

Was ist Twitter?

Lehrstuhl für Medieninformatik
FAKULTÄT FÜR SPRACH-, LITERATUR- UND
KULTURWISSENSCHAFTEN



Universität Regensburg

Überblick

- 2006: Entwicklung von Twitter von Jack Dorsey, Evan Williams, Biz Stone, Noah Glass als soziales Netzwerk als auch als Mikrobloggingdienst
- Herausbildung von Twitter als soziales Informationsnetzwerk
- 2010: Endgültiger Durchbruch von Twitter durch mobile Geräte
- 2014: über 200 Mio aktive User

Aufbau

- Timeline: chronologische Liste von Tweets
- Tweet: Nachricht mit maximal 140 Zeichen bestehend aus Text, URLs und Medien aus verschiedenen Zusatzdiensten
- Followers: User können Tweets anderer User abonnieren
- Antwort: Nachricht an den Nutzer
- ReTweet: Übernahme eines Tweets in die eigene Timeline
- Favoriten: Pendant zum Like-Button von Facebook

Features

- # (Hashtag): Gruppierung von Posts nach bestimmtem Thema
- @-replies: Verweis auf einen bestimmten Twitter-User
- viele Add-ons für Twitter:
 - Verkürzung von URLs mit bit.ly, ow.ly, TinyURL etc.
 - Lokalisierung von Twitterrends mit Trendsmat
 - Erleichtern von Bilduploads mit Twitpic
 - etc.

Beispiel



Suchmöglichkeiten

- @Username: Suche nach Tweets die User "Username" erwähnen
- #haiku: gesuchter Tweet enthält den Hashtag "haiku"
- Flug :(: enthält "Flug" mit negativer Einstellung
- Verkehr ? : enthält "Verkehr" und stellt eine Frage

Search Tasks und Informationsbedürfnisse der Blogosphäre und speziell auf Twitter/Microblogging

Lehrstuhl für Medieninformatik
FAKULTÄT FÜR SPRACH-, LITERATUR- UND
KULTURWISSENSCHAFTEN



Universität Regensburg

Was ist die Blogosphäre?

- Kollektion aller Blogs im Web
- Unabhängige Blogger produzieren Informationen über so gut wie alle Themengebiete
- Leser können mit Blog interagieren:
 - Kommentare
 - Diskussionen
 - Subscribe
- Blogger sind Produzenten und Konsumenten
- Netzwerk von Blogs durch Links
 - Blogroll links
 - Citation links
 - Linkbacks

Was sind Informationsbedürfnisse der Blogosphäre?

- Web Searchers Info.bedürfnisse kategorisiert nach Broder:
 - Informational
 - Navigational
 - Transactional
- Info.bedürfnisse der Blogosphäre:
 - Name queries
 - News-oriented queries
 - Adult-oriented queries
- Kategorisiert nach:
 - Context query
 - Concept query

Blog Search

- Thema des Blogs spiegelt Interesse des Bloggers wieder
- Wiederholt Artikel/Posts zum Thema
- Sucharten:
 - Adhoc Suche
 - Filtered Suche

Blog Post Search

- Post → Dokument einer Suchmaschine
 - Enthält darüber hinaus Kommentare
 - Bestimmtes Thema
- Sucharten:
 - Adhoc Suche
 - Suche nach Meinungen

Microblog Search

- Beispiel Twitter
 - Post auf 140 Zeichen beschränkt
- Sucharten:
 - Context Query
 - Concept Query
- Gesucht wird v.a. nach aktuellsten Nachrichten

Suchprobleme

- Seriöse Blogs finden, da:
 - Spam-Blogs
 - Fake-Blogs
 - Top Themen sind nur kurzlebig
 - Lokale Beschränktheit
- Microblogs:
 - Aktualität \Leftrightarrow Relevanz
 - Wenig Inhalt, da nur kurze Posts
- Adhoc Suche:
 - Inhalte mit Werbung
 - Umgangssprache der Posts (subjektiv)

Suche auf Twitter

- Viele neue Tweets in kürzester Zeit
- Kurze Tweets
- Aktuelle Informationen über Personen
- Allgemeine, wiederholende Suche
- Fokus auf Sozialem
- Tweets tauchen vollständig in Ergebnisliste auf

Was sind Informationsbedürfnisse auf Twitter?

