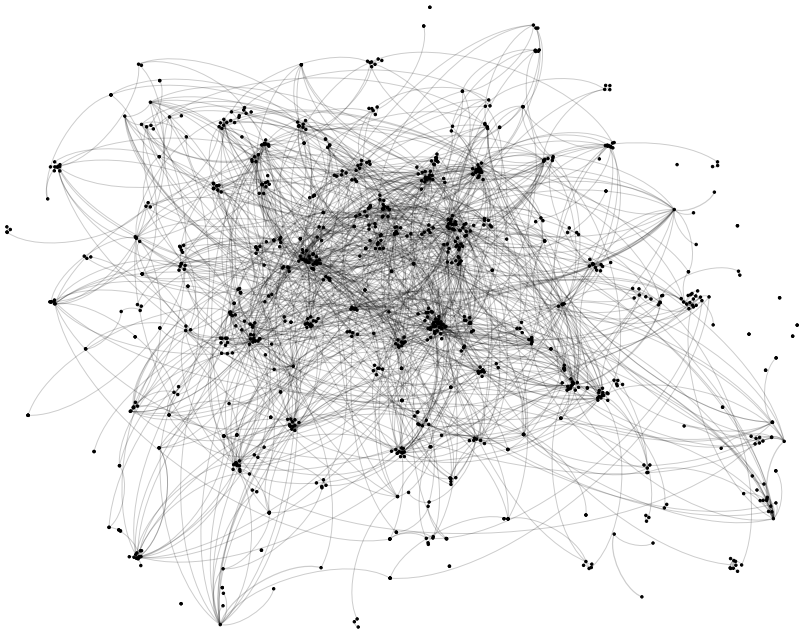


Abbildung 7: Graph der Interaktionen in einem Thread: Jeder Knoten stellt einen Author dar. Authoren die auf ein Kommentar geantworted haben sind durch eine Kante verbunden.

Repräsentationen

Die Darstellung der Kommentartexte alleine übersteigt schnell die Größe eines HD-Bildschirms: Bei durchschnittlich 177 Zeichen pro Kommentar und z.B. 4000 Kommentaren müsste man 708000 Zeichen darstellen. Die Menge an Zeichen pro Textzeile ist auf etwa 80 begrenzt, da Texte sonst nur schwer lesbar sind. Mit einer Zeilenhöhe von etwa 22 Pixel sind allein für den Text fast 200000 Pixel nötig.

Um trotzdem den Kontext eines Kommentares zu visualisieren habe ich ein Overview+Detail Schema verwendet. Ein overview+detail Interface stellt Kontext und Detail gleichzeit, aber räumlich getrennt dar [CKB09]. Das Detail ist der Text in lesbarer Auflösung. Der Kontext ist die Datenstruktur des Kommentartthreads ohne Text.

Diese Datenstruktur ist ein Baum, aber die klassische Darstellung von Bäumen hat kein gutes “data/ink”-Verhältniss [Tuf09]. Zum anderen, ist die Darstellung optisch gesehen nicht sehr simpel (Die rote Linie in Abbildung (a) soll die “unruhige” optische Linie zeigen). Der “icicle plot” ist für die Darstellung von Clusterungen gedacht, deswegen ist die Höhe eines Knotens durch die Summe der Höhen seiner Kinder bestimmt.

Für die Darstellung von Kommentaren zeichne ich aber alle Knoten mit gleicher Höhe

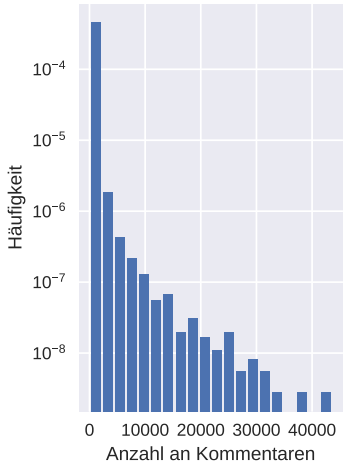
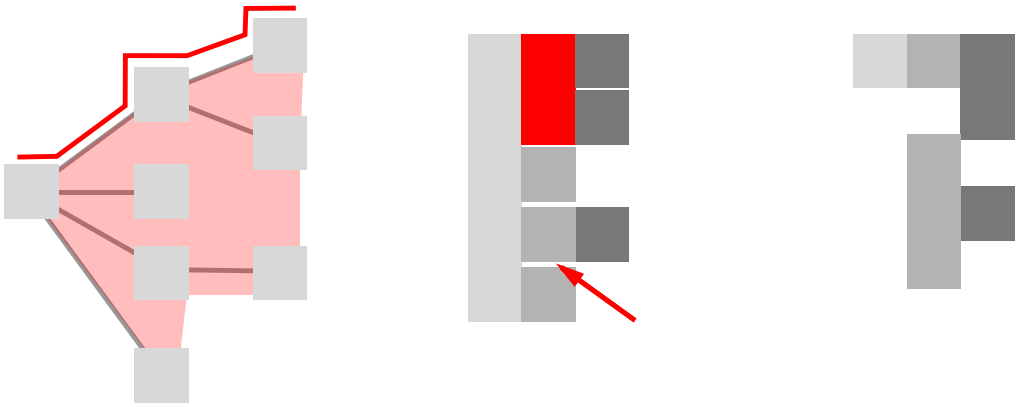


