

Assoziationsbasierte Trenderkennung auf Microblogdaten

Text

Master-Thesis von Klaus Wilhelmi

Tag der Einreichung:

1. Gutachten: Prof. Johannes Fürnkranz
2. Gutachten: Dr. Frederik Janssen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik
Knowledge Engineering Group

Assoziationsbasierte Trenderkennung auf Microblogdaten

Vorgelegte Master-Thesis von Klaus Wilhelmi

1. Gutachten: Prof. Johannes Fürnkranz
2. Gutachten: Dr. Frederik Janssen

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 31. Mai 2015

(Klaus Wilhelmi)

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 5 |
| 1.1 | Motivation | 5 |
| 1.2 | Ziel | 5 |
| 1.3 | Aufbau der Arbeit | 6 |
| 2 | Grundlagen | 7 |
| 2.1 | Twitter | 7 |
| 2.2 | Data Mining | 8 |
| 2.2.1 | Selektion | 9 |
| 2.2.2 | Preprocessing | 9 |
| 2.2.3 | Transformation | 9 |
| 2.2.4 | Mustererkennung | 9 |
| 2.2.5 | Interpretation & Evaluierung | 10 |
| 2.3 | Locality-Sensitive Hashing | 10 |
| 2.3.1 | Simhash | 10 |
| 2.3.2 | Finden von Kandidatenpaaren | 11 |
| 2.4 | Assoziationsanalyse | 12 |
| 2.4.1 | Frequent Itemsets | 13 |
| 2.4.2 | Assoziationsregeln | 14 |
| 2.4.3 | FP-Growth Algorithmus | 15 |
| 2.5 | Klassifizierung | 17 |
| 2.5.1 | Naive Bayes | 18 |
| 3 | Die Streaming API | 21 |
| 3.1 | Inhalt der gesammelten Daten | 21 |
| 3.2 | Datenbestand | 22 |
| 3.3 | Preprocessing | 23 |
| 3.3.1 | Filterung anhand einer Blacklist | 24 |
| 3.3.2 | Entfernen von Sonderzeichen | 24 |
| 3.3.3 | Korrektur von Umgangssprache | 24 |
| 3.3.4 | Entfernen von Stopwords | 25 |
| 3.3.5 | Twitter-spezifische Korrekturen | 25 |
| 3.3.6 | Handhabung langer Wörter | 26 |
| 3.3.7 | Säuberung von sonstigem Spam | 26 |
| 4 | Erkennen von Trends | 27 |
| 4.1 | Was ist ein Trend? | 27 |
| 4.2 | Häufigkeitsanalyse anhand von Keywords | 28 |
| 4.3 | Vergleich von Assoziationen von Keywords | 29 |
| 4.4 | Frequent Pattern Mining Ansatz | 29 |
| 4.5 | Online Clustering Ansatz | 32 |
| 4.6 | Personalisierung von Trends | 35 |

| | | |
|----------|---|-----------|
| 5 | Implementiertes System | 36 |
| 5.1 | Allgemeine Architektur | 36 |
| 5.2 | Preprocessing Pipeline | 37 |
| 5.3 | Frequent Pattern Mining Ansatz | 38 |
| 5.3.1 | Übersicht | 38 |
| 5.3.2 | Detailansicht | 40 |
| 5.3.3 | Klassifizierung | 42 |
| 5.4 | Online Clustering Ansatz | 42 |
| 5.4.1 | Übersicht | 43 |
| 5.4.2 | Detailansicht | 43 |
| 5.4.3 | Klassifizierung | 45 |
| 5.5 | TweetGrabber | 45 |
| 6 | Evaluierung | 46 |
| 6.1 | Evaluierung der Trenderkennung | 47 |
| 6.1.1 | Frequent Pattern Mining Ansatz | 47 |
| 6.1.2 | Online Clustering Ansatz | 49 |
| 6.1.3 | Zusammenfassung | 49 |
| 6.2 | Evaluierung der Personalisierungsfunktion | 50 |
| 7 | Aktuelle Forschung | 52 |
| 7.1 | Document-Pivot Ansätze | 52 |
| 7.2 | Feature-Pivot Ansätze | 55 |
| 7.3 | Sonstige Forschung | 56 |
| 7.4 | Zusammenfassung der aktuellen Forschung | 58 |
| 8 | Zusammenfassung | 59 |
| 8.1 | Ausblick | 60 |
| A | Twitter-spezifische Stopwordliste | 61 |
| B | Häufigkeitsanalyse Keywords | 62 |
| C | Häufigkeitsanalyse n-Tupel | 63 |
| D | Skript zur Ermittlung der häufigen Paare | 64 |
| E | Frequent und maximal Frequent Itemsets | 67 |
| F | Ergebnisse der Benutzerevaluierung | 68 |

Zusammenfassung

Daten von Twitter zeichnen sich dadurch aus, dass sie sehr vielfältige Informationen enthalten. Tweets können sowohl von Themen aus der echten Welt motiviert sein als auch Werbung oder Spam beinhalten. Es gilt, die interessanten und aktuellen Informationen von den nicht relevanten zu trennen. Der in dieser Arbeit beschriebene Ansatz nutzt die Assoziationsanalyse kombiniert mit einer Repräsentation von Trends innerhalb eines Graphen. Das entworfene System ist damit in der Lage Themenfragmentierungen im Gegensatz zu bisher beschriebenen Ansätzen aus der Forschung deutlich zu verringern und Trends zuverlässiger zu erkennen.

Abstract

Data from Twitter is characterized by a high diversity of information. Tweets may contain topics inspired by the real world or their purpose is mainly advertisement or spam. The question is how to separate current and interesting information from non-relevant data. The approach described in this thesis uses Association Rule Learning combined with a graph-based representation of the detected trends. Based on this the developed system is able to reduce fragmentation of topics and detect trends more reliable compared to approaches that solely rely on use Association Rule Learning.

1 Einleitung

1.1 Motivation

Microblog Dienste und im Speziellen Twitter sind Medien, die intensiv genutzt werden, um aktuelle Themen aller Art mit anderen Menschen zu teilen und zu diskutieren. Die Einführung des Smartphones hat die Möglichkeit eröffnet, nahezu zu jeder Zeit und an jedem Ort „online“ zu gehen und miteinander zu kommunizieren. Dies hat Diensten wie Twitter zu ihrer Popularität verholfen. Durch mobile Geräte wie Smartphones ist es möglich, kurze Texte, Bilder und Videos über aktuelle Geschehnisse dem Rest der Welt in Echtzeit zur Verfügung zu stellen.

Auswertungen zeigen, dass gerade Twitter in anderen Ländern, im Vergleich zu Deutschland, deutlich intensiver genutzt wird. [36, 40] Während des arabischen Frühlings wurde Twitter in den betroffenen Ländern ausgiebig genutzt, um Bilder und Videos über die aktuelle Lage zu veröffentlichen. Es fällt auf, dass in Ländern, in denen die Presse- und Meinungsfreiheit eingeschränkt wird, die sozialen Medien und besonders Twitter von den Menschen als Werkzeug eingesetzt werden, um ihre Meinung zu äußern. [26] In solchen Fällen sind soziale Medien eine interessante Quelle, um auf aktuelle Entwicklungen aufmerksam zu werden oder sich über die Lage in einem Land zu informieren, in dem es keine objektive Berichterstattung durch die Medien gibt.

