API’s für den Zugriff auf Daten

aus sozialen Netzwerken

zur Sentimentanalyse bzw.

zum Option Mining.

Julian Nischler, B.Sc.

(S1310457017)

SEMINARARBEIT

Eingereicht am

Fachhochschul-Masterstudiengang

Information Engineering Management

in Hagenberg

im Juni 2014

Diese Arbeit entstand im Rahmen des Gegenstands

Seminar aus Information Management

(SIM)

im

Sommersemester 2014

Betreuer:

DI. Dr. Gabriel KRONBERGER

Inhalt

[1 Einführung 1](#_Toc390271883)

[1.1 Motivation 1](#_Toc390271884)

[1.2 Fragestellungen 2](#_Toc390271885)

[1.3 Zielsetzung und Methodik 2](#_Toc390271886)

[2 „soziale“ API’s 3](#_Toc390271887)

[2.1 Funktionsbeschreibung und Entwicklung 3](#_Toc390271888)

[2.2 Übersicht populärer API’s 7](#_Toc390271889)

[2.3 Facebook 8](#_Toc390271890)

[2.4 Twitter 11](#_Toc390271891)

[2.5 Historische Daten 14](#_Toc390271892)

[3 Analyse der Daten 15](#_Toc390271893)

[3.1 Generelles 15](#_Toc390271894)

[3.2 Analysetools und API’s 16](#_Toc390271895)

[4 Praxisbeispiel 17](#_Toc390271896)

[4.1 Generelles 17](#_Toc390271897)

[4.2 Twitter Stream 17](#_Toc390271898)

[4.3 Analyse mittels DatumBox 17](#_Toc390271899)

[5 Schlussbetrachtung **Fehler! Textmarke nicht definiert.**](#_Toc390271900)

[6 Quellen 19](#_Toc390271901)

[7 Weiterführende Literatur 20](#_Toc390271902)

# Einführung

## Motivation

Soziale Netzwerke sind von der derzeitigen „Thumb“-Generation nicht mehr weg zu denken. Netzwerke wie Facebook erfahren einen enormen Aufschwung an Nutzern und Verwendung. Kaum jemanden sind diese Netzwerke kein Begriff. Diese Arbeit beschäftigt sich nicht mit den Vor- bzw. Nachteilen solche Netzwerke, vielmehr mit den nicht direkt ersichtlichen Möglichkeiten, den API’s. Viele Unternehmen fürchten einen Auftritt in solchen Netzwerken da sich negative Informationen bzw. Meinungen sehr schnell verbreiten und nur sehr schwer eindämmen lassen. In vielen Fällen lässt sich eine Präsenz jedoch nicht vermeiden, so auch in meinem Arbeitsumfeld. Tätig im Eventbereich sind wir auf solche Netzwerke zur Kundengewinnung stark angewiesen. Da diese Netzwerke völlig neue Marketingtechniken eröffnen. Es ist verhältnismäßig sehr günstig neue Kunden zu bewerben. Leider sind auch wir immer wieder mit der Problematik der Bildung einer negativen Meinung konfrontiert. Mittels der API’s der Netzwerke soll versucht werden negative Meinungen schnell zu erkennen um eine Ausbreitung zu verhindern.

Aus der einleitenden Motivation ergeben sich folgende Kernthemen für die Seminararbeit:

* Einführung in die Thematik „soziale“ API‘s
* Evaluierung der einzelnen API’s
* Kombination mit der Thematik „Sentiment Analyse“

## Fragestellungen

Folgende zentrale Fragestellungen sollen in der Seminararbeit beantwortet werden.

1. Sozial API spezifische Fragen
   1. Was ist eine „soziale“ API?
   2. Welche API’s existieren? (Auszug)
   3. Welche API für welchen Zweck geeignet?
   4. Wie werden die API’s verwendet?
2. Fragen in Verbindung mit der Daten Analyse
   1. Was ist Sentiment Analysis?
   2. Wie gut eignen sich die API’s dafür?
   3. Problematiken der Sentiment Analyse?
   4. Welche Frameworks existieren?

Welche API’s existieren, wie werden diese verwendet, sowie eignen sich diese zur Sentimentanalyse sprich zur Erkennung von Meinungsbildungen?

## Zielsetzung und Methodik

Ziel dieser Arbeit ist, dem interessierten Leser einen Einstieg in die Verwendung von „sozialen“ API’s zu vermitteln, sowie in die Thematik der Sentimentanalyse. Aufbauend auf diesen Erkenntnissen soll gezeigt werden wie durch Sentimentanalyse der verfügbaren Daten profitiert werden kann und welcher Mehrwert aber auch welche Risiken dadurch entstehen. Anhand von Praxisbezogenen Beispielen wird versucht die Thematik zu veranschaulichen.