Abbildung 8: Die Menge der Kommentare ist logarithmisch auf die Einreichungen verteilt

Abbildung 9: fokus + overview = Viewport



(a) Klassisch

(b) icicle plot, der Knoten überspannt die Unterknoten

(c) “Icicle” Baum, uniforme Knoten

Abbildung 10: Baumlayoutmethoden

refine

Die Höhe des *icicle plots* wird an die Höhe des Browser-Viewports angepasst. Bei einer sehr hohen Anzahl an Knoten kann es vorkommen, dass die Höhe eines Knotens kleiner wird als 1px. Die Visualisierung wird dann durch das *anti aliasing* des Browsers heller als gewollt (siehe Abbildung a).

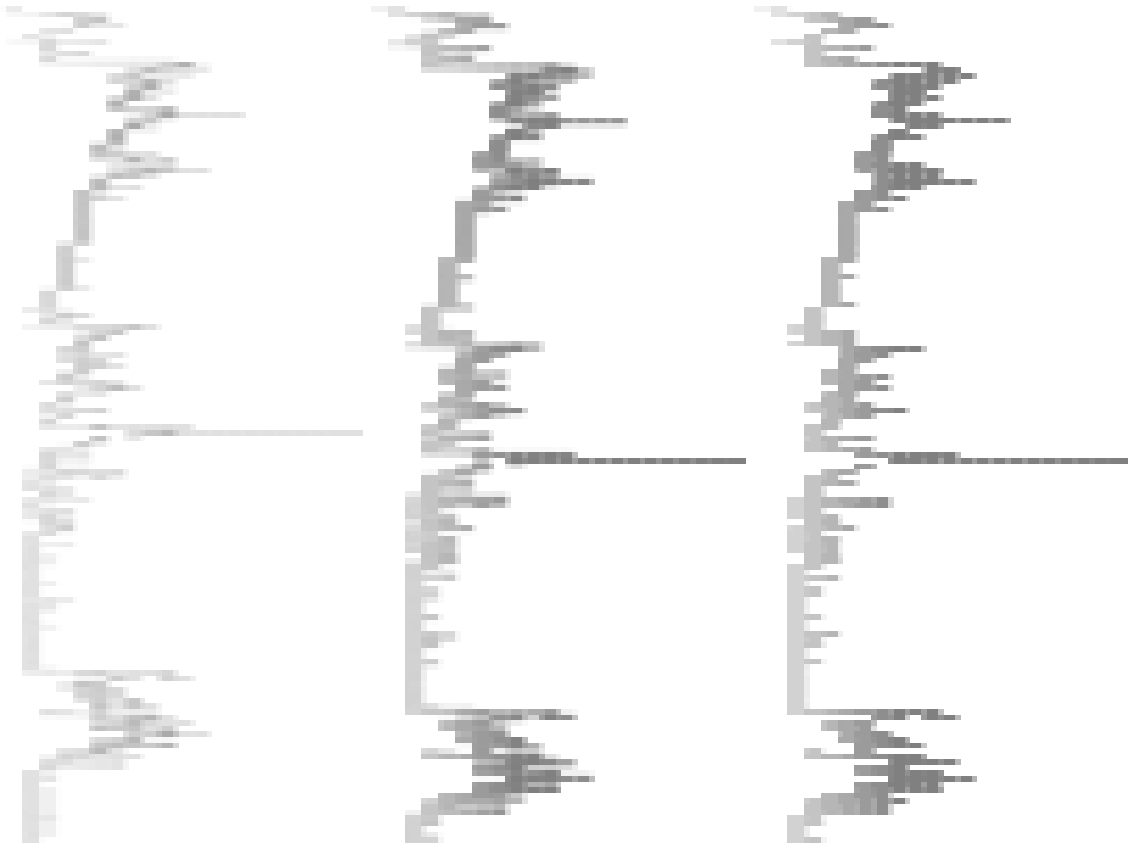
Durch eine Mindesthöhe wird das verhindert. Des weiteren wird die y-Position der Knoten auf ganzzahlige Werte gerundet, damit die Kanten schärfer dargestellt werden.