Twitter kann als ein nicht endender Strom von thematisch sehr unterschiedlichen Nachrichten gesehen werden. Selbst die Filterung nach bestimmten zu einem Thema zugehörigen Schlüsselwörtern kann zu vielen themenfremden Nachrichten führen. So werden oft Schlüsselwörter, die zu einem Zeitpunkt gerade beliebt sind in Nachrichten integriert, die thematisch einen anderen Hintergrund haben, um ein größeres Publikum zu erreichen.

Trends sind Themen über die viele Menschen gemeinsam diskutieren und die eine breite Öffentlichkeit bewegen. Dies ist der Grund, warum es von Interesse ist, diese aus dem Strom von Twitternachrichten zu extrahieren. Trends sind sehr vielfältig und können beispielsweise durch Naturkatastrophen, Terroranschläge, Todesfälle von berühmten Persönlichkeiten oder Sportereignisse ausgelöst werden.

Die Schwierigkeit bei der Erkennung von Trends liegt darin, in der Flut von Nachrichten, relevante Elemente zu erkennen. Gleichzeitig muss man sich auch bewusst machen, dass Informationen, die über soziale Medien verbreitet werden, nicht zwangsläufig korrekt sind. Wenn also ein Thema von vielen Leuten diskutiert wird, bedeutet das nicht, dass man sich blind auf diese Information verlassen kann. Ein Beispiel hierfür lieferte ein Schüler in England, der sich als Sportjournalist ausgab. Er schaffte es, mit selbst ausgedachten Meldungen bekannte Medien auf sich aufmerksam zu machen, welche dann seine Meldungen veröffentlichten und ihn als Quelle angaben. [29]

1.2 Ziel

Ziel dieser Arbeit ist es, das Thema der Erkennung und Verfolgung von Trends im Bereich von Microblogs zu beleuchten. Im Speziellen wird hier der Dienst Twitter betrachtet, da dessen Daten frei verfügbar sind und Entwickler und Forscher leicht darauf zugreifen können. Zu Beginn soll die Fragestellung betrachtet werden, was man unter einem Trend versteht und wie man diesen definieren kann. Außerdem soll ein System erstellt werden, das Daten von Twitter möglichst in Echtzeit verarbeitet und auswertet. Relevante Trends sollen entsprechend aufgearbeitet und für den Anwender verständlich dargestellt werden. Zur Einordnung soll ein Überblick über die aktuelle Forschung gegeben werden.

Das zukünftige System soll auf dem allgemeinen Meldungsstrom von Twitter arbeiten. Es soll keine thematische Eingrenzung stattfinden, indem der Datenstrom beispielsweise durch bestimmte Schlüsselwörter vorgefiltert wird. Dies liefert den Vorteil, dass das zukünftige System out-of-the-box einsatzfähig ist, ohne dass Benutzereingaben notwendig sind oder konkrete Themen vorgegeben werden müssen. Dieser Ansatz birgt aber auch eine Reihe von Gefahren und Schwierigkeiten, welche im späteren Verlauf der Arbeit erläutert werden. Bei einer großen Anzahl an Trends muss eine Eingrenzung vorgenommen werden. Dies kann zum Beispiel anhand der Häufigkeit des Vorkommens oder deren Veränderungen geschehen. Dadurch kann das Risiko entstehen, dass ein Trend der sich nur schwach in den Daten widerspiegelt nicht erkannt wird, da er von den Standardthemen überlagert wird.

1.3 Aufbau der Arbeit

Der Aufbau der Arbeit gestaltet sich folgendermaßen: Im zweiten Kapitel werden die technischen und theoretischen Grundlagen erläutert, die für das Verständnis der Arbeit relevant sind. Im Anschluss werden die Erfahrungen und Erkenntnisse erörtert, die im Bearbeitungszeitraum gewonnen wurden. Aus den Erkenntnissen wurden zwei konkrete Ansätze implementiert und um eigene Ideen ergänzt. Die gewonnenen Ergebnisse werden in Kapitel 3 und 4 erläutert. Nach der Beschreibung des implementierten Systems folgt ein Kapitel über dessen Evaluierung. In Kapitel 7 wird ein Überblick über die aktuelle Forschung gegeben. Der letzte Abschnitt der Arbeit enthält eine Zusammenfassung und einen Ausblick über weitere Ideen für zukünftiges Vorgehen.

2 Grundlagen

Im folgenden Abschnitt werden alle technischen und theoretischen Hintergründe erläutert, die für die Ausführungen in den nächsten Kapiteln relevant sind. Als Erstes wird daher auf den Dienst Twitter und seine APIs eingegangen. Es wird eine kurze Einführung in den Bereich Data Mining gegeben und die in einem Data Mining Prozess relevanten Verarbeitungsschritte. Für das Verständnis der in der Arbeit selbst entwickelten Ansätze werden die Themen Locality-Sensitive Hashing, Assoziationsanalyse und Naive Bayes Text Klassifizierung näher erläutert.

2.1 Twitter

Bei Twitter handelt es sich um einen Microblogging-Dienst, der es seinen Benutzern ermöglicht, kurze Nachrichten zu veröffentlichen. Diese Nachrichten, genannt Tweets, dürfen eine maximale Länge von 140 Zeichen haben. Mit ca. 500 Millionen Tweets am Tag¹ ist der Dienst Marktführer im Bereich Microblogging und ebenfalls einer der Big Player im Bereich Social Media. Abbildung 2.1 zeigt einen exemplarischen Tweet, der in Zusammenhang mit der Landung der Raumsonde Rossetta² veröffentlicht wurde.

Im Laufe der Zeit wurden bestimmte Konventionen und Schreibweisen für das Erstellen von Tweets entwickelt. Um Nachrichten einem bestimmten Themenkomplex oder einer speziellen Diskussion zuzuordnen, können sogenannte *Hashtags* verwendet werden. Hierzu wird eine Raute gefolgt von einem Begriff der den Themenkomplex beschreibt in den Tweet eingebettet (Beispiele: #politik, #bundesliga, #CometLanding).

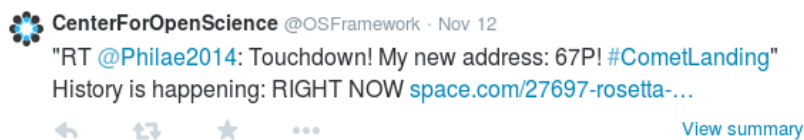


Abbildung 2.1.: Beispiel eines Tweets

Es besteht die Möglichkeit, andere Benutzer von Twitter innerhalb einer Nachricht zu referenzieren. Entweder, weil der Inhalt des Tweets in Zusammenhang mit dem jeweiligen Nutzer steht oder weil man einen anderen Nutzer auf einen Tweet aufmerksam machen möchte. Hierzu wird das @-Zeichen gefolgt vom Benutzernamen an einer Stelle des Tweets eingebettet (Beispiel: „Bundeskanzlerin @AngelaMerkel besucht Italien“).