→ Zeitfaktor

Aktuelle Ereignisse

Trends

Veranstaltungen/Events von Bekannten und Freunden

→ Echtzeitinformationen

→ Sozialfaktor

Interessen, Meinungen anderer Nutzer

Eigene Personenbezogene Informationen

Öffentliche Meinungen

Was sind Informationsbedürfnisse auf Twitter?

- ➔ Thematisch
 - Informationen zu bestimmtem Thema
 - Wiederfinden von Informationen zum Thema

Wie wird auf Twitter gesucht?

- Kurze Suchanfragen
- Hashtags #
- Suche nach Nutzern @
- Suchanfragen zu Prominenten und Top Themen

„Churn“ in Tweets

- Durch Echtzeit ändern sich populäre Suchanfragen sehr schnell
- Beeinflussung der Suche durch Top Themen
- Vokabular ändert sich ständig durch Hashtags → out-of-vocabulary Problem
- Churn bei Tweets 4x höher als bei Web Search

Zusammenfassung

- Eher aktuelle Trends/Monitoring als Wissensaneignung
- Personenbezogene Suchanfragen
- Kurze, allgemeine, wiederkehrende Suchanfragen
- Mehr soziale Inhalte und aktuelle Ereignisse (weniger Fakten)
- Trends
- Spezielle Syntax: #, @



Lehrstuhl für Medieninformatik
**FAKULTÄT FÜR SPRACH-, LITERATUR- UND
KULTURWISSENSCHAFTEN**



Universität Regensburg

Allgemeines

- Inoffizielle Bibliothek für die Twitter API
- Ermöglicht einfache Integration von Twitter in Java – Applikationen
- Basiert auf Java => Läuft auf jeder Java – Plattform (ab Version 5)
- Unterstützt Android und Google APP Engine
- Support für Gzip und OAuth

Voraussetzungen

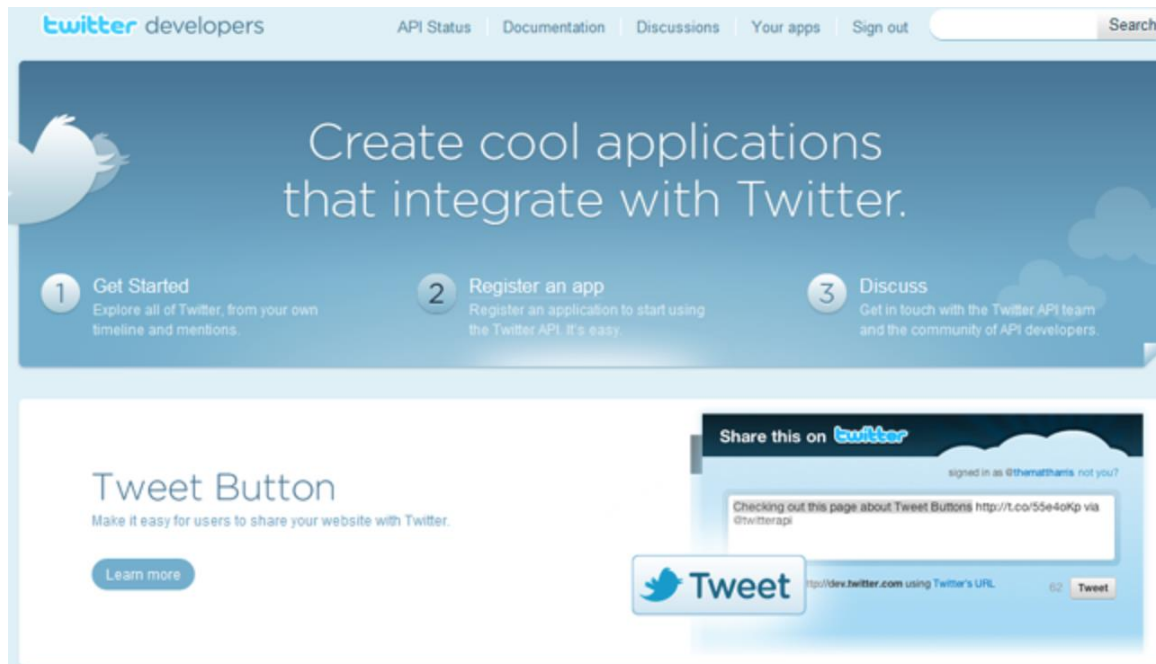
- Windows oder Unix-Variante, die Java unterstützt
- Java 5 oder neuer
- twitter4j-core-4.0.1.jar – Datei