# „soziale“ API’s

## Funktionsbeschreibung und Entwicklung

Dieses Kapitel soll einen Überblick der am Stärksten verbreiteten API’s schaffen. Das Internet ist heutzutage eines der am Meisten verwendeten Medien zum Verbreiten von Information. Vor allen durch soziale Netzwerke wie Facebook, Twitter, YouTube und Co. Um dem Benutzer eine noch bessere Erfahrung zu ermöglichen gibt es unzählige Anwendungen welche die einzelnen Plattformen verknüpfen. Eine sehr bekannte solche Anwendung ist „Instagram“. Mittels dieser ist es möglich sehr einfach und schnell Fotos mit einem Smartphone zu erstellen, bearbeiten und vor allem zu verteilen. Dabei wird das neue Foto mit nur einem Klick in nahezu allen sozialen Netzwerken verteilt oder geteilt. Solche Anwendungen werden von den Entwicklern meist „mashups“ genannt, sprich die nahtlose Kombination von Plattformen und Innhalten.

So gut wie alle sozialen Netzwerke stellen leistungsstarke API’s zur Verfügung, wodurch solche „mashups“ überhaupt erst ermöglicht werden. Fast ausschließlich fungieren diese in Form von Web Services[[1]](#footnote-1). Die Anzahl der verfügbaren API’s ist in den letzten Jahren förmlich explodiert, so stieg die Anzahl der API’s von etwa 200 Anfang 2006 auf über 10000 im Jahr 2013 und dabei handelt es sich lediglich um auf „ProgrammableWeb“ registrierte API‘s. Abbildung 1 veranschaulicht das Wachstum der letzten Jahre. (1)

Viele frühen API’s verwendeten SOAP[[2]](#footnote-2) ein Netzwerkprotokoll mit dessen Hilfe Daten zwischen Systeme ausgetauscht werden können, zur Repräsentation der Daten wird XML[[3]](#footnote-3) verwendet.

Durch das enorme Wachstum von Webanwendungen in den letzten Jahren wurden einfachere Protokolle gefordert, welche mittels einfacher Skriptsprachen wie JavaScript verwendet werden können. SOAP war nie für Webanwendungen gedacht, dessen Anwendungsbereich liegt im Enterprise Bereich.

Weshalb sich in den letzten Jahren das REST[[4]](#footnote-4) Programmierparadigma durchsetzte und mittlerweile von mehr als 70% aller API’s angeboten wird. Abbildung 2 zeigt die Entwicklung von REST in Vergleich zu SOAP in den letzten Jahren. (1)

REST beschreibt strenggenommen kein Protokoll auch keine Norm es beschreibt eigentlich lediglich die Methodik einer Web URL genau einer Serverseitigen Aktion zuzuordnen. Als Protokoll wird HTTP oder HTTPS verwendet. Dabei werden die folgenden HTTP Befehle *(HTTP-Verb)* verwendet.

* GET
  + Anfordern der angegeben Ressource (URL), es wird nur gelesen.
* POST
  + Zusätzlich zum Anfordern der Ressource werden zusätzlich Nutzdaten vom Client an den Server übertragen. Es wird gelesen und geschrieben.
* DELETE
  + Löscht die angeforderte Ressource.

Sowie die zusätzlichen Befehle *PATCH, PUT, HEAD, OPTIONS, CONNECT* und *TRACE* auf diese wird im Zuge dieser Arbeit nicht weiter eingegangen da Diese im Kontext nicht von Bedeutung sind.

REST Spezifiziert auch nicht die Form der Datenrepräsentation. Ein sehr praktisches und einfaches Datenformat stellt JSON dar, da es direkt von JavaScript interpretiert werden kann. Weshalb sich JSON nahezu gleich mit REST verbreitet so bieten mittlerweile weit über 50% der API’s das JSON Format an. Viele Moderne API’s wie Facebook Open Graph bieten nur noch JSON an.

In Abbildung 3 wird ein einfaches Beispiel gezeigt. Mittels eines REST GET Request wird das Wetter von Salzburg abgefragt. Der Web Service antwortet wahlweise mit XML oder JSON. Es ist ersichtlich das JSON Daten leichter lesbar sind und wirken übersichtlicher. Auf den ersten Blick würde man nicht glauben das die JSON Daten kleiner als die XML Repräsentation. JSON brauch für die Darstellung Vertikal zwar viel mehr Platz ist jedoch deutlich kleiner. 441 Bytes JSON gegen 659 Bytes XML.

