API 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

1.1 Motivation 1

1.2 Fragestellungen 2

1.3 Zielsetzung und Methodik 2

2 „soziale“ API 3

2.1 Funktionsbeschreibung und Entwicklung 3

2.2 Übersicht populärer API 7

2.3 Facebook 8

2.4 Twitter 10

2.5 Historische Daten 14

3 Analyse der Daten 15

3.1 Generelles 15

3.2 Analysetools und API 16

4 Praxisbeispiel 17

4.1 Generelles 17

4.2 Twitter Stream 17

4.3 Sentiment Analyse 17

4.4 Ergebnisse 18

5 Schlussbemerkung 20

6 Quellen 21

7 Weiterführende Literatur ..... **Fehler! Textmarke nicht definiert.**

1 EINFÜHRUNG

1.1 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. 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 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
- Evaluierung der einzelnen API
- Kombination mit der Thematik „Sentiment Analyse“

1.2 Fragestellungen

Folgende zentrale Fragestellungen sollen in der Seminararbeit beantwortet werden.

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

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

1.3 Zielsetzung und Methodik

Ziel dieser Arbeit ist, dem interessierten Leser einen Einstieg in die Verwendung von „sozialen“ API 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.

2 „SOZIALE“ API

2.1 Funktionsbeschreibung und Entwicklung

Dieses Kapitel soll einen Überblick der am stärksten verbreiteten API schaffen. Das Internet ist heutzutage eines der am meisten verwendeten Medien zum Verbreiten von Informationen. 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 Inhalten.

So gut wie alle sozialen Netzwerke, stellen leistungsstarke API zur Verfügung, wodurch solche „mashups“ überhaupt erst ermöglicht werden. Fast ausschließlich fungieren diese in Form von Web Services[1]. Die Anzahl der verfügbaren API sind in den letzten Jahren förmlich explodiert. So stieg die Anzahl der API von etwa 200 seit Anfang 2006 auf über 10000 im Jahr 2013. Dabei handelt es sich lediglich um auf „ProgrammableWeb“ registrierte API. *Siehe Quelle (1),* (2).

Abbildung 1 veranschaulicht das Wachstum der letzten Jahre.

[1] Web Services stellen Maschine to Maschine Systeme dar welche Daten über Netzwerke übertragen.

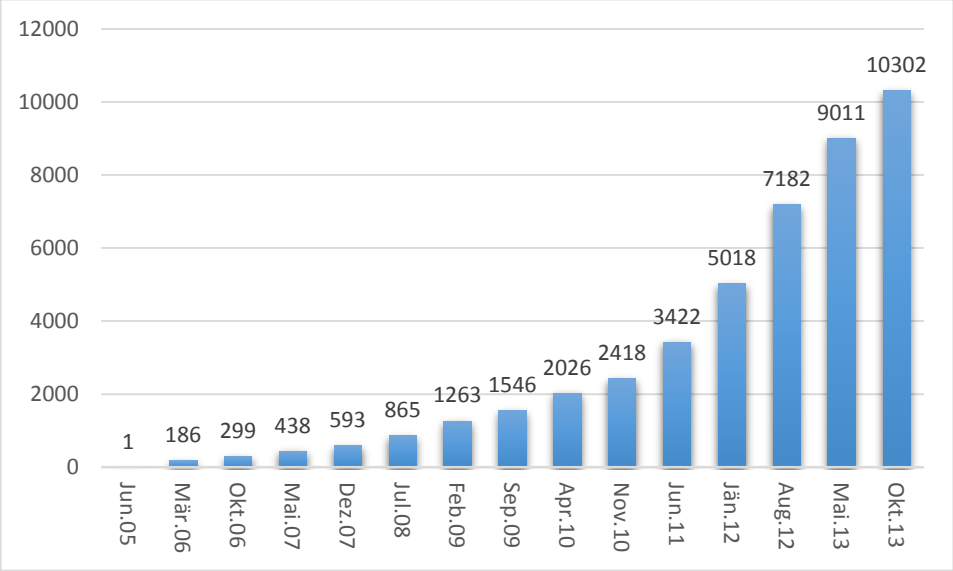

Abbildung 1 Wachstum Anzahl angebotene API auf ProgrammableWeb

Viele der frühen API verwendeten SOAP[2], ein Netzwerkprotokoll mit dessen Hilfe, Daten zwischen Systeme ausgetauscht werden können. Zur Repräsentation der Daten wird XML[3] verwendet.