Um auf Tweets anderer Nutzer Bezug zu nehmen, existieren zwei Möglichkeiten. Eine öffentliche Antwort, genannt *Reply*, beginnt mit einem @-Zeichen, gefolgt vom Benutzernamen auf dessen Tweet man antworten möchte. Die zweite Möglichkeit sind sogenannte Retweets. Hier beginnt ein Tweet wie in Abbildung 2.1 zu sehen ist, mit „RT @Username:“, gefolgt vom ursprünglichen Tweet. Retweets werden eingesetzt, um interessante Inhalte im Netzwerk von Twitter zu verteilen und andere Benutzer auf ein Thema aufmerksam zu machen. Wird ein Tweet besonders oft „retweetet“ kann das darauf hindeuten, dass er ein Thema behandelt, das für eine große Benutzergruppe relevant ist. Wird ein Retweet erstellt, ist dieser sichtbar für alle Benutzer, die dem Ersteller folgen.

¹ <https://about.twitter.com/company>

² <http://rosetta.esa.int>

Benutzer können innerhalb von Twitter miteinander in Beziehung stehen, in dem sie sich gegenseitig folgen. Diese Beziehungen können als gerichteter Graph gesehen werden. Folgt Benutzer A Benutzer B, so erhält A alle Tweets die B verfasst. A wird als „Follower“ von B bezeichnet. Nutzer B hingegen erhält auf seiner Timeline keine Tweets von A.

Twitter bietet mehrere Schnittstellen an, über die Benutzer Daten beziehen können. Für Auswertungen, die das aktuelle Vorgehen auf Twitter untersuchen, bieten sich die Streaming APIs an. Durch sie erhält ein Entwickler Tweets, die aktuell veröffentlicht werden. Auf Daten aus der Vergangenheit kann nicht zugegriffen werden. Twitter bietet allen Entwicklern einen kostenlosen Zugang über den bestenfalls 1% der gesamten Tweets ausgelesen werden können. Die sogenannte Firehose bietet die Möglichkeit auf 100% aller Tweets in Echtzeit zuzugreifen. Ein Zugang ist sehr kostenintensiv und auch nur einer bestimmten Anzahl von Data Resellern vorbehalten. Die Unterschiede der zwei Schnittstellen sind auch Gegenstand der aktuellen Forschung. Morstatter et al. [37] haben untersucht, wie sich die reduzierte Datenmenge des kostenlosen Zugangs gegenüber dem uneingeschränkten Zugriff durch die Firehose Schnittstelle auf Forschungsergebnisse auswirkt. Hierauf wird gesondert in Kapitel 3 eingegangen.

Die REST-basierten APIs bieten zahlreiche Abfragemöglichkeiten. Im Gegensatz zu den Streaming APIs lassen sich auch bestehende Daten manipulieren und neue Daten an Twitter übermitteln. Für den Zugriff existiert eine Begrenzung, die die Häufigkeit der Zugriffe reguliert. Pro Endpoint dürfen 180 Requests per Zeitfenster durchgeführt werden. Ein Fenster besteht jeweils aus 15 Minuten. Rückwirkend kann auf die Tweets der letzten sechs bis neun Tage zugegriffen werden.

2.2 Data Mining

Der Begriff Data Mining beschreibt das Finden und Erkennen von Mustern in Daten. [31] Anders gesagt, wie kann man aus einer großen Menge von unstrukturierten Daten, in angemessener Zeit, sinnvolle Informationen extrahieren. Unstrukturiert bedeutet in diesem Fall, dass Daten so wie sie vorliegen nicht zwangsläufig verarbeitet werden können. Gegebenenfalls ist einige Arbeit im Vorfeld notwendig, um sie in eine brauchbare Form zu bringen.

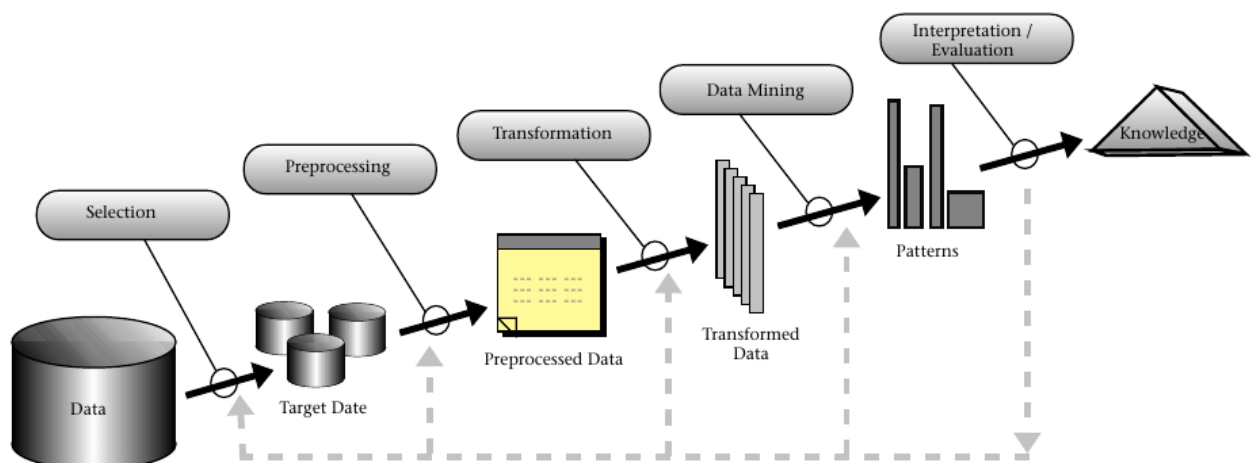


Abbildung 2.2.: Knowledge Discovery in Databases (KDD) [15]

Das folgende Beispiel, welches Data Mining kurz beschreibt, stammt aus *A Programmer's Guide to Data Mining* [58]: Der Mensch ist sehr gut darin in kleinem Umfang mentale Modelle zu erstellen und daraus Muster abzuleiten. Ein mentales Modell kann zum Beispiel das Bild im eigenen Kopf sein, das man über die Vorlieben des eigenen Partners bezüglich der Vorlieben bei Kinofilmen hat. Durch die enge Bindung zum Partner und die damit gesammelte Erfahrung lassen sich manche Dinge leicht einschätzen. Mag der Partner zum Beispiel keine gewalttätigen Filme, so kommen bei der Planung eines Kinobesuchs beispielsweise Horrorfilme nicht in Frage. Data Mining erlaubt diesen Prozess durch Methoden und Algorithmen auf große Mengen von Daten anzuwenden. So lassen sich durch Machine Learning Algorithmen Lieblingsserien vorhersagen oder Produktvorschläge unterbreiten.

Es gibt kein generelles Vorgehen, das auf jede Art von Daten angewendet werden kann, um Wissen daraus zu gewinnen. Es ist notwendig, zu Beginn ein Gefühl für die Art der Daten zu bekommen und sich mit diesen auseinanderzusetzen. Um an Informationen und Muster zu gelangen gibt es aber eine Reihe von Teilschritten, die in den meisten Fällen angewendet werden. Der in Abbildung 2.2 gezeigte Knowledge Discovery in Databases Prozess von Fayyad [15] stellt diese dar. Im Folgenden wird auf diese Teilschritte genauer eingegangen und sie werden am Beispiel der Domäne „Microblog“ näher erläutert.

2.2.1 Selektion

Heute ist es möglich, dass eine einzelne Person durch Smartphones, Fitnessbänder oder Smartwatches pro Tag eine Vielzahl an Daten generiert. Meist sind Datenmengen schnell zu groß, um effizient bearbeitet werden zu können. Daher ist es sehr wichtig, im Vorfeld relevante Daten zu ermitteln, um die Menge, die in den Gesamtprozess einfließt, möglichst gering zu halten. Die Selektion der Eingangsdaten ist nicht nur nötig um die Verarbeitungszeit auf ein Minimum zu reduzieren, sondern auch um das spätere Endergebnis des Data Mining Prozesses zu beeinflussen.