Abbildung 1 Wachstum Anzahl angebotene API’s auf ProgrammableWeb

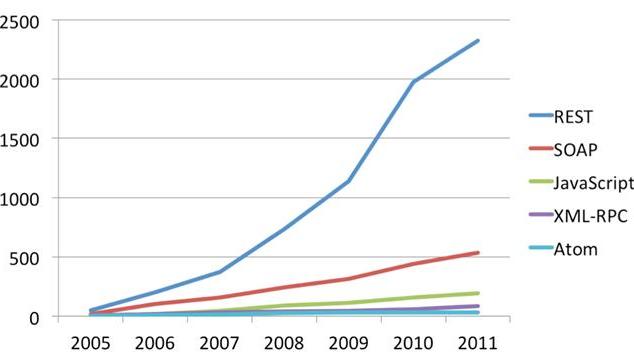


Abbildung 2 Vergleich REST zu SOAP und Alternativen

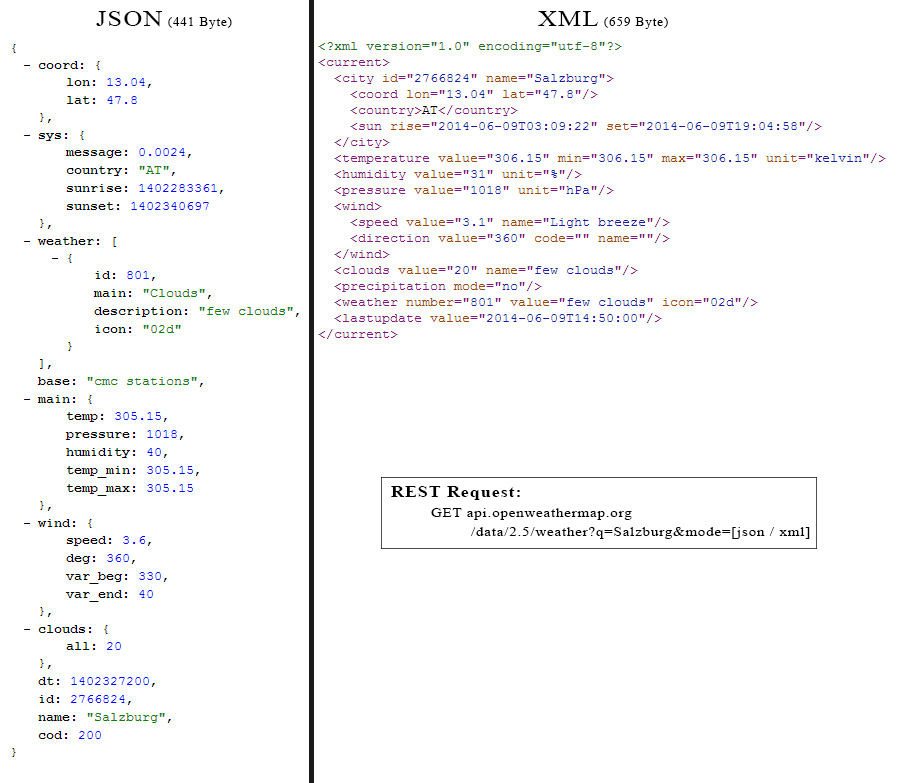


Abbildung 3 Beispiel für REST Request mit XML bzw. JSON Antwort

## Übersicht populärer API’s

### Facebook

Siehe 2.3 Facebook.

### Twitter

Siehe 2.4 Twitter.

### Google+

Google bietet mit Google+ ein soziales Netzwerk welches im Funktionsumfang Facebook ähnelt. Mittels der verfügbaren REST API ist es möglich Personen, Aktivitäten, Kommentare sowie Momente anzufragen und erstellen. Zusätzlich können Abfragen auch mittels Volltextsuche formuliert werden. Dadurch kann man sehr schnell Informationen einer Person oder Zielgruppe erhalten. Google verwendet vorzugsweiße JSON als Datenformat. (2)

### bitly

Bitly zählt zu den am häufigsten verwendeten „URL-shorten“ Diensten. Dieser Dienst wird vor allem gerne für Twitter-Nachrichten verwendet um die maximale Nachrichtenlänge nicht zu überschreiten. So werden Täglich rund 1 Million bitly links erstellt und über 4 Billionen pro Monat angeklickt. Bitly bietet API’s mit denen es Möglich ist die Populärsten Links ausfindig zu machen. Und gerade im Business Intelligence Bereich sehr interessant die Verbreitung von Phrasen bzw. Innhalten der Zielseiten. Als interessantes Beispiel zeigt bitly eine Echtzeitkarte, welche die Verteilung von Artikel der einzelnen Nachrichtenportale in den USA zeigt. (3)