Twitter 4J integrieren (pom.xml)

```
<dependencies>
  <dependency>
    <groupId>org.twitter4j</groupId>
    <artifactId>twitter4j-core</artifactId>
    <version>[4.0,)</version>
  </dependency>
  ...
</dependencies>
```

Zugriff auf Nutzerdaten (OAuth)

- Schritt 1: App unter <http://dev.twitter.com/apps> registrieren
=> consumer key und consumer secret



Zugriff auf Nutzerdaten (OAuth)

- Schritt 2: Nutzer akzeptiert die Anwendung
=> PIN Code

You've successfully granted access to MarvinAPI!

Simply return to MarvinAPI and enter the following PIN to complete the process.

6635676

Zugriff auf Nutzerdaten (OAuth)

- Schritt 3: Durch den PIN erhält man einen Zugriffs-Code und ein Passwort

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret".
Do not share your oauth_token_secret with anyone.

Access token

72241980-9LOun-----081

Access token secret

g6rGf8m5Hd2sPG-----ib4

Access level

Read-only

Recreate my access token

Twitter 4J konfigurieren – twitter4j.properties

```
debug=true  
oauth.consumerKey=*****  
oauth.consumerSecret=*****  
oauth.accessToken=*****  
oauth.accessTokenSecret=*****
```

- Erstellen einer twitter4j.properties – Datei
- Im Verzeichnis der Anwendung speichern

Twitter 4J konfigurieren – ConfigurationBuilder

```
ConfigurationBuilder cb = new ConfigurationBuilder();
cb.setDebugEnabled(true)
  .setOAuthConsumerKey("*****")
  .setOAuthConsumerSecret("*****")
  .setOAuthAccessToken("*****")
  .setOAuthAccessTokenSecret("*****");
TwitterFactory tf = new TwitterFactory(cb.build());
Twitter twitter = tf.getInstance();
```

- Twitter 4J wird im Programmcode konfiguriert
- Erstellen einer Instanz der Klasse ConfigurationBuilder

Aufbau eines Tweets (Tweet)

```
{ "id"=>12296272736,  
  "text"=>  
    "An early look at Annotations:  
    http://groups.google.com/group/twitter-api-announce/browse\_thread/thread/fa5da2608865453",  
  "created_at"=>"Fri Apr 16 17:55:46 +0000 2010",  
  "in_reply_to_user_id"=>nil,  
  "in_reply_to_screen_name"=>nil,  
  "in_reply_to_status_id"=>nil,  
  "favorited"=>false,  
  "truncated"=>false,
```

Aufbau eines Tweets (User)

```
"user"=>
  "id"=>6253282,
  "screen_name"=>"twitterapi"
  "name"=>"Twitter API",
  "description"=>
    "The Real Twitter API. I tweet about API changes, service issues and
    happily answer questions about Twitter and our API. Don't get an answer? It's on my website."
  "url"=>"http://apiwiki.twitter.com"
  "location"=>"San Francisco, CA",
  "profile_background_color"=>"cldfee",
  "profile_background_image_url"=>
    "http://a3.twimg.com/profile_background_images/59931895/twitterapi-background-new.png",
  "profile_background_tile"=>false,
  "profile_image_url"=>"http://a3.twimg.com/profile_images/689684365/api_normal.png",
  "profile_link_color"=>"0000ff",
  "profile_sidebar_border_color"=>"87bc44",
  "profile_sidebar_fill_color"=>"e0ff92",
  "profile_text_color"=>"000000",
  "created_at"=>"Wed May 23 06:01:13 +0000 2007",
  "contributors_enabled"=>true,
  "favourites_count"=>1,
  "statuses_count"=>1628,
  "friends_count"=>13,
  "time_zone"=>"Pacific Time (US & Canada)",
  "utc_offset"=>-28800,
  "lang"=>"en",
  "protected"=>false,
  "followers_count"=>100581,
  "geo_enabled"=>true,
  "notifications"=>false,
  "following"=>true,
  "verified"=>true}
```