2.2.2 Preprocessing

Daten sind in der Regel mit Fehlern behaftet. Arbeitet man mit Daten, die von Menschen generiert wurden, wie Einträgen in Microblogs, können absichtlich oder durch Fehler die gleichen Begriffe auf viele unterschiedliche Arten formuliert sein. Während des Preprocessings wird versucht, die Daten in eine Form zu bringen, in der sie besser weiterverarbeitet werden können. Hierzu können das Entfernen von Stopwords (siehe Abschnitt 3.3.4), die Korrektur von umgangssprachlichen Wörtern und Rechtschreibfehlern (siehe Abschnitt 3.3.3) zählen.

2.2.3 Transformation

Algorithmen und Methoden sind unter Umständen auf ein bestimmtes Format von Eingabedaten angewiesen. So ist es für manche Anwendungen nicht immer möglich, Texte direkt zu verarbeiten und es ist notwendig, diese in eine Vektorform umzuwandeln. Teilweise werden auch nur bestimmte Bestandteile für das weitere Vorgehen benötigt. Hierfür werden die Daten in ein Format überführt, welches es erlaubt, sie weiter zu bearbeiten.

2.2.4 Mustererkennung

Die Eingangsdaten wurden soweit selektiert und vorverarbeitet, dass damit begonnen werden kann, aus ihnen Wissen abzuleiten. Im eigentlichen Data Mining Prozess wird durch Anwenden von Methoden wie Klassifizierung, Regression oder Clustering versucht, sinnvolle Muster zu finden, die in den Daten enthalten sind.

2.2.5 Interpretation & Evaluierung

Die durch den Data Mining Prozess generierten Ergebnisse sind eventuell in einer Form, in der nicht direkt Wissen abgeleitet werden kann. Eventuell müssen sie erst wieder in einem Postprocessing Schritt in eine verständliche Form überführt werden. An dieser Stelle ist auch zu überprüfen, ob das angewendete Verfahren sinnvolle Ergebnisse liefert und einen Gewinn an Wissen darstellt.

2.3 Locality-Sensitive Hashing

Ein gängiges Problem im Data Mining Umfeld ist das Identifizieren von Duplikaten. [31] Hierzu können sowohl Dokumente gehören, welche auf Bitebene gleich sind aber auch nahe Duplikate, welche sich nur in bestimmten Teilen unterscheiden. Anwendungsfälle in der Praxis sind zum Beispiel das Finden ähnlicher Webseiten oder Texte um Urheberrechtsverletzungen zu ermitteln oder das Vorschlagen ähnlicher Produkte in Onlineshops. Im Kontext dieser Arbeit kann dieses Vorgehen eingesetzt werden, um Tweets zu ermitteln, welche sich in einzelnen Wörtern oder Formulierungen unterscheiden aber einen identischen Inhalt aufweisen. Dies ist für den Clustering Ansatz relevant, der in Abschnitt 4.5 beschrieben wird.

Ein naiver Ansatz ähnliche Objekte innerhalb einer Menge zu finden, ist alle Objekte untereinander zu vergleichen. Bei einem solchen Paarvergleich für eine Menge von N Objekten sind $\frac{N*(N-1)}{2}$ Vergleiche notwendig, was einer Komplexität von $O(N^2)$ entspricht. Durch Locality-Sensitive Hashing kann die Komplexität auf $O(N)$ reduziert werden.

Eine herkömmliche Hashfunktion bildet beliebige Werte auf Werte einer festen Länge ab. Durch den Vergleich der Hashwerte können Duplikate von Objekten ermittelt werden. Es werden allerdings nur Duplikate erkannt, die identisch sind. Liegen zwei Objekte vor, die sich nur in einem Bit unterscheiden, resultieren daraus zwei völlig unterschiedliche Hashwerte. Das stellt oft ein Problem dar und ist so nicht gewünscht. Ein langer Text, in dem nur einzelne Wörter oder Buchstaben geändert wurden, stellt nach wie vor ein Duplikat dar, aber wird durch den Einsatz einer normalen Hashfunktion nicht als solches erkannt.

Sogenannte Locality-Sensitive Hashverfahren nehmen sich dieses Problems an. Sie erlauben nicht nur Duplikate zu erkennen, sondern auch Objekte, die nur zu einem gewissen Prozentsatz identisch sind. Die Hashwerte sind eine kurze Repräsentation des ursprünglichen Objekts. Ein Vergleich der Hashwerte von zwei Objekten gibt Auskunft, wie ähnlich die Objekte sind. Ein Beispiel für diese Familie von Hashfunktionen ist der Simhash aus [8].

2.3.1 Simhash

Wir gehen nun von einem Szenario aus, bei dem Webseiten mit ähnlichen Nachrichtenartikeln gefunden werden sollen. Nachdem eine Vorverarbeitung durchgeführt wurde, bei der z.B. HTML-Tags und nicht relevante Inhalte der Webseiten wie Werbung entfernt wurden, erhält man eine Menge von Dokumenten. Diese können wiederum als Bag-of-Words oder als N-Gramme repräsentiert werden. Bei einer Bag-of-Words Repräsentation wird ein Dokument als die Menge seiner Wörter dargestellt. Bei einer Repräsentation durch N-Gramme werden für ein Dokument alle auftretenden Fragmente der Länge N erzeugt. Bei beiden Möglichkeiten erhält man eine Menge von Token für jedes Dokument. Wie man die Eingangsdaten in Token unterteilt, wirkt sich auf die Genauigkeit der späteren Ergebnisse aus. In [31] wird beschrieben, dass für Nachrichtenartikel bei einer Aufteilung in Token die Berücksichtigung von Stopwords zu verbesserten Ergebnissen führt. Beim Vergleich von zwei Dokumenten wäre es nun nötig, das Vorkommen der einzelnen Token zu vergleichen. Dies ist gerade bei großen Dokumenten nicht sehr effizient. Beim Einsatz von Simhashing wird für jedes Dokument eine Signatur gebildet, die dessen Inhalt repräsentiert. Ein Simhash besitzt die Eigenschaft, dass man aus der Hammingdistanz zweier Signaturen

eine Abschätzung der Kosinus-Ähnlichkeit der Dokumente erhält, aus denen die Signaturen erstellt wurden. Bei einer niedrigen Hammingdistanz ist die Kosinus-Ähnlichkeit hoch und bei einer hohen Distanz ist die Ähnlichkeit niedrig.