### del.icio.us (delicious)

Delicious ermöglicht es dem Benutzer Lesezeichen abzulegen und diese von all seinen „smart - Devices“ abzufragen. Lesezeichen werden nicht in Ordner gespeichert sondern mit Tags versehen, dies geschieht Teilweise automatisch. Mit den von delicious zur Verfügung gestellten API ist es leider nur Möglich Daten des aktuellen Benutzers abzufragen, diese Einschränkung ist sehr bedauerlich. Da Benutzer die Links bereits mit Tags klassifizieren hätten diese Daten einen sehr hohen Mehrwert für unter anderem das Suchen und Finden von Referenzen oder Querverweisen.

### Foursquare

Foursquare der wohl bekannteste Geotagging Dienst. Grundsätzlich eine Weltkarte auf der Benutzer Plätze eintragen können und diese auch bewerten. In den Letzten Jahren ist Foursquare stark gewachsen und weißt mittlerweile eine der größten Geodatenbanken auf. Mittels der angebotenen API’s ist es möglich anhand von Koordinaten oder Suchbegriffe Plätze ausfindig zu machen. So ist es unter anderem Möglich dem Benutzer Informationen aus seiner Umgebung anzubieten. (4)

## Facebook

Facebook das bekannteste und größte soziale Netzwerk der Welt mit Monatlich über 1,2 Billionen aktive Nutzer bietet eine Vielzahl an API’s.

Am bekanntesten sind die „sozialen Plug-Ins“ wie die „Like“ oder die Teilen –Schaltfläche. Diese lassen sich sehr einfach in bestehende Anwendungen integrieren. Ein Trend der Internet Nutzer besteht darin möglichst viel Inhalt mit anderen zu teilen, so verwenden immer mehr Firmen Facebook und Co als mächtiges Marketingtool. Klick ein Benutzer zum Beispiel auf eine solche „Like“ Schaltfläche, innerhalb einer Anwendung, werden dessen Freunde über die Interaktion benachrichtigt, so Mancher wird die Anwendung genauer Betrachten und eventuell sogar in Zukunft nutzen. Dadurch öffnen sich für das Marketing völlig neue Türen.

Immer mehr Unternehmen wagen den Sprung in die sozialen Netzwerke um auf einfachen Weg neue Kunden finden zu können. Geblendet von den neuen Möglichkeiten bleiben sehr oft die möglichen Nebenwirkungen unbedacht. Denn verbreiten sich positive Meinungen sehr schnell in sozialen Netzwerken, so tuen dies Negative noch viel rasanter. Und ist erst einmal eine negative Meinung gebildet so wächst diese wie ein Geschwür innerhalb der Netzwerke. In vielen Fällen hilft dann nur noch eine Notbremsung sprich das Entfernen des „sozialen“ Auftrittes.

Auftritte von Unternehmen innerhalb sozialer Netzwerke müssen stets überwacht werden und dabei die herrschende Meinung kritisch hinterfragen. Aus Erfahrung lässt sich ableiten es wird in etwa pro 100 tausend Nutzer oder „Likes“ ein Sozialmanager benötig. Dessen Aufgabe in der Überwachung nicht jedoch im eigentlichen Marketing liegt.

Facebook’s mächtigste API‘s sind die Daten API‘s diese können auch einem Sozialmanager sehr in die Karten spielen da mittels dieser API‘s Daten strukturiert abgefragt werden können. Diese API’s sind natürlich nicht ausschließlich für BI Operationen gedacht, sondern dienen allen möglichen Interaktionen mit dem sozialen Netzwerk. So lässt sich mittels dieser API zum Beispiel eine Anwendung erstellen welche im Namen des Nutzers automatisch folgenden Post erstellt. „Ich habe soeben die Anwendung XYZ installiert, versuche diese doch auch [LINK]“

### FQL

FQL ist eine dieser Daten API’s, steht abgekürzt für **F**acebook **Q**uery **L**anguage. Mittels dieser API ist es möglich SQL ähnliche Queries auszuführen. Gegenüber den anderen API’s hat diese einen großen Vorteil, die Daten können bereits gefiltert werden es sind sogar logische Verknüpfungen möglich. Mittels dieser API ist kein schreiben oder löschen möglich es kann nur gelesen werden.

Folgender FQL Befehl liefert einige Felder des aktiven Users.