Aufbau eines Tweets (Ort)

```
"place"=>
{
  "id"=>"2b6ff8c22edd9576",
  "url"=>"http://api.twitter.com/1/geo/id/2b6ff8c22edd9576.json",
  "name"=>"SoMa",
  "full_name"=>"SoMa, San Francisco",
  "place_type"=>"neighborhood",
  "country_code"=>"US",
  "country"=>"The United States of America",
  "bounding_box"=>
  {
    "coordinates"=>
    [
      [[-122.42284884, 37.76893497],
       [-122.3964, 37.76893497],
       [-122.3964, 37.78752897],
       [-122.42284884, 37.78752897]],
      "type"=>"Polygon"
    ]
  },
  "source"=>"web"
}
```

Codebeispiel #1 – Eine Timeline auslesen

```
// The factory instance is re-useable and thread safe.  
Twitter twitter = TwitterFactory.getSingleton();  
List<Status> statuses = twitter.getHomeTimeline();  
System.out.println("Showing home timeline.");  
for (Status status : statuses) {  
    System.out.println(status.getUser().getName() + ":" +  
                        status.getText());  
}
```

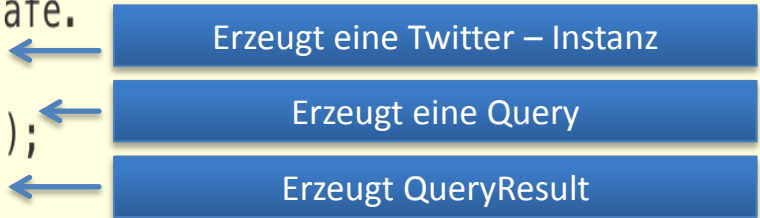
Erzeugt eine Twitter – Instanz

Erzeugt eine ArrayList

- ArrayList „statuses“ wird durch twitter.getHomeTimeline() befüllt
- Die Liste wird dem Nutzer ausgegeben

Codebeispiel #2 – Tweets suchen

```
// The factory instance is re-useable and thread safe.  
Twitter twitter = TwitterFactory.getSingleton();  
Query query = new Query("source:twitter4j yusukey");  
QueryResult result = twitter.search(query);  
for (Status status : result.getTweets()) {  
    System.out.println("@ " + status.getUser().getScreenName() + ":" + status.getText());  
}
```



- Query-Objekt wird mit dem Wert „source:twitter4j yusukey“ erstellt.
- Objekt „result“ wird durch die Methode twitter.search(query) befüllt. (Return-Type ist eine ArrayList)
- Die Ergebnisse werden einzeln ausgegeben.

Codebeispiel #3 - Streaming API

```
public static void main(String[] args) throws TwitterException, IOException{
    StatusListener listener = new StatusListener(){
        public void onStatus(Status status) {
            System.out.println(status.getUser().getName() + " : " + status.getText());
        }
        public void onDeleteNotice(StatusDeletionNotice statusDeletionNotice) {}
        public void onTrackLimitationNotice(int numberOfLimitedStatuses) {}
        public void onException(Exception ex) {
            ex.printStackTrace();
        }
    };
    TwitterStream twitterStream = new TwitterStreamFactory().getInstance();
    twitterStream.addListener(listener);
    // sample() method internally creates a thread which manipulates TwitterStream and calls the
    // adequate listener methods continuously.
    twitterStream.sample();
}
```

← StatusListener

← Leere Methoden

← TwitterStream + Listener

- StatusListener implementieren
- Thread wird erzeugt
- Instanz der TwitterStream-Klasse erzeugen
(TwitterStreamFactory().getInstance())
- Listener auf den twitterStream setzen