Durch das enorme Wachstum von Webanwendungen in den letzten Jahren, wurden einfachere Protokolle gefordert, welche mittels einfacher Skriptsprachen wie z.B. JavaScript verwendet werden. SOAP war nie für Webanwendungen gedacht, dessen Anwendungsbereich liegt im Enterprise Bereich. Weshalb sich in den letzten Jahren das REST [4] Programmierparadigma durchgesetzt hat und mittlerweile von mehr als 70% aller API angeboten wird. Abbildung 2 zeigt die Entwicklung von REST in Vergleich zu SOAP in den letzten Jahren. *Siehe Quelle (1).*

REST beschreibt streng genommen kein Protokoll auch keine Norm. Es beschreibt lediglich die Methodik einer Web-URL genau einer serverseitigen Aktion zuzuordnen. Als Protokoll wird HTTP oder HTTPS verwendet unter Einbeziehung der folgenden HTTP Befehle *(HTTP-Verb).*

[2] Simple Object Access Protocol
[3] Extensible Markup Language
[4] Representational State Transfer

- GET, anfordern der angegeben Ressource (URL), 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 noch zusätzlichen Befehle wie *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 das JSON Format an. Viele moderne API wie, Facebook Open Graph bieten nur noch JSON an.

In Abbildung 3 wird ein einfaches Beispiel gezeigt. Mittels eines REST GET Request wird unter anderem, das Wetter von Salzburg abgefragt. Der Web Service antwortet wahlweise mit XML oder JSON. Es ist ersichtlich das JSON Daten leichter lesbar sind und auch übersichtlicher sind. Auf den ersten Blick würde man nicht glauben, dass die JSON Daten kleiner sind als die XML Repräsentationen. JSON braucht für die Darstellung vertikal zwar viel mehr Platz, ist jedoch deutlich kompakter.