*SELECT uid, name, pic\_square FROM user WHERE uid = me()*

JSON Antwort:

{

"data": [

{

"uid": "1226612954",

"name": "Julian Nischler",

"pic\_square": "https://scontent-b.xx.fbc .... 90\_n.jpg"

}

]

}

Für den BI Bereich viel interessanter ist das Abfragen von Post an einer Seite. Lässt sich mittels eines FQL Queries sehr leicht formulieren es lassen sich mittels FQL auch gleich alle eigenen Posts entfernen.

SELECT actor\_id, message, created\_time

FROM stream

WHERE source\_id = 8419133006 and actor\_id != 8419133006

limit 100

Obiger Query gibt die letzten 100 Posts von Nutzern auf der Seite des *„Eurovision Song Contest“*(uid: 8419133006) zurück. Etwas verwirrend im Query ist source\_id die uid der Seite und actor\_id die uid der postenden Person. Die Antwort ist wie im obigen Beispiel als JSON verfügbar und könnten nun mittels geeigneten BI Tools ausgewertet werden.

Facebook hat vor kurzem die FQL API als veraltet gekennzeichnet und diese wird im Jahr 2016 auslaufen. Dies ist vermutlich ein weiterer Schritt in Richtung der geplanten neuen „GraphSearch“ API welche mittels Volltextsuche funktioniert. Weitere Informationen zu dieser neuen Funktion sind Verfügbar via folgendem Link.

<https://www.facebook.com/about/graphsearch>

### Graph API

Die Graph API stellt die de facto Standard API von Facebook da und ist eine alternative zu FQL, diese wird auch intensiv weiter entwickelt und ist nicht als veraltet gekennzeichnet. Im Unterschied zu FQL ist es möglich mittels dieser API sowohl möglich Daten zu schreiben und zu löschen. Befehle werden nicht in Form eines Query gesendet sondern als normale URL.

Zur Veranschaulichung ein Befehlt welcher wieder alle Posts auf der Seite des USC[[5]](#footnote-5) anfragt.

<http://graph.facebook.com/v2.0/EurovisionSongContest/feed?fields=from,message,created_time>

Mittels dieser API ist es nicht möglich die Ergebnisse zu Filtern, es kann lediglich angegeben werden welche Felder zurückgegeben werden können. Im Zuge dieser Arbeit wird nicht detaillierter auf die einzelnen Parameter eingegangen. Diese sind sehr verständlich dokumentiert. (5)

## Twitter

Twitter ist der bekannteste Kurznachrichtendienst, täglich werden über 200 Millionen Nachrichten, so genannte „Tweets“ erstellt. Nachrichten werden mittels Twitter in Echtzeit übertragen, Twitter zählt zu den größten Informationsnetzwerken der Welt. Tweets sind kurze meist prägnante Nachrichten, maximal 140 Zeichen lang und mit einer Mange zusätzlicher Metadaten versehen. Metadaten beinhalten unter anderem die Position von wo der Tweet stammt oder auch Informationen über die Verbreitung. Die Twitter API’s erlauben Zugriff auf diese enorme Menge an Daten.

Twitter bietet für unterschiedliche Aufgaben unterschiedlichste API’s so gibt es eine Vielzahl an Funktionen zum Verwalten eines Benutzers, wie das Erstellen von Nachrichten oder abfragen von Benutzerdaten. Diese API’s sind jedoch für die BI-Bereich von geringem Interesse. Aber auch für den BI-Bereich bietet Twitter nützliche API’s, vor Allen die Search API und die Streaming API. (6)

Alle API’s, exklusive der Streaming API, sind konventionelle Web Services die nach dem REST Paradigma abgefragt werden können. Ausschließlich alle API’s liefern Daten in Form von JSON.

### Tweet

Eventuell noch beschreiben wie ein Tweet aufgebaut ist!

### Search API

Mittels der Search API sprich der Such API ist es möglich auf Twitter nach speziellen Tweets zu suchen. So können zum Beispiel Tweets anhand von speziellen Schlüsselwörter gesucht werden oder anhand von Personen. Es ist möglich komplexe Abfragen mit bis zu 10 Schlüsselwörter zu bilden. Die Suchabfrage *worldcup 2014* liefert alle relevanten Tweets die entweder *worldcup* und *2014* enthalten jedoch nicht gezwungenermaßen als Phrase. Um die gesamte Phrase zu suchen muss diese unter Anführungszeichen gesetzt werden, sprich *“wolrdcup 2014“*. Um Tweets zu suchen die nur mindesten einen der 2 Schlüsselwörter beinhalten kann der Operator “OR“ verwendet werden. Um nach Personen zu suchen gibt es verschiedene Operatoren wie “from:“, “to:“, sowie “@“ mittels “@“ ist es möglich Tweets anhand von referenzierten Personen zu finden. Gefundene Tweets lassen sich nach Datum Sortieren dies gescheit mit den Operatoren “until:“ und “since:“. Zwei weitere sehr interessante Operatoren sind “:)“ und “:(“, mittels diesen Operatoren lassen sich Tweets anhand von positiven oder negativen Inhalt filtern.