Twitters Streaming API vs. Twitters Firehose

Lehrstuhl für Medieninformatik
FAKULTÄT FÜR SPRACH-, LITERATUR- UND
KULTURWISSENSCHAFTEN



Universität Regensburg

Datenrepräsentation

Twitter ist eins der wenigen sozialen Netzwerke das ihre Daten zur Verfügung stellt

Angebot von Twitter selbst:

- API
- Firehose

Twitters Streaming API

- Kostenlos
- Limitiert
- Wird häufig benutzt
- Mangelhafte Dokumentation über Datenausgabe hinsichtlich Art und Menge

Ist die API eine gute Repräsentation der Aktivitäten auf Twitter?

API nimmt 3 Arten von Parametern an:

- Schlüsselworte (Worte, Phrasen, Hashtags)
- Geographische Begrenzungen
- Benutzer IDs

Resultat: Sobald die API über 1% der Tweets die zu den Parametern passen kommt, werden die Daten an den Benutzer zurückgegeben (Nachteil)

Alternative: Firehose

Firehose

Feed, der erlaubt auf 100% der Daten auf Twitter zuzugreifen

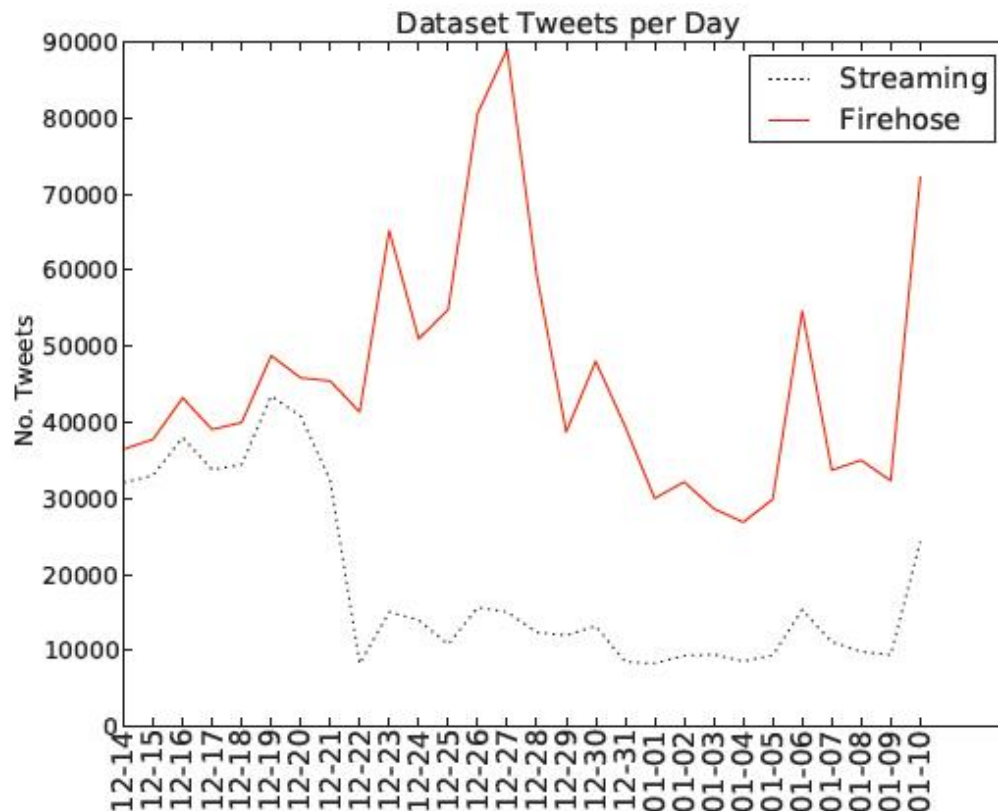
Nachteile:


- Kostenpflichtig
- Aufwendige Ressourcen (Speicher, Server, Netzwerk Verfügbarkeit)

Vergleich

Sammeln von rohen Tweetdaten innerhalb des gleichen Zeitraums mit den gleichen Parametern sowohl mithilfe von der API als auch mit Firehose