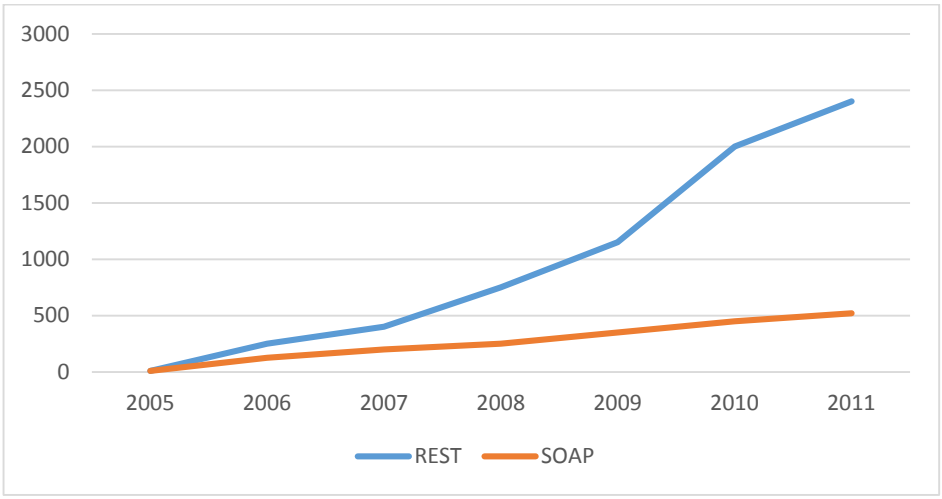

Abbildung 2 Vergleich REST zu SOAP

JSON (441 Byte)

```
{
  - coord: {
       lon: 13.04,
       lat: 47.8
    },
  - sys: {
       message: 0.0024,
       country: "AT",
       sunrise: 1402283361,
       sunset: 1402340697
    },
  - weather: [
      - {
           id: 801,
           main: "Clouds",
           description: "few clouds",
           icon: "02d"
        }
    ],
    base: "cmc stations",
  - main: {
       temp: 305.15,
       pressure: 1018,
       humidity: 40,
       temp_min: 305.15,
       temp_max: 305.15
    },
  - wind: {
       speed: 3.6,
       deg: 360,
       var_beg: 330,
       var_end: 40
    },
  - clouds: {
       all: 20
    },
    dt: 1402327200,
    id: 2766824,
    name: "Salzburg",
    cod: 200
}
```

XML (659 Byte)

```
<?xml version="1.0" encoding="utf-8"?>
<current>
  <city id="2766824" name="Salzburg">
    <coord lon="13.04" lat="47.8"/>
    <country>AT</country>
    <sun rise="2014-06-09T03:09:22" set="2014-06-09T19:04:58"/>
  </city>
  <temperature value="306.15" min="306.15" max="306.15" unit="kelvin"/>
  <humidity value="31" unit="%"/>
  <pressure value="1018" unit="hPa"/>
  <wind>
    <speed value="3.1" name="Light breeze"/>
    <direction value="360" code="" name=""/>
  </wind>
  <clouds value="20" name="few clouds"/>
  <precipitation mode="no"/>
  <weather number="801" value="few clouds" icon="02d"/>
  <lastupdate value="2014-06-09T14:50:00"/>
</current>
```

REST Request:
GET api.openweathermap.org
/data/2.5/weather?q=Salzburg&mode=[json / xml]

Abbildung 3 Beispiel für REST Request mit XML bzw. JSON Antwort, 441 Bytes JSON gegen 659 Bytes XML.

2.2 Übersicht populärer API

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 zu erstellen. Zusätzlich können Abfragen auch mittels Volltextsuche formuliert werden. Dadurch kann man sehr schnell Informationen zu einer Person oder Zielgruppe erhalten. Google verwendet vorzugsweiße JSON als Datenformat. *Siehe Quelle (2).*

bitly

Bitly zählt zu den am häufigsten verwendeten „URL Shortener“ 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 mit denen es möglich ist, die populärsten Links ausfindig zu machen. Gerade im „Business Intelligence“ Bereich ist die Verbreitung von Phrasen bzw. der Inhalt der Zielseiten sehr interessant. Als Beispiel zeigt bitly eine Echtzeitkarte, welche die Verteilung von Artikel der einzelnen Nachrichtenportale in den USA zeigt. *Siehe Quelle (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 höchst bedauerlich, da Benutzer die Links bereits mit Tags klassifizieren. Diese Daten hätten einen sehr hohen Mehrwert, für unter anderem dem Suchen und Finden von Referenzen oder Querverweisen.

Foursquare

Foursquare, der wohl bekannteste Geotagging Dienst, ist eine Weltkarte auf der Benutzer Plätze eintragen und auch bewerten können. In den letzten

Jahren ist Foursquare stark gewachsen und weist mittlerweile eine der größten Geodatenbanken auf. Mittels der angebotenen API ist es möglich anhand von Koordinaten oder Suchbegriffen, Plätze ausfindig zu machen. So ist es einer Anwendung möglich dem Benutzer Informationen aus seiner Umgebung anzuzeigen. *Siehe Quelle (4).*

2.3 Facebook

Facebook das bekannteste und größte soziale Netzwerk der Welt mit monatlich über 1,2 Billionen aktiven Nutzern, bietet eine Vielzahl an API.