Die genannten Operatoren und Parameter sind nur ein Auszug aus dem Funktionsumfang der API, mittels diesen lässt sich jedoch ein Großteil der Abfragen formulieren.

Zusätzlich zum Query lassen sich noch weitere Parameter angeben so ist es mittels “geocode“ möglich die Suche auf einen geografischen Bereich einzuschränken, dieser wird mittels Längen und Breitengraden angegeben sowie einem Radius. Weiteres lässt sich die Sprache der Tweets einschränken dies geschieht mittels dem Parameter “lang“.

Grundsätzlich ist diese API sehr mächtig, leider wird diese von Twitter stark eingeschränkt so ist es nur möglich Tweets der letzten Woche zu finden. Die Anzahl der Abfragen ist auch auf in etwa 180 pro 15 Minuten beschränkt was für produktive Dataminingsysteme relativ wenig ist. (6)

### Stream API

Die Stream API liefert in Echtzeit neue Tweets die gewissen Filtern entsprechen. Man kann sich diese API wie einen Wasserschlauch vorstellen, einmal verbunden spritzen alle neuen Tweets heraus. Im Vergleich zu einer normalen REST API bleibt die Verbindung auf unbeschränkte Zeit bestehen. Dadurch müssen Daten nicht zyklisch abgefragt werden es reicht eine einzige Abfrage. Abbildung 4 zeigt den Schematischen Aufbau dieser API. (6)

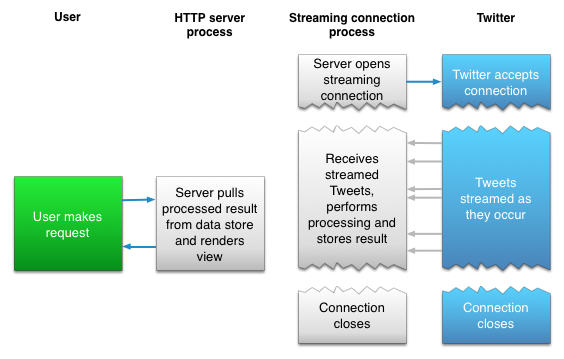


Abbildung 4 Aufbau der Streaming API

Twitter bietet verschiedene Endpunkte für die Streaming API, dabei gibt es drei Gruppen, die öffentlichen Streams, personenbezogene Streams sowie gruppenbezogene Streams. Für BI-Anwendungen sind eigentlich nur die öffentlichen Streams sinnvoll, diese beinhalten alle öffentlich verteilten Tweets, jedoch keine privaten Nachrichten. Auf Twitter werden primär Tweets öffentlich gepostet weshalb die Datenmenge dieser Streams enorm ist.

Es gibt drei öffentliche Endpunkte, der Endpunkt *„statuses/firehose“* liefert alle Tweets die irgendwo auf der Welt erstellt werden. Dieser Endpunkt ist für spezielle Anwendungen gedacht und benötigt eine exklusive Freischaltung Seites Twitter, es ist nicht möglich Filter zu übergeben es werden wirklich alle Tweets empfangen. (über 200 Millionen pro Tag) Ein weiterer Endpunkt ist *„statuses/sample“* dieser funktioniert wie *„statuses/firehose“* jedoch liefert dieser Endpunkt nur einen kleinen zufälligen Teil aller Tweets, dieser ist für Testzwecke gedacht.

Der wohl interessanteste Endpunkt ist *„statuses/filter“* dieser liefert einen Echtzeitstrom aller Tweets die gewisse Parameter erfüllen. Als Suchparameter können Benutzer ID’s, Schlüsselwörter sowie Kartenbereiche angegeben werden. Es muss mindesten ein Suchparameter verwendet werden. Einzelne Schlüsselwörter können mittels “UND“ sowie “ORDER“ verknüpft werden. Mit dem Standard Zugriffs Level lassen sich bis zu 400 Schlüsselwörter, 5000 Benutzer und 25 Kartenbereiche angeben. Die einzelnen Suchparameter werden immer mit “ODER“ verknüpft. Will man nur Tweets eines Landes unter Erfüllung spezieller Schlüsselwörter so muss man einen Stream starten der nur auf die Schlüsselwörter zieht und die Geografische Prüfung im nachhingen durchführen. Twitter bietet im Standard Zugriffs Level keine Möglichkeit die einzelnen Parameter mit “UND“ zu verknüpfen. (6)