Algorithmus 1 : Berechnung des Simhashs

Input: Dokument repräsentiert als Token t_1 bis t_n

Output: Simhash zum gegebenen Dokument

```
1:  $W[k] = 0$ 
2: for  $i = 1$  to  $n$  do
3:    $H \leftarrow Hash(t_i)$ 
4:   for  $j = 1$  to  $k$  do
5:     if  $H[j] = 1$  then
6:        $W[j] \leftarrow W[j] + 1$ 
7:     else
8:        $W[j] \leftarrow W[j] - 1$ 
9: for  $j = 1$  to  $k$  do
10:  if  $W[j] > 0$  then
11:     $S[j] \leftarrow 1$ 
12:  else
13:     $S[j] \leftarrow 0$ 
14: return  $S$ 
```

Algorithmus 1 zeigt den Ablauf der Berechnung des Simhashs eines Dokuments. Als erster Schritt wird eine Hashfunktion einer Länge k ausgewählt und für alle n Token eines Dokuments ein Hashwert gebildet. Es kann eine beliebige Hashfunktion wie zum Beispiel MD5 [48] oder SHA-256 [18] verwendet werden. Es werden alle j Bitstellen der Hashwerte der Token betrachtet. Enthält das Bit an Stelle j eine 1 wird das Zwischenergebnis für diese Bitstelle um 1 erhöht, enthält es eine 0, wird es um 1 erniedrigt. Ist das berechnete Zwischenergebnis von Stelle j größer 0, ist der Simhash an Stelle $j = 1$. Ist das Zwischenergebnis kleiner oder gleich 0, ist der Simhash an dieser Stelle 0. Eine bitweise XOR-Verknüpfung der Signaturen zweier Dokumente liefert die Bitstellen, in denen sich die Hashwerte unterscheiden. Die Anzahl dieser Stellen entspricht der Hammingdistanz. Diese geteilt durch die Länge des Hashwerts liefert eine Abschätzung für die Kosinus-Ähnlichkeit beider Dokumente.

2.3.2 Finden von Kandidatenpaaren

Anstatt nun die Signaturen aller Dokumente untereinander zu vergleichen, wird eine Anzahl an Kandidatenpaaren pro Dokument ermittelt, die einzeln auf ihre Ähnlichkeit überprüft werden. Die Signaturen werden in b gleich lange Teile gespalten. Jeder dieser Teile repräsentiert ein Band. Der Inhalt der Teile entscheidet, welchem Topf (engl. Bucket) innerhalb eines Bandes die Signatur (und damit das Dokument) zugeordnet wird. Werden zwei Dokumente dem gleichen Topf innerhalb eines der Bänder zugeordnet, gelten sie als Kandidaten für eine hohe Ähnlichkeit. Durch eine entsprechende Aufteilung in Bänder und Töpfe erreicht man, dass die Paare, die nicht ähnlich genug sind, nicht in einen gleichen Topf innerhalb eines Bandes fallen werden. Durch den Vergleich in mehreren Bändern erhöht man wiederum die Wahrscheinlichkeit, dass ähnliche Paare in den gleichen Topf innerhalb eines Bandes fallen. Wir betrachten exemplarisch den Fall, dass in einer Menge die Dokumente identifiziert werden sollen, die wenigstens eine Ähnlichkeit von 80% aufweisen. Die Signaturen der Dokumente haben eine Länge von 128 Bit ($n = 128$). Es wird eine Aufteilung in 16 Bänder ($b = 16$) und 8 Töpfe ($r = 8$) gewählt. Bei zwei Dokumenten, die eine Ähnlichkeit von 80% zueinander aufweisen, ist die Wahrscheinlichkeit, dass sie in einem Band in den gleichen Topf fallen $0,8^8 \approx 0,17$. Die Wahrscheinlichkeit, dass sie nicht in einem der 16 Bänder identisch sind, berechnet sich folgendermaßen: $(1 - s^r)^b = (1 - 0,8^8)^{16} \approx 0,05$. Das bedeutet,

dass in 5% der Fälle Dokumente mit einer Ähnlichkeit von 0,8 nicht korrekt als Kandidatenpaar ermittelt werden. In 95% ($1 - 0,05 = 0,95$) der Fälle werden Paare also korrekt als Kandidatenpaar ermittelt. Bei zwei Dokumenten mit einer Ähnlichkeit von 30% zueinander ergibt sich die Wahrscheinlichkeit, dass sie in einem Band dem gleichen Topf zugewiesen werden von $0,3^8 \approx 0,00007$. Die Wahrscheinlichkeit, dass die Dokumente in den gleichen Topf fallen und fälschlicherweise als Kandidatenpaar ermittelt werden, liegt bei lediglich 0,1% ($16 * 0,3^8 \approx 0,001$). Dokumente mit einer Ähnlichkeit von 30% werden also in 99,9% der Fälle nicht als Kandidatenpaar ermittelt. Es gilt also mit der Wahl der Aufteilung in Bänder und Töpfe, die Anzahl der False Positives (Kandidatenpaar, das eine zu niedrige Ähnlichkeit aufweist) und False Negatives (Dokumente die sehr ähnlich sind aber nicht als Kandidatenpaar ermittelt werden) auszubalancieren.

Die gerade beschriebenen Schritte lassen sich zu folgenden Ablauf zusammenfassen, der bei der Anwendung von Locality-Sensitive Hashing durchgeführt wird:

1. Ähnlichkeit s wählen, die beschreibt, ab wann ein Dokument als gleich angesehen werden soll.
2. Aufteilung in b Bänder und r Töpfe wählen, wobei $n = b * r$ gelten muss (n beschreibt die Länge der Hashfunktion, die beim Erstellen der Hashwerte der Token verwendet wurde). Zu einer Aufteilung lässt sich durch die Formel $(1/b)^{1/r}$ [31] eine Schwelle berechnen, ab der ein Paar höchstwahrscheinlich als Kandidatenpaar identifiziert wird. Wenn die Genauigkeit wichtig ist (also False Negatives vermieden werden sollen), muss die Schwelle niedriger angesetzt werden. Wenn die Performance wichtig ist (also False Positives verringert werden sollen), muss die Schwelle höher angesetzt werden.
3. Die Signaturen aller Dokumente werden nach der vorher beschriebenen Weise auf die Töpfe verteilt. Dabei ist jedes Dokument in jedem Band einem Topf zugeordnet.
4. Bei einer Kollision in einem Topf bilden die Dokumente Kandidatenpaare.
5. Signaturen der Kandidatenpaare werden geprüft, ob sie der Ähnlichkeitsschwelle s entsprechen oder ob es sich um False Positives handelt.
6. Optional wird der Inhalt der Dokumente direkt überprüft.

2.4 Assoziationsanalyse

Große Onlinehändler wie Amazon, die täglich eine große Menge von Transaktionen bearbeiten, sind stark daran interessiert, welche Produkte bevorzugt, beziehungsweise welche Produkte in Kombination gekauft werden. Dieses Wissen wird genutzt, um Käufern gezielt Produkte vorzuschlagen oder Angebote zu unterbreiten. Das Finden solcher Beziehungen wird als Assoziationsanalyse (engl. Association Rule Learning) bezeichnet. Es werden dabei große Datenmengen analysiert, um Frequent Item Sets oder Assoziationsregeln zu entdecken. In der Literatur wird oft die Warenkorbanalyse verwendet, um das Konzept der Assoziationsanalyse zu erläutern. Folgende Definitionen von Agrawal et al. [1] sind hierzu von Bedeutung. Eine *Transaktion* ist der Inhalt eines Warenkorbs, also die Menge der Produkte, die von einer Person gekauft werden. Ein *Item* beschreibt ein Produkt im Warenkorb. Die Menge aller Transaktionen wird als *Datenbank* bezeichnet. Wir betrachten das Beispiel in Tabelle 2.1, welches dem Buch *Introduction to Data Mining* [52] entnommen wurde.

| <i>TID</i> | <i>Items</i> |
|------------|---------------------------------------|
| 1 | { <i>Brot, Milch</i> } |
| 2 | { <i>Brot, Windeln, Bier, Eier</i> } |
| 3 | { <i>Milch, Windeln, Bier, Cola</i> } |
| 4 | { <i>Brot, Milch, Windeln, Bier</i> } |
| 5 | { <i>Brot, Milch, Windeln, Cola</i> } |

Tabelle 2.1.: Warenkorbbeispiel [52]

2.4.1 Frequent Itemsets

Ziel ist es nun, Produkte zu ermitteln, die oft in Kombination gekauft werden. In einer kleinen Menge von Transaktionen wie der vorliegenden, lässt sich dies leicht lösen. Um für große Mengen die Frequent Itemsets zu ermitteln sind effektive Algorithmen notwendig. Neben dem bekannten Apriori Algorithmus [2] gibt es den effizienteren FP-Growth Algorithmus der in Abschnitt 2.4.3 beschrieben wird.