Vergleich



Keywords	Geoboxes	Users
#syria, #assad, #aleppovolcano, #alawite, #homs, #hama, #tartous, #idlib, #damascus, #daraa, #aleppo, #سوريا*, #houla	 (32.8, 35.9), (37.3, 42.3)	@SyrianRevo

* Arabic word for “Syria”

Geotagging

Überraschenderweise konnte die Streaming API trotz des Samplings fast die gesamte Anzahl der geomarkierten Tweets zurückgeben

Möglicher Grund wäre die Begrenzung der Ortungsdaten

Wiederholung mit Ausschluss der vorher gesammelten Tweets

→ größere Ungleichheit

=> Zuversicht, dass ortsmarkierte Tweets eine große Abdeckung ergeben

Fazit

Die Ergebnisse der Streaming API hängen stark von der Abdeckung und dem Typ der Analyse ab die ausgeübt werden soll

Führt zu der Frage wie groß die Datenmenge ist die man innerhalb eines Zeitrahmens bekommt, worin auch noch Unklarheit besteht.

Exploratory Search/ Exploratory (Tweet) Search Engine

Lehrstuhl für Medieninformatik
**FAKULTÄT FÜR SPRACH-, LITERATUR- UND
KULTURWISSENSCHAFTEN**

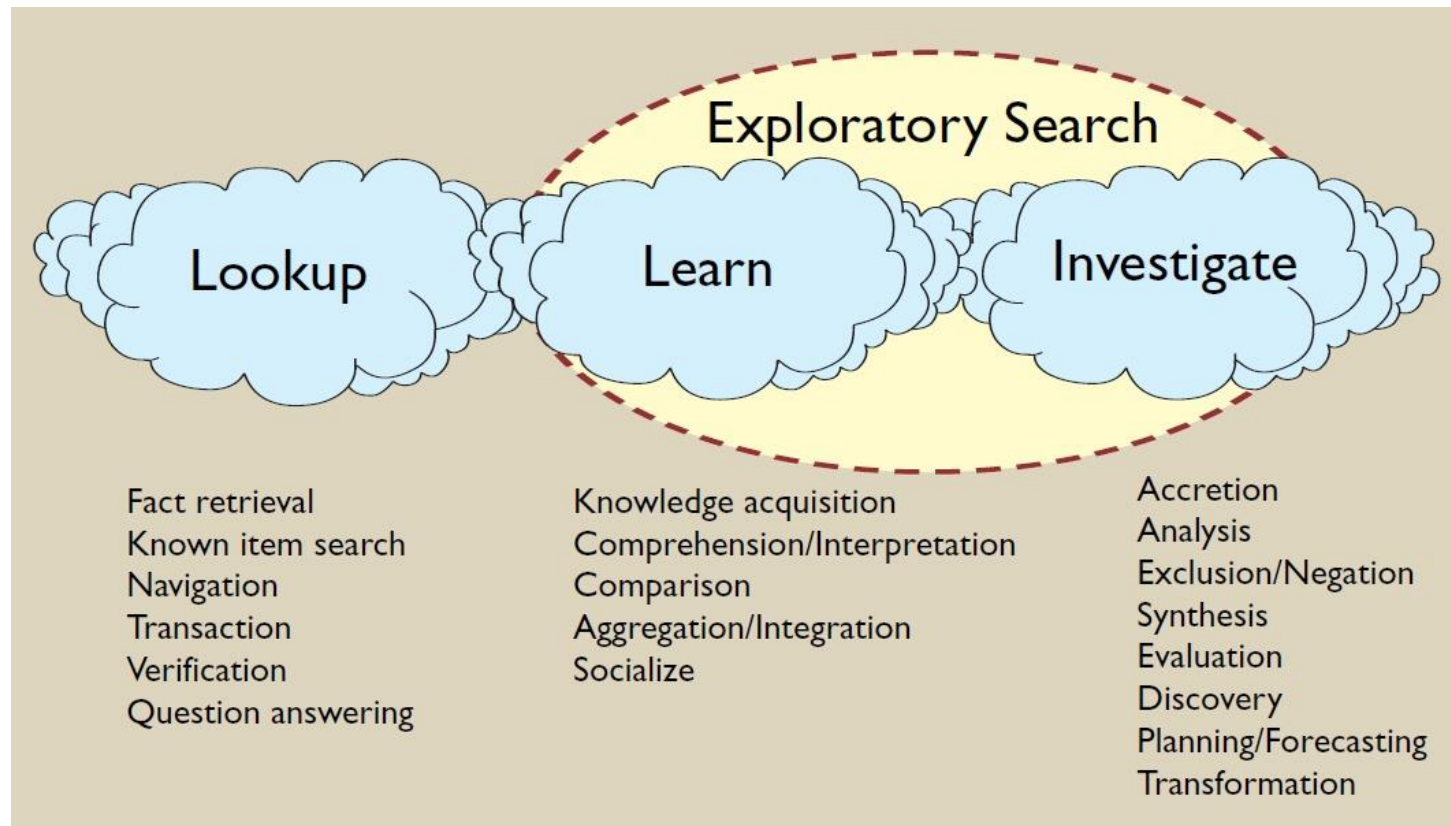


Universität Regensburg

Exploratory Search allgemein

- Art des **Informationssuchverhaltens**
- Gary Marchionini: **3 Arten** von Informationssuchverhalten
 - „Lookup“
 - „Learn“
 - „Investigate“
- „Learn“ + „Investigate“ -> „Exploratory search“

Schaubild Informationssuchverhalten (Marchionini)



„Lookup“

- Alltägliches Suchverhalten
- Meist vertrautes Thema („known item search“)
- Reines Faktenretrieval
- Zahlen, Geburtsdaten, Namen, ...

„Learn“ + „Investigate“

- **Komplexere Zielsetzung** als beim simplen „Lookup“
- Komplexes Themengebiet soll **erschlossen** („Learn“ -> „knowledge acquisition“) und **erforscht** („Investigate“) werden
- **Ausgangssituation:**