interact

Die Abbildung 12 zeigt alle Elemente der Visualisierungs-Anwenung. Die Visualisierung selbst wurde mittels der JavaScript Bibliothek D3.js umgesetzt [Bos11].

Der Source-Code der Visualisierung, geschrieben in ECMAScript 6, wird zusammen mit allen Abhängigkeiten in eine Scriptdatei transpiliert.

Vom Browser des Nutzers wird die REST-API des Backends abgefragt. Das Backend führt dann den SQL-Befehl aus und liefert die Daten in Form von JSON an den Browser. Aufgrund der hohen Anzahl von Elementen wird der Kommentarbaum im Browser in einem Canvas-Element gezeichnet und nicht in Form von SVG-Elementen dargestellt.



(a) Normal (b) Höhe min. 1px (c) y Position gerundet

Abbildung 11: Ausgabe des Layouts auf dem HTML-Canvas

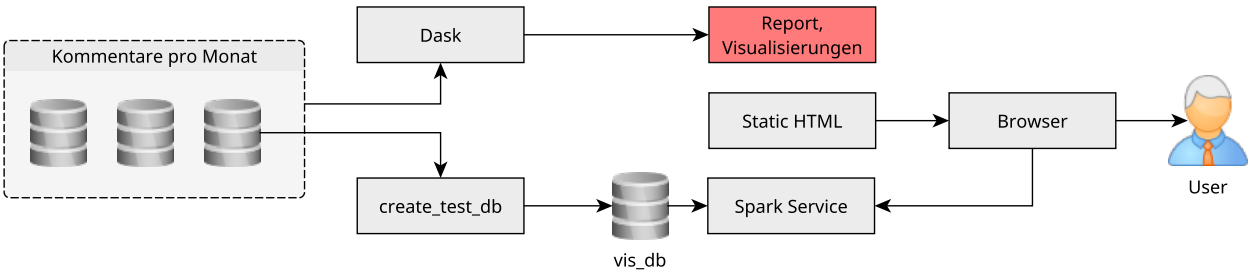


Abbildung 12: Workflow zur Erarbeitung der Visualisierungen

Zusammenfassung

Die vorgestellte Visualisierung zeigt eine Möglichkeit die Struktur von Thread-Diskussionen sehr kompakt darzustellen. Die größten Schwierigkeiten lagen im Bereich der Anwendungsentwicklung und Koordination der heterogenen Webtechnologien.