Eine Itemmenge wird zu einem Frequent Itemset, wenn ihr Vorkommen eine bestimmte Häufigkeitsschwelle überschreitet. Die Häufigkeit des Auftretens wird als *Support* bezeichnet. Die Schwelle kann individuell empirisch bestimmt werden. Geht man von einem minimalen Support von 30% aus, würde sich im Beispiel folgender absoluter Schwellwert ergeben, wobei N die Anzahl der Transaktionen ist:

$$\text{minSupport} = \lceil 30\% * N \rceil = \lceil 0,3 * 5 \rceil = \lceil 1,5 \rceil = 2$$

Das bedeutet, alle Itemsets, die in weniger als 2 Transaktionen vorkommen, werden als nicht „häufig“ angesehen. Zur Veranschaulichung betrachten wir die Itemmenge {*Windeln, Bier*}. Die Kombination der beiden Items kommt in der Menge der Transaktionen in Tabelle 2.1 genau drei mal vor. Somit ergibt sich für die Menge folgender Support:

$$\text{SupportCount}(\{\text{Windeln, Bier}\}) = 3$$

$$\text{Support}(\{\text{Windeln, Bier}\}) = \frac{\text{SupportCount}(\{\text{Windeln, Bier}\})}{N} = \frac{3}{5} = 0,6$$

Da der SupportCount den vorher berechneten minimalen Support überschreitet, kann die Menge in die Liste der Frequent Itemsets aufgenommen werden. Die Menge der Frequent Itemsets, die aus einer Menge von Transaktionen gewonnen werden, kann sehr groß sein. Daher ist es unter Umständen sehr wichtig die Frequent Itemsets so einzugrenzen, dass sie die Gesamtmenge möglichst gut repräsentieren. Oft ist es ausreichend, nur die maximalen Frequent Itemsets während einer Analyse zu betrachten, da in der gesamten Menge zu viele redundante Itemsets enthalten sind. [52]

Ein *maximales Frequent Itemset* ist definiert als eine Menge für die es keine „häufigen“ Übermengen gibt. Das bedeutet, dass alle Itemsets, die eine Teilmenge eines anderen Itemsets sind, kein maximales Frequent Itemset sein können. Die Support-Werte sind dabei unerheblich. Von der Menge der maximalen Frequent Itemsets lassen sich alle Frequent Itemsets ableiten. Es handelt sich also um eine kompakte Darstellung der Gesamtmenge. Lediglich die Support-Werte der Teilmengen gehen so verloren und müssen bei Bedarf über einen zusätzlichen Lauf über die Datenbank wieder ermittelt werden.

Ein Itemset ist *geschlossen*, wenn keine seiner direkten Übermengen den gleichen Support-Wert aufweist. Es handelt sich um ein *geschlossenes Frequent Itemset*, wenn das Itemset geschlossen ist und der Support-Wert größer oder gleich dem minimalen Support ist. Die Beziehung zwischen den verschiedenen Arten von Itemsets ist in Abbildung 2.3 dargestellt.

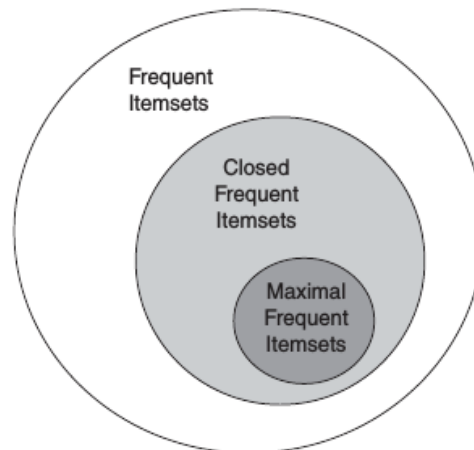


Abbildung 2.3.: Beziehung der Arten von Itemsets [52]

2.4.2 Assoziationsregeln

Aus der Menge der Frequent Itemsets können Regeln der Form $\{X\} \rightarrow \{Y\}$ bestimmt werden. Die Regel $\{\text{Windeln}\} \rightarrow \{\text{Bier}\}$ gibt an, dass Personen die Windeln gekauft haben, auch Bier kauften. Für jedes Frequent Itemset kann eine Menge von Regeln bestimmt werden. Hierzu werden die Elemente der Menge rückstandslos in zwei nicht-leere Teilmengen $\{X\}$ und $\{Y-X\}$ aufgeteilt. Regeln bei denen der linke oder rechte Teil leer ist, sind nicht zulässig, da diese keinen Gewinn an Wissen darstellen. Für eine Menge mit k Elementen können $2^k - 2$ Regeln abgeleitet werden. Aus der Menge $\{\text{Milch}, \text{Windeln}, \text{Bier}\}$ ergeben sich sechs mögliche Regeln:

1. $\{\text{Windeln}\} \rightarrow \{\text{Bier}, \text{Milch}\}$
2. $\{\text{Bier}\} \rightarrow \{\text{Windeln}, \text{Milch}\}$
3. $\{\text{Milch}\} \rightarrow \{\text{Windeln}, \text{Bier}\}$
4. $\{\text{Bier}, \text{Milch}\} \rightarrow \{\text{Windeln}\}$
5. $\{\text{Windeln}, \text{Milch}\} \rightarrow \{\text{Bier}\}$
6. $\{\text{Windeln}, \text{Bier}\} \rightarrow \{\text{Milch}\}$

Wir betrachten exemplarisch die erste Regel. Das Vorgehen bei den restlichen Regeln kann analog durchgeführt werden. Der Support der Regel ist identisch zu dem der gesamten Itemmenge. Jede Regel muss auf ihren Confidence-Wert hin überprüft werden, der sich für eine Regel $\{X\} \rightarrow \{Y\}$ folgendermaßen ergibt:

$$\text{Confidence}(\{X\} \rightarrow \{Y\}) = \frac{\text{SupportCount}(\{X \cup Y\})}{\text{SupportCount}(\{X\})}$$

Der berechnete Wert gibt Auskunft über die Wahrscheinlichkeit, dass Items aus Y auftreten, wenn auch Items aus X vorkommen. Umgangssprachlich wird der Confidence-Wert auch als „Stärke“ einer Regel bezeichnet. Eine gängige Praxis ist es - ähnlich der Bestimmung der Frequent Itemsets - Regeln die unter einem bestimmten Confidence-Wert liegen, zu verwerfen. Für die Regel $\{\text{Windeln}\} \rightarrow \{\text{Bier}, \text{Milch}\}$ ergibt sich ein Wert von 0,5. Zusätzlich zu Support und Confidence existieren verschiedene Maße von

Interessantheit. Meist wird hier der Lift genannt. Dieser gibt an, um welchen Faktor die Wahrscheinlichkeit des Auftretens von $\{Y\}$ gegenüber seiner durchschnittlichen Wahrscheinlichkeit erhöht ist, wenn $\{X\}$ gegeben ist:

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{Confidence(\{X\} \rightarrow \{Y\})}{Support(\{Y\})}$$

Die Regel $\{Windeln\} \rightarrow \{Bier, Milch\}$ in unserem Beispiel hat einen Support von 0,4, eine Confidence von 0,5 und einen Lift von 1,25. Das bedeutet, dass die Regel in 40% aller Transaktionen auftritt. In 50% aller Fälle in denen $\{Windeln\}$ auftritt, tritt auch $\{Bier, Milch\}$ auf. Außerdem ist die Wahrscheinlichkeit des Auftretens von $\{Bier, Milch\}$ gegenüber der durchschnittlichen Wahrscheinlichkeit erhöht, wenn $\{Windeln\}$ auftritt.