 - Kein/ unzureichendes Wissen über den Themenkomplex
 - Kein/ unzureichendes Wissen über die Suchstrategien/ -techniken, die zum Ziel führen
 - Kein/ unzureichendes Wissen über das Ziel allgemein

Vorgehensweise „exploratory search“

- Nutzer sammelt/ erschließt sich Wissen auf dem Weg/ während des Retrievalprozesses (Vage Anfangsquerie -> Stück für Stück zum Ziel)
- Nutzer kombiniert verschiedenste Techniken, um Informationen zu sammeln:
 - Querys
 - Browsing
 - Navigation
 - Selection
- Trial-and-Error Prinzip
- Unterstützung des Nutzers durch Retrievalsystem/ Search Engine wichtig

Bekannte Vertreter

- Gary Marchionini (3 facets of search behaviour)
- Marcia Bates (berry picking)
- Nicolas Belkin (ASK-Hypothese)

Anforderungen an eine Exploratory Search Engine

- „exploratory search“- Nutzer **benötigen besondere Unterstützung** beim Suchprozess durch das Retrievalsystem/ die Search Engine
- **Problem:** Kurze Suchanfragen in klassischen Suchmaschinen reichen ab einem bestimmten Komplexitätslevel nicht mehr aus, um das Informationsbedürfnis des Nutzers zu befriedigen
- Erweiterte **Interface- und Interaktionsoptionen** nötig
- Dynamische **Verwaltung, Verknüpfung** und **Analyse** von gefundenen Informationen

Informationsvisualisierung

- Highlighting wichtiger Informationen
- Zusammenhänge visualisieren
- Graphiken

->sense making, information foraging/ information scent

Dokument-Clustering

- Zusammenfassung von Dokumenten
- Gruppierung der Dokumente nach Themen

->besserer Überblick, Zusammenhänge werden ersichtlich

Dynamic queries

- Auswirkungen einer best. Aktion werden dem Nutzer sofort angezeigt
- Mögliche Optionen/ Hilfen zur „query refinement“ werden angeboten