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 Internetnutzer besteht darin möglichst viel Inhalt mit anderen zu teilen. So verwenden immer mehr Firmen Facebook und Co als mächtiges Marketingtool. Klickt 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 tun dies Negative noch viel rasanter. 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 vorherrschende Meinung kritisch hinterfragen. Es ist ratsam für diese Problematik einen Sozialmanager zu beauftragen, dessen Aufgabe primär in der Überwachung und weniger im eigentlichen „sozialem“ Marketing liegt.

Facebooks mächtigste API sind die Daten API. Diese können auch einem Sozialmanager in die Karten spielen, da mittels dieser API Daten strukturiert abgefragt werden können. Diese API 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, und steht abgekürzt für **F**acebook **Q**uery **L**anguage. Mittels dieser API ist es möglich, SQL ähnliche Queries auszuführen. Gegenüber den anderen API hat diese einen großen Vorteil, die Daten können bereits gefiltert werden und es sind sogar logische Verknüpfungen möglich. Diese API bietet nur die Möglichkeit *GET* Befehle zu verwenden *Delete* und *Post* ist nicht verfügbar. *Siehe Quelle (6).*

Folgender FQL Befehl liefert einige Felder des aktiven Users.

```
SELECT uid, name, pic_square FROM user WHERE uid = me()
```

Beispiel einer einfachen FQL Abfrage

JSON Antwort:

```
{
  "data": [
    {
      "uid": "1226612954",
      "name": "Julian Nischler",
      "pic_square": "https://scontent-b.xx.fbc .... 90_n.jpg"
    }
  ]
}
```

Beispiel einer Facebook JSON Antwort

Für den BI Bereich viel interessanter ist das Abfragen von Posts auf einer Seite. Ein solcher Befehl ist mittels FQL sehr schnell formuliert.

```
SELECT actor_id, message, created_time
FROM stream
WHERE source_id = 8419133006 and actor_id != 8419133006
```

```
limit 100
```

Beispiel einer erweiterten FQL Abfrage

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 wird wieder im JSON Format geliefert und könnte nun mittels geeigneten BI Tools ausgewertet werden.

Graph API

Die Graph API stellt die de facto Standard API von Facebook dar 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, auch Daten zu schreiben, oder zu löschen. Befehle werden nicht in Form eines „Query“ gesendet, sondern als normale URL. *Siehe Quelle (6).*

Zur Veranschaulichung folgt ein Befehl, welcher wieder alle Posts auf der Seite des ESC[5] abfragt.

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

Beispiel einer Facebook Graph Abfrage

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, da diese sehr verständlich dokumentiert sind. *Siehe Quelle (5).*

2.4 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 Menge zusätzlicher Metadaten versehen. Metadaten beinhalten unter anderem die Position von

[5] Eurovision Song Contest

wo der Tweet stammt, oder auch Informationen über die Verbreitung. Die Twitter API erlauben Zugriff auf diese enorme Menge an Daten. *Siehe Quelle (7),* (2).

Twitter bietet für unterschiedliche Aufgaben unterschiedlichste API. So gibt es eine Vielzahl an Funktionen zum Verwalten eines Benutzers, wie das Erstellen von Nachrichten oder abfragen von Benutzerdaten. Diese API sind jedoch für die BI-Bereich von geringem Interesse. Für den BI-Bereich bietet Twitter nützliche API, vor allen die Search API und die Streaming API. *Siehe Quelle (6).*

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

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 *worldcup* und *2014* enthalten, jedoch nicht gezwungenermaßen als Phrase. Um die gesamte Phrase zu suchen, muss diese unter Anführungszeichen gesetzt werden, sprich *"worldcup 2014"*. Um Tweets zu suchen, die nur mindesten eine der 2 Schlüsselwörter beinhalten, kann der Operator "OR" verwendet werden. Um nach Personen zu suchen, gibt es verschiedene Operatoren wie "from:", "to:", sowie "@", so ist es möglich Tweets anhand von referenzierten Personen zu finden. Gefundene Tweets lassen sich nach Datum sortieren. Das geschieht mit den Operatoren "until:" und "since:". Zwei weitere sehr interessante Operatoren sind ":)" und ":(", mittels dieser 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 dieser, 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 mit Längen und Breitengraden angegeben, sowie einem Radius. Weiters 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. *Siehe Quelle (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, sondern es reicht eine einzige Abfrage.