## Historische Daten

Die großen sozialen Netzwerke bieten mächtige API’s. Diese sind jedoch Meistens auf die Abfrage von aktuellen Daten oft sogar auf Live Daten beschränkt. Oft werden jedoch historische Daten benötigt um Vergleiche anzustellen sowie um Trends zu berechnen. Bietet zwar Twitters Search API die Möglichkeit bis zu einer Woche in die Vergangenheit zu schauen reicht dieser Zeitraum in den Meisten Fällen nicht aus um aussagekräftige Analysen zu fällen.

Es ist zwar Möglich alle anfallenden Echtzeitdaten zu speichern und im Nachhinein selbst zu analysieren, doch benötigen diese Daten enormen Speicherplatz. Und will man zu einem Späteren Zeitpunkt andere Schlüsselwörter analysieren muss man erst wieder zu speichern beginnen.

Es gibt einige Anbieter historische Daten anbieten, diese stellen bezahlte API’s zur Verfügung. Der wohl bekannteste Anbieter ist *GNIP*, dieser bietet Daten einer Vielzahl an Plattformen zur Verfügung unter anderen Facebook und Twitter. GNIP liefert nicht nur die Rohdaten sondern ermöglich diese bereits zu Filtern und zu Analysieren. GNIP liefert nicht nur historische Daten mittels diesem System ist es möglich die größten sozialen Netzwerke zu verbinden und einen einzelnen Echtzeitstrom an Daten zu erhalten. Diese Funktion erleichtert das Datamining da nur noch eine Quelle benötigt wird.

Meiner Anfrage mir historische Daten für die Seminararbeit zur Verfügung zu stellen blieb leider unbeantwortet.

# Analyse der Daten

## Generelles

Im letzten Kapitel wurde das beschaffen der Daten beschreiben. Dieser enorme Haufen an Daten muss natürlich erst Analysiert werden um aussagekräftigt zu werden.

So lässt sich mittels Sentimentanalyse sprich der Stimmungserkennung erkennen ob ein gewisser Text „positiv“, „negativ“ oder „neutral“ geschrieben ist. Sentimentanalyse eignet sich hervorragend um Twitter Tweets auszuwerten, da diese kurz und prägnant formuliert sind. Ein großes Problem für die Sentimentanalyse ist die Ironie, moderne Algorithmen können diese schon relativ gut Filtern. Meistens wird jedoch einfach ein Korrekturfaktor verwenden sprich ein gewisser prozentteil der negativen Meinung ist Ironie, und somit zumindest als Neutral zu werten.

Sentimentanalyse Tools verwenden oft große Datenbanken an Phrasen und analysieren anhand dieser einen Text. Andere Ansätze verwenden neurale Netzwerke. Im Zuge dieser Arbeit wird nicht genauer auf die Funktionsweiße dieser Algorithmen eingegangen. Es soll viel mehr gezeigt werden wie mittels bestehender API’s Texte analysiert werden können. (7) (8)

Wie schon erwähnt eignen sich die API’s von Twitter ideal für die Sentimentanalyse. Mittels Twitter ist es möglich Meinungswechsel nahezu in Echtzeit zu erkennen und es kann sehr schnell gegengesteuert werden. Weitere API’S wie jene von Facebook eigenen sich auf andere weiße ebenso gut, so lässt sich mittels Facebook Open Graph schnell die Meinung über ein Produkt analysieren, indem alle Posts an der Produktseite analysiert werden.

## Analysetools und API’s

### DatumBox

Bietet 14 verschiedene „Machine Learning“ API’s an diese können kostenlos verwendet werden und erlauben bis zu 1000 Abfragen pro Tag. Zu den API’s zählen Sentimentanalyse, Sprachregelung, Geschlechtserkennung, Klassifizierung und noch viele weitere Funktionen. In Verbringung mit einem sozialen Netzwerk wie Twitter lassen sich mit Leichtigkeit Analysen durchführen. Die angebotenen Web Services können mittels einfacher REST Abfragen verwendet werden die Ergebnisse werden im JSON Format geliefert.

### Repustate

Repustate bietet ähnliche API’s wie DatumBox, zusätzlich ist es möglich online Daten bereits zu analysieren. Sozialmanager können mittels diesem System bereits einfache Analysen durchführen. Es können eigene Quellen definiert werden dabei wird Twitter und Facebook unterstützt.