->berry picking

Browsing-Systeme

- Unterstützung der Navigation während des Suchprozesses
- Beispiel: „faceted-search“ Interfaces

-> bessere, schnellere Orientierung innerhalb des Retrievalsystems/ der Search Engine

Ziele

- Search Engine unterstützt den Nutzer, den **richtigen Weg** auf seiner Suche nach Informationen zu finden
- **Maximal möglicher Erfolg** soll erreicht werden
- Nutzer soll die gewonnen Informationen **verstehen** und sie in einen **(richtigen) Zusammenhang** setzen können

„TweetMotif“ als Beispiel einer Exploratory (Tweet) Search Engine

- **Projekt** von Brendan O'Connor (Carnegie Mellon University), Michel Krieger (Meebo) und David Ahn (Microsoft)
- Mittlerweile **nicht mehr online** (experimenteller **Prototyp**)
- [Demopaper](#) für ICWSM 2010
- Exploratory Search Engine für den Anwendungsfall Twitter

„TweetMotif“ Startseite



(tweet this) (about) (tips)

What are people saying about...

Search

Trending topics

Jay-Z • #phrasesihate • Oprah • #Iamproudof •
Grey's Anatomy • Chiodos • Craig Owens • G20 •
G-20 • ODS

Or try

sandwich • coffee • :) • :(• aw • awwwww •
@the_real_shaq • @twitter • "san francisco"
weather • tweetmotif

related themes

over the last 7 minutes

baum and liberty	#resistg20
#pittsburgh	police fire
fire tear gas	teargas
at g20 protesters	for g20
tinyurl.com/ybxsxzm	kdka
at g20 summit	baum
g20 trend	teargas used
soft rounds	anarchists
g20 protesters	protesters
west penn	stopping traffic
hospital	protestors
tear gas	rt @amycnn / #
protest g20	as g20
protesters are	riot police
twitpic.com/izcyo	lawrenceville
africa bit.ly/2tfyar	@jimlokay
armored	@pgpolitweets
what	winchiddle

tweets by theme

“baum and liberty”

teargas, rubber bullets, 1 arrest on [baum and liberty](#) beat people with batons, shot with rubber bul [#resistg20](#) [#g20](#) [resistg20](#) [show 1 similar tweet »](#)

Protestors at [Baum and Liberty](#). Soft rounds fired near Ritter's diner. Windows broke at Boston Market. Scanner. [#G20](#) [cpdavngr](#) [show 2 similar tweets »](#)

Minor scuffle at [Baum and Liberty](#). Some protesters threw rocks and bricks at police.[#g20](#) [#Pittsburgh](#) [#CNN](#) [amyCNN](#) [show 2 similar tweets »](#)

police boxing in protesters in parking lot - [baum and liberty](#) [#g20](#) [g20](#) [#resistg20](#) [SocialistZine](#) [show 1 similar tweet »](#)

(drilldown 1)

“police fire”

RT @SkyNews Riot [Police Fire](#) Tear Gas at [G20](#) Protesters: Protesters & police are clashing on the streets of Pittsburgh [http://bit.ly/wXvQA](#) [KM_Zencat](#)

RT@DrudgeReport Drudge [Police fire](#) tear gas at [G20](#) protesters [http://tinyurl.com/ybxsxzm](#) How much tear gas did they use on 912 marchers NONE [LibertyinOhio](#) [show 1 similar tweet »](#)

[Police fire](#) tear gas at [G20](#) protesters [http://tinyurl.com/ybxsxzm](#) [wlfurant](#)

Funktionsweise

- Darstellung der gefunden Tweets **gruppiert** nach häufigen Termen/Themen
- Terme/Themen bestimmt durch **syntaktische Filterung, language models** und **Heuristiken**

Interface

- Verwendung eines „**faceted-search**“ Interfaces
- 2 Spalten:
 - **Links**: einzelnen Terme/ Themen, nach denen die Tweets sortiert/ gruppiert wurden
 - **Rechts**: konkreter Beispieltweet bei entsprechender Themenauswahl in der linken Spalte
- Oben: klassische Suchleiste, darunter eine Auswahl von Suchvorschlägen und „trending topics“