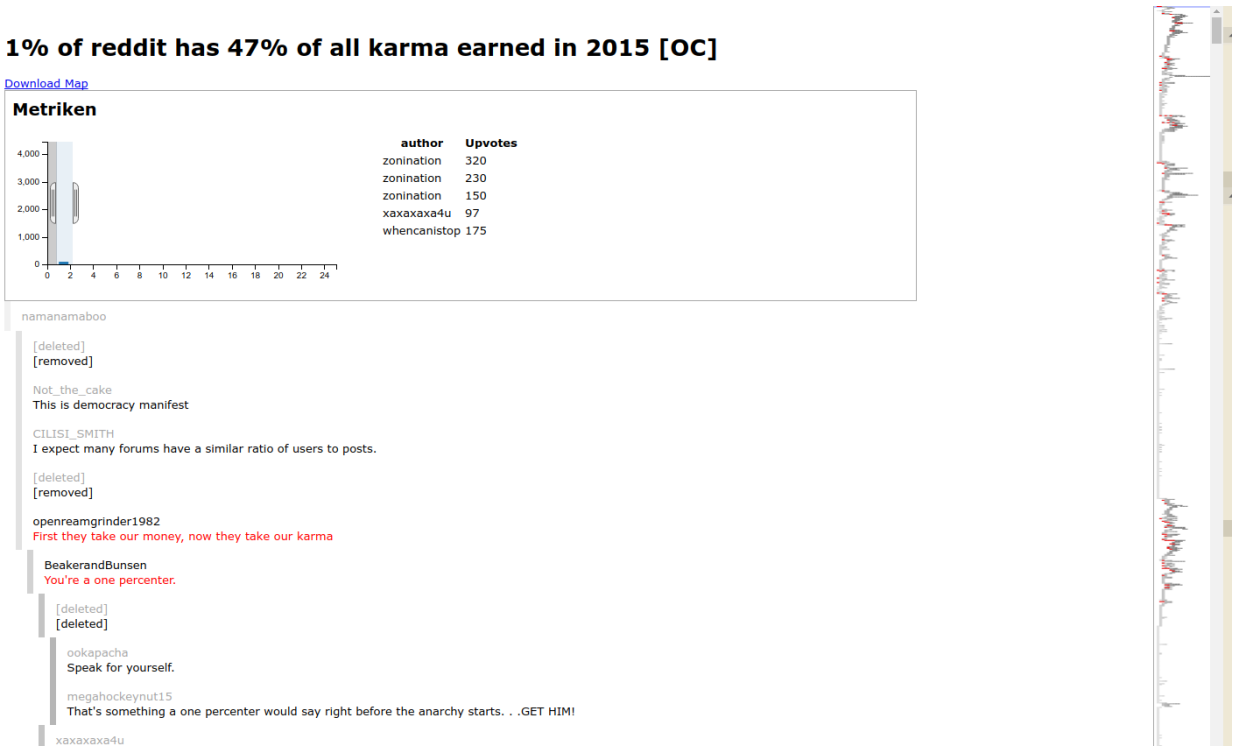


Abbildung 13: Screenshot der Endanwendung

Literatur

[17] *phpBB*. 2017. URL: https://www.phpbb.com/ (besucht am 14. 09. 2017).

[Bau17] Jason Baumgartner. *Reddit Data*. 2017. URL: http://files.pushshift.io/reddit/ (besucht am 13. 09. 2017).

[Bos11] Mike Bostock. *D3.Js*. 2011. URL: https://d3js.org/.

[CKB09] Andy Cockburn, Amy Karlson und Benjamin B. Bederson. „A Review of Overview+Detail, Zooming, and Focus+Context Interfaces“. In: *ACM Comput. Surv.* 41.1 (Jan. 2009), 2:1–2:31. ISSN: 0360-0300. DOI: 10.1145/1456650.1456652. (Besucht am 19. 09. 2017).

[DS13] Maeve Duggan und Aaron Smith. *6% of Online Adults Are Reddit Users*. 3. Juli 2013. URL: http://www.pewinternet.org/2013/07/03/6-of-online-adults-are-reddit-users/ (besucht am 20. 09. 2017).

[Edb13] Jeremy Edberg. „Scaling Reddit from 1 Million to 1 Billion–Pitfalls and Lessons“. 2013. URL: https://www.infoq.com/presentations/scaling-reddit (besucht am 16. 09. 2017).

[Fry04] Benjamin Jotham Fry. „Computational Information Design“. Massachusetts Institute of Technology, 2004. URL: https://pdfs.semanticscholar.org/4a0f/14b9a2c751c8b36e78b14affd6eb20ffd593.pdf (besucht am 02. 05. 2017).

[GS16] Jeffrey Gottfried und Elisa Shearer. *News Use Across Social Media Platforms 2016*. 26. Mai 2016. URL: http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/ (besucht am 20. 09. 2017).

[Mil09] Evan Miller. *How Not To Sort By Average Rating*. 2009. URL: http://www.evanmiller.org/how-not-to-sort-by-average-rating.html (besucht am 20. 09. 2017).

[red] reddit. *Reddiquette*. URL: https://www.reddit.com/wiki/reddiquette (besucht am 20. 09. 2017).

[Roc17] Matthew Rocklin. *Dask: Parallel Computing with Task Scheduling*. 20. Sep. 2017. URL: https://github.com/dask/dask (besucht am 21. 09. 2017).

[Str+] Natalie Jomini Stroud u. a. *Comment Section Survey Across 20 News Sites*.

[Tuf09] Edward R. Tufte. *The Visual Display of Quantitative Information*. Second Edition edition. Cheshire, Conn: Graphics Press, 1. Jan. 2009. ISBN: 978-0-9613921-4-7.