### Brandwatch

Brandwatch ist einer der Modernsten Sozialmedia Überwachungs- und Analysetool. Es werden zwar auch API’s angeboten jedoch primär fungiert Brandwatch als Werkzeug für das soziale Marketing von großen Unternehmen. Neben aussagekräftigen Analysen lassen sich einfach übersichtliche Dashboards erstellen. Zu den Datenquellen zählen neben Facebook und Twitter so gut wie alle bekannten sozialen Netzwerke. Brandwatch verwendet nicht nur live Daten sondern auch historische Daten dadurch lassen sich dynamischere Analysen erstellen, wie vergleiche mit dem vor Jahr usw.

# Praxisbeispiel

## Generelles

## Twitter Stream

## Analyse mittels DatumBox

# Schlussbemerkung

Es gibt eine Vielzahl an sozialen API’s und es ist schwer den Überblick zu behalten. Es kommt sehr stark auf den geforderten Einsatz an und man kann Pauschal keine API bevorzugen. Grundsätzlich lassen sich jedoch die meisten einfachen Marketing aufgaben mittels Twitter oder Facebook bewerkstelligen. Der Sozial Media Auftritt sollte jedoch stets überlegt werden denn gefahren lauern an vielen Stellen.

Auf im Bereich der Auswertung und Analyse werden gute API’s Angebot einige sogar mit Einschränkung kostenfrei. Durch die Kombinierung dieser API’s lassen sich mit relativ einfachen Mitteln aussagekräftige Analysen oder Überwacungen erstellen, auch mit kleinem Budget.

Für große Unternehmen bieten sich „all in one“ Systeme wie Brandwatch an diese sind zwar sehr kostenintensiv ermöglichen jedoch eine flächendenkende Überwachung der sozialen Netzwerke.

Im Zuge der Arbeit konnte eine kleine Anwendung erstellt werden die mit einfachen Mitteln Twitter Daten Analysieren kann. Dadurch konnte praktisch gezeigt werden wie mächtig die heutzutage Angebotenen API’s sind. Die Möglichkeiten einer einzelnen API mag beschränkt sein jedoch durch die Verknüpfung unterschiedliche Systeme entstehen nahezu unbeschränkte Möglichkeiten.

# Quellen

1. **Programmable Web.** Api research. [Online] 2013. http://www.programmableweb.com/api-research.

2. **Google Inc.** Developers. *Google+ Plattform.* [Online] 2014. https://developers.google.com/+/.

3. **bitly inc.** bitly Developer. [Online] 2014. http://dev.bitly.com/index.html.

4. **Foursquare Inc.** Foursquare A. [Online] 2014. https://developer.foursquare.com/.

5. **Facebook Inc.** API Documentation. [Online] 2014. https://developers.facebook.com/docs/.

6. **Twitter Inc.** Developers Documentation. [Online] 2014. https://dev.twitter.com/docs.

7. **Liu, Bing.** *Sentiment Analysis and Opinion Mining.* 2012.

8. —. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data.* Berlin : Springer, 2017.

9. **Programmable Web.** Social API's. [Online] 2014. http://www.programmableweb.com/category/social/apis?category=20087.

# Weiterführende Literatur

* Techniques and applications for sentiment analysis  
  R Feldman - Communications of the ACM, 2013 - dl.acm.org
* Social network analysis: Methods and applications  
  S Wasserman – 1994
* Automatic Sentiment Analysis in On-line Text.  
  E Boiy, P Hens, K Deschacht, MF Moens - ELPUB, 2007 - law.kuleuven.be
* Twitter as a Corpus for Sentiment Analysis and Opinion Mining  
  A Pak, P Paroubek - LREC, 2010
* Twitter sentiment analysis: The good the bad and the omg!  
  E Kouloumpis, T Wilson, J Moore - ICWSM, 2011 - aaai.org
* Sentiment analysis and opinion mining  
  B Liu - Synthesis Lectures on Human Language Technologies, 2012 - morganclaypool.com
* A survey of opinion mining and sentiment analysis  
  B Liu, L Zhang - Mining Text Data, 2012 – Springer
* Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More  
  MA Russell - 2013

1. Web Services stellen Maschine to Maschine Systeme dar welche Daten über Netzwerke übertragen. [↑](#footnote-ref-1)
2. Simple Object Access Protocol [↑](#footnote-ref-2)
3. Extensible Markup Language [↑](#footnote-ref-3)
4. Representational State Transfer [↑](#footnote-ref-4)
5. Eurovision Song Contest [↑](#footnote-ref-5)