Abbildung 4 zeigt den schematischen Aufbau dieser API. *Siehe Quelle (6).*

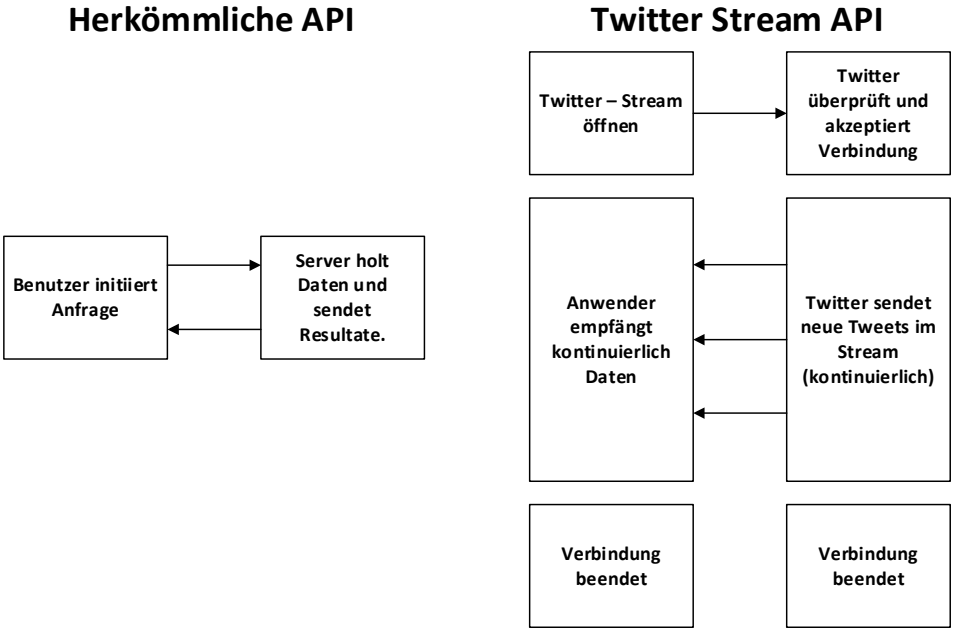

Abbildung 4 Aufbau der Streaming API

Twitter bietet verschiedene Endpunkte für die Streaming API. Dabei gibt es drei Gruppen: die öffentlichen Streams, die personenbezogenen Streams sowie die gruppenbezogenen 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 seitens 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. Er ist nur für Testzwecke gedacht.

Der wohl interessanteste Endpunkt ist *„statuses/filter“*, er liefert einen Echtzeitstrom aller Tweets die gewisse Suchparameter erfüllen. Als Suchparameter können Benutzer IDs, Schlüsselwörter sowie Kartenbereiche angegeben werden. Es muss mindesten ein Suchparameter verwendet werden. Einzelne Schlüsselwörter können mittels “UND“ sowie “ORDER“ verknüpft werden. Mit dem Standard Berechtigung 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 erhalten, so muss man einen Stream starten welcher nur auf die Schlüsselwörter zielt und die geografische Prüfung im Nachhinein durchführen. Twitter bietet mit der Standard Berechtigung, keine Möglichkeit die einzelnen Parameter mit “UND“ zu verknüpfen. *Siehe Quelle (6).*

2.5 Historische Daten

Die großen sozialen Netzwerke bieten mächtige API. Diese sind jedoch meistens auf die Abfrage von aktuellen Daten, oft sogar nur auf Live Daten beschränkt. Oft werden jedoch historische Daten benötigt, um Vergleiche anzustellen, sowie um Trends zu berechnen. Es bietet zwar Twitters Search API die Möglichkeit bis zu einer Woche in die Vergangenheit zu schauen. In den meisten Fällen reicht dieser Zeitraum nicht aus, um aussagekräftige Analysen zu ermöglichen.