2.4.3 FP-Growth Algorithmus

Der FP-Growth Algorithmus [25] bestimmt zu einer Datenbank die Menge der Frequent Itemsets. Hierzu setzt er eine spezielle Baumstruktur ein, den FP-Tree, welcher es ermöglicht, sehr effizient zu arbeiten. Beim Apriori Algorithmus ist die Laufzeit abhängig vom ausgewählten Support Faktor. Außerdem nehmen die Läufe über die Datenbank mit der Dimension der Itemsets zu. [23] Der FP-Growth Algorithmus hingegen benötigt konstant nur zwei Durchläufe. Der Ablauf des Algorithmus ist in zwei Phasen unterteilt. In der ersten Phase wird der FP-Tree generiert und in der zweiten Phase werden daraus die Frequent Itemsets extrahiert.

Generierung des FP-Trees

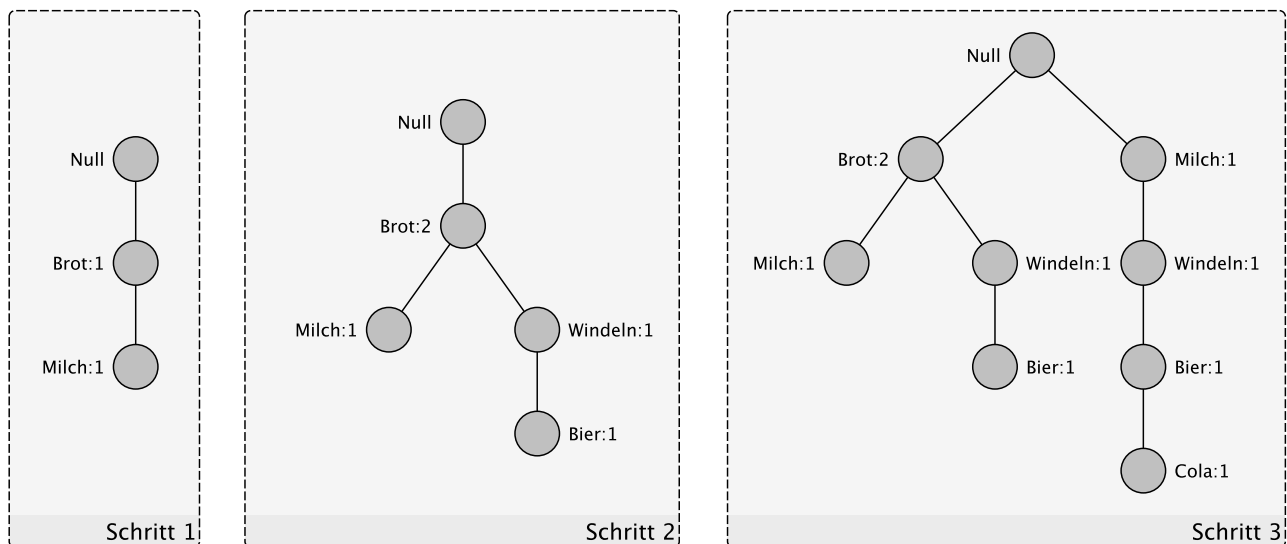


Abbildung 2.4.: Generierung FP-Tree Schritte 1 bis 3

Der FP-Tree ist nichts anderes als eine komprimierte Darstellung der Eingangsdaten. Die Wurzel des Baums wird mit „Null“ gekennzeichnet. Jeder Knoten repräsentiert ein Item und besitzt ein Zählerfeld, welches die Häufigkeit des Auftretens repräsentiert. Wie eingangs beschrieben sind zwei Läufe über die Datenbank notwendig. Im ersten Durchlauf werden die Häufigkeiten der Items ermittelt. Das Ergebnis für die Beispieldaten aus Tabelle 2.1 ist in Tabelle 2.2 zu sehen. Im zweiten Durchlauf wird jede Transaktion abgearbeitet und deren Items jeweils in den FP-Tree eingefügt. Die Items innerhalb einer

Transaktion werden hierfür nach der Häufigkeit absteigend sortiert. Das Item mit der größten Häufigkeit wird als erstes in den Baum eingefügt, um gemeinsame Präfixe zu nutzen und so die Größe des Baums zu minimieren. Der schrittweise Aufbau ist in den Abbildungen 2.4 und 2.5 dargestellt.

| Item | Häufigkeit |
|---------|------------|
| Brot | 4 |
| Milch | 4 |
| Windeln | 4 |
| Bier | 3 |
| Cola | 2 |
| Eier | 1 |

Tabelle 2.2.: Häufigkeiten der Items

Im ersten Schritt wird die erste Transaktion $\{Brot, Milch\}$ eingefügt. Items werden sortiert nach ihrer Häufigkeit in den Baum eingefügt. Begonnen wird mit dem häufigsten Item. Im zweiten Schritt wird die Menge $\{Brot, Windeln, Bier, Eier\}$ hinzugefügt. Für *Brot* existiert bereits ein Knoten und es wird der Präfix der ersten Transaktion genutzt. Es wird lediglich der Wert des Knotens von 1 auf 2 erhöht. Das Item *Eier* wird nicht in den FP-Tree aufgenommen, da seine Häufigkeit geringer als der minimale Support-Wert ist. Die restlichen Transaktionen werden analog zum Baum hinzugefügt und man erhält den finalen FP-Tree.