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

Es gibt einige Anbieter die historische Daten anbieten, diese stellen dann kostenpflichtige API zur Verfügung. Der wohl bekannteste Anbieter ist *GNIP*, dieser bietet Daten von einer Vielzahl an Plattformen, unter anderen Facebook und Twitter. Die von GNIP bereitgestellten API liefern nicht nur die Rohdaten, sondern ermöglicht diese zu filtern und zu analysieren. Mittels dieses Systems, 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.

3 Analyse der Daten

3.1 Generelles

Im vorangegangenen Kapitel werden Möglichkeiten für das Beschaffen von Daten beschrieben. Diese enorme Mange an Daten muss natürlich erst analysiert werden, um aussagekräftig 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 ironische Phrasen schon relativ gut filtern. Meistens werden jedoch einfach Korrekturfaktoren verwendet, sprich ein gewisser Prozentteil der negativen Meinung wird als Ironie abgestempelt und somit als neutral gewertet. *Siehe Quelle (8).*

Sentimentanalyse Tools, verwenden oft große Datenbanken an Phrasen und analysieren anhand dieser einen Text. Andere Tools basieren auf neuronale Netzwerke, welche zuvor auf diese Problematik trainiert wurden. Im Zuge dieser Arbeit wird nicht genauer auf die Funktionsweise dieser Algorithmen eingegangen. Es soll viel mehr gezeigt werden, wie mittels bestehender API Texte analysiert werden können.

Die API von Twitter eignen sich ideal für die Sentimentanalyse. Mittels Twitter ist es möglich, Meinungswechsel nahezu in Echtzeit zu erkennen. Dadurch kann sehr schnell gegengesteuert werden. Weitere API, wie jene von Facebook, eignen sich auf andere Weise ebenso gut. So lässt sich mittels Facebook Open Graph, schnell die Meinung über ein Produkt

analysieren, indem alle Posts auf der jeweiligen Produktseite analysiert werden.

3.2 Analysetools und API

DatumBox

DatumBox bietet 14 verschiedene „Machine Learning“ API an. Diese können kostenlos verwendet werden und erlauben bis zu 1000 Abfragen pro Tag. Zu den API zählen, Sentimentanalyse, Sprachregelung, Geschlechtserkennung, Klassifizierung und noch viele weitere Funktionen. In Verbindung mit einem sozialen Netzwerk wie Twitter, lassen 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, wie die DatumBox. Zusätzlich ist es möglich, Daten bereits online zu analysieren. Sozialmanager können mittels dieses Systems bereits einfache Analysen durchführen. Es können eigene Quellen definiert werden, dabei werden Netzwerke wie Twitter und Facebook unterstützt.

Brandwatch

Brandwatch ist einer der modernsten Sozialmedia Überwachungs- und Analysetools. Es werden zwar auch API angeboten, jedoch fungiert Brandwatch primär 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 dynamische Analysen erstellen, wie Vergleiche mit dem Vorjahr usw.

4 PRAXISBEISPIEL

4.1 Generelles

Im Zuge dieser Arbeit wurde ein kleines Tool entwickelt, welches die Möglichkeiten einiger API praktisch veranschaulicht. Dieses Tool empfängt in Echtzeit Twitter Stream Daten und stellt diese mittels Diagrammen dar. Zusätzlich werden zur Analyse der Daten verschiedene API, wie jene von DatumBox oder uclassify verwendet. Dadurch kann ein Trend bzw. eine Quote der Tweets berechnet werden. Alle empfangenen Daten werden nur im Arbeitsspeicher gespeichert, sprich sind flüchtig. Um die Daten persistent speichern zu können, müsste die Anwendung um eine Datenbank erweitert werden.