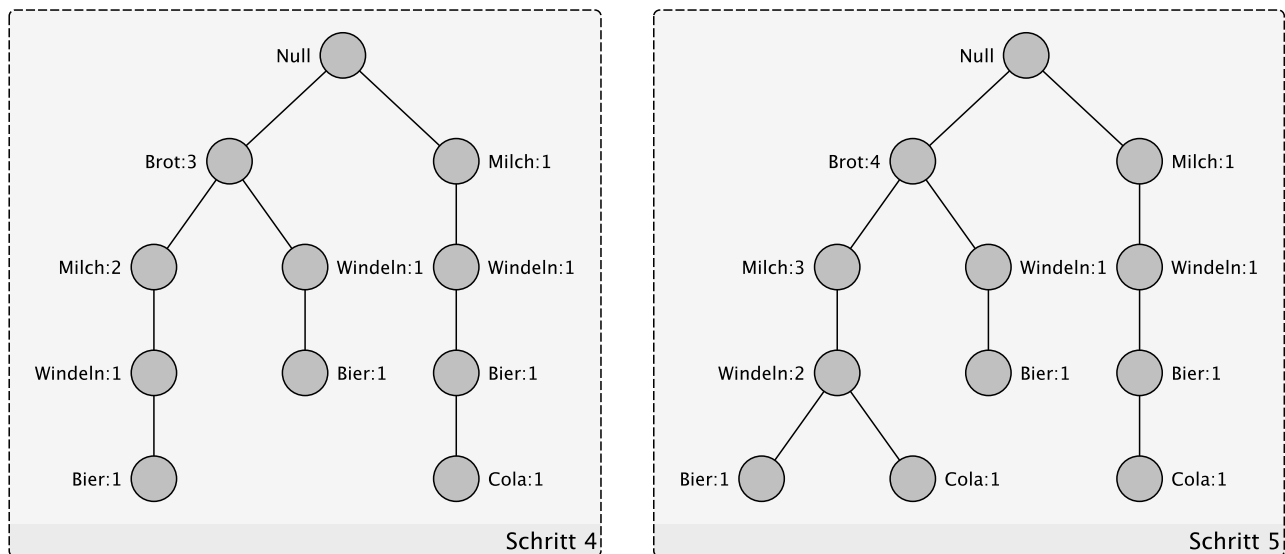


Abbildung 2.5.: Generierung FP-Tree Schritte 4 und 5

Extrahieren von Frequent Itemsets

In der zweiten Phase des Algorithmus wird der FP-Tree genutzt, um die Frequent Itemsets zu extrahieren. Für jedes Item werden Teilbäume des FP-Trees erstellt, die dann rekursiv von den Blättern zur Wurzel abgearbeitet werden. In unserem Fall würden also fünf Teilbäume gebildet werden. Um dieses Vorgehen zu verdeutlichen, betrachten wir den Teilbaum *Bier*, der in Abbildung 2.6 zu sehen ist.

Es wird geprüft ob das Item *Bier* in einer ausreichenden Anzahl enthalten ist. Hierzu werden die Werte aller Knoten des Items addiert, was einen Support von 3 ergibt. Die Menge $\{Bier\}$ bildet damit das Erste der Frequent Itemsets. Anschließend wird der Conditional Tree aus Abbildung 2.6 erstellt, indem folgende Schritte durchgeführt werden:

1. Entlang der Pfade von der Wurzel zu den Blättern müssen die Support-Werte angepasst werden, weil im Teilbaum noch Transaktionen enthalten sind, die das Item *Bier* nicht enthalten. Konkret handelt es sich dabei um die Transaktionen $\{Brot, Milch\}$ und $\{Brot, Milch, Windeln\}$. Für jede der Transaktionen wird der Wert am jeweiligen Knoten um 1 erniedrigt, was im linken äußeren Ast im Conditional Tree zu den Werten Brot:2, Milch:1 und Windeln:1 führt.
2. Nun werden alle Blätter für *Bier* entfernt. Dies ist möglich, da der Baum nach dem Aktualisieren der Werte im vorherigen Schritt jetzt nur noch Transaktionen des Items *Bier* enthält.
3. Durch die Aktualisierung der Werte in Schritt 1 kann es vorkommen, dass der Baum Knoten enthält, die keinen ausreichenden Support-Wert aufweisen. Solche Knoten werden ebenfalls entfernt.

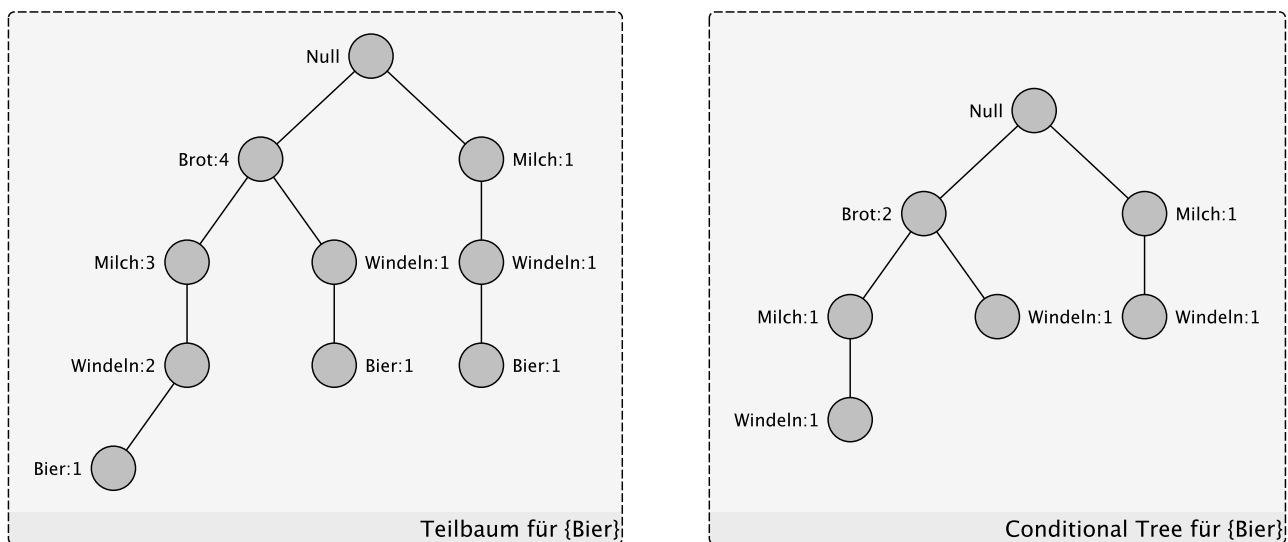


Abbildung 2.6.: Teilbaum des Items Bier

Für die Items, deren Support nach dem Anpassen der Werte größer-gleich dem minimalen Support liegt, wird jetzt ein Rekursionsschritt durchgeführt. In diesem Fall wären das die Mengen $\{Windeln, Bier\}$, $\{Milch, Bier\}$ und $\{Brot, Bier\}$, die auch in die Menge der Frequent Itemsets aufgenommen werden. Für die Teilbäume in Abbildung 2.7 wird das gleiche Vorgehen wiederholt, wie wir es für den Teilbaum $\{Bier\}$ aus Abbildung 2.6 durchgeführt haben. Aus dem Baum auf der linken Seite ergeben sich die Frequent Itemsets $\{Milch, Windeln, Bier\}$ und $\{Brot, Windeln, Bier\}$. Die beiden anderen Teilbäume sind leer und die Rekursion endet ohne neue Itemsets.

Der Algorithmus endet nachdem die Rekursionen für jeden Teilbaum abgeschlossen wurden mit der Menge der Frequent Itemsets:

2.5 Klassifizierung

Während einer Klassifizierung wird ein Objekt anhand bestimmter Merkmale einer oder mehreren Kategorien zugeordnet. Ein Anwendungsfeld ist die Erkennung von Spam. Ein Spamfilter erhält eine Email

| | | |
|---------------------------------------|---------------------------------------|---------------------------------------|
| $\{\text{Bier}\} : 3$ | $\{\text{Brot}\} : 4$ | $\{\text{Milch}\} : 4$ |
| $\{\text{Brot, Bier}\} : 2$ | $\{\text{Windeln, Brot}\} : 2$ | $\{\text{Windeln, Milch}\} : 3$ |
| $\{\text{Windeln, Brot, Bier}\} : 2$ | $\{