4.2 Twitter Stream

Wie in Kapitel 2.4 beschrieben, bietet Twitter eine Streaming API. Mittels dieser können anhand von Suchparametern Tweets in Echtzeit empfangen werden. Aus den empfangenen Tweets werden der Text, die Sprache und das Land extrahiert und in der Oberfläche mittels Diagrammen dargestellt.

4.3 Sentiment Analyse

Die Anzahl der Tweets ist je nach Schlüsselwort enorm und es werden weit über 1000 Tweets pro Minute empfangen. Die Analyse der Daten braucht viel länger und hinkt den Tweets immer hinterher. Um diese Problematik zu umgehen, arbeiten im Hintergrund mehrere Dienste, welche parallel die Daten analysieren. In der Oberfläche ist ersichtlich wie viel Prozent der Tweets bereits analysiert wurden. Nach erfolgreicher Analyse eines Tweets, werden die Statistiken neu errechnet und die Diagramme neu

gezeichnet. So entsteht kontinuierlich ein genaueres Abbild der aktuellen Meinung.

4.4 Ergebnisse

Die Anwendung funktionierte nach einer Entwicklungszeit von etwa 12h erstaunlich gut. Es handelt sich um keine finale Software, viel mehr um eine Machbarkeitsstudie. Das größte Problem verursachte die enorme Mengen an Tweets. Alle getesteten API zur Analyse, hatten Probleme mitzuhalten. Paralleles Abfragen der API löste das Problem ansatzweise. Mit den kostenfreien API ist die Anzahl der Anfragen stark limitiert. Durch die enorme Menge an Tweets, welche von Twitter geliefert werden, stößt man sehr schnell an die Limits und erhält folgende Fehlermeldung.

```
„You have reached your daily free request limit of 1000 request.“
```

Grundsätzlich könnte diese Software produktiv verwendet werden, jedoch sollte eine bezahlte API zur Analyse herangezogen werden.

Abbildung 5 sowie Abbildung 6 zeigt die Anwendung während einer Analyse. Dabei wurden die Schlüsselwörter „worldcup“ sowie „iraq“ analysiert. Gut ersichtlich ist, dass die Tweets bezüglich „iraq“ viel negativer sind als jene des Schlüsselworts „worldcup“.

```
Der Quellcode der Anwendung liegt der Arbeit als Anhang bei und kann
lizenzfrei verwendet werden.
```

Da mittels den kostenlosen online Sentiment Analyse Diensten, nur mäßige Ergebnisse erzielt werden konnten, wurde die Software mit einer Open Source Analyse Bibliothek erweitert. Mittels dieser Lösung konnte die Analyse enorm beschleunigt werden. Leider ist diese Bibliothek nicht auf Tweets trainiert so dass sehr viele Tweets als neutral eingestuft werden. Die Bibliothek wird von der Standort Universität entwickelt und ist unter *http://nlp.stanford.edu/sentiment/* verfügbar.

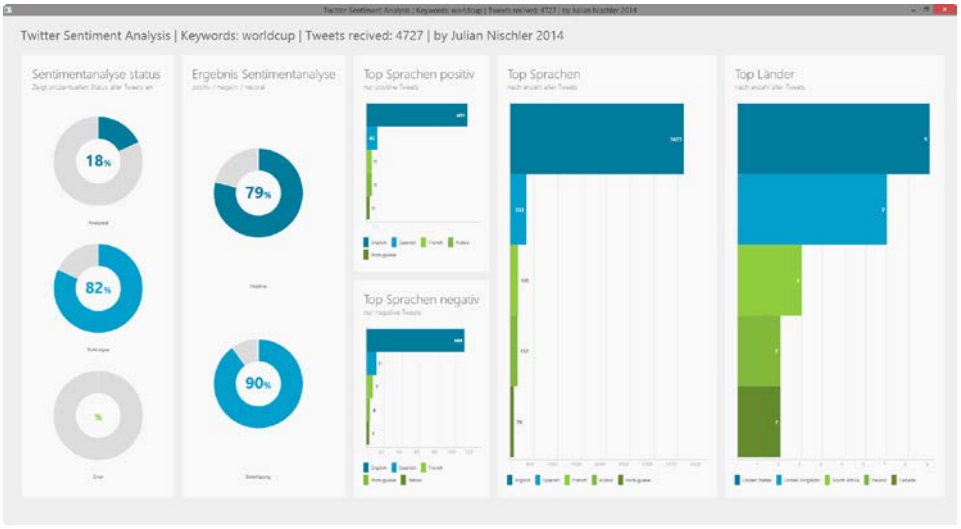

Abbildung 5 Analyse Schlüsselwort "worldcup" am 12.06.14 nach einer Laufzeit von ca 3 Minuten

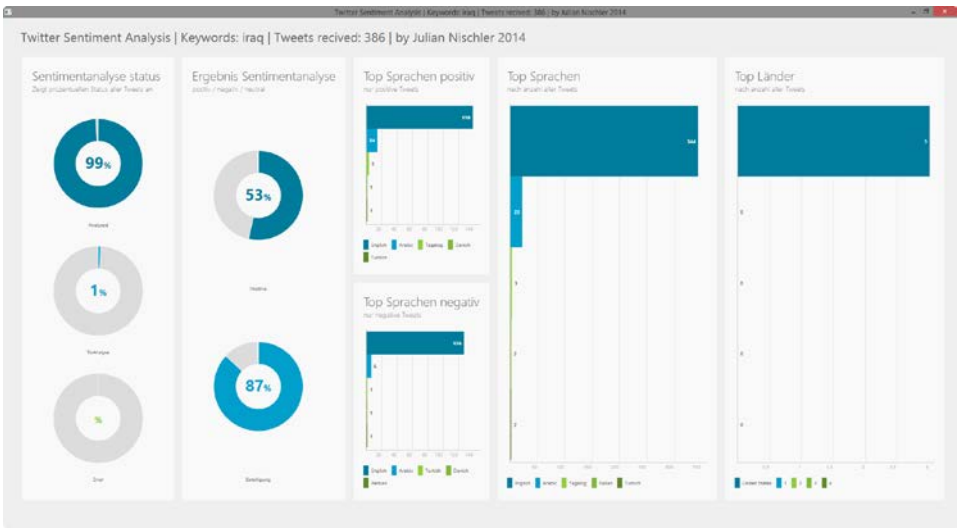

Abbildung 6 Analyse Schlüsselwort "iraq" am 12.06.14 nach einer Laufzeit von ca. 5 Minuten

5 SCHLUSSBEMERKUNG

Es gibt eine Vielzahl an sozialen API 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 Marketingaufgaben mittels Twitter oder Facebook bewerkstelligen. Der „Sozial Media“ Auftritt sollte jedoch stets überlegt werden, den Gefahren lauern an vielen Stellen.

Auch im Bereich der Auswertung und Analyse von Daten, werden gute API geboten, einige sogar mit Einschränkungen kostenfrei. Durch kombinieren dieser API lassen sich mit relativ einfachen Mitteln aussagekräftige Analysen oder Überwachungen erstellen.

Auch mit kleinem Budget ist das möglich!

Für große Unternehmen bieten sich die „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 sind. Die Möglichkeiten einer einzelnen API mögen beschränkt sein, jedoch durch die Verknüpfung unterschiedlicher Systeme entstehen nahezu unbeschränkte Möglichkeiten.

Es konnte ein Überblick über verfügbare API geschaffen werden sowie dessen Verwendung und Zweck. Im Bereich der Sentiment Analyse konnte ebenfalls eine für weitere Anwendungen / Arbeiten verwendbare Basis geschaffen werden.

6 LITERATUR VERZEICHNIS

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

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

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

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

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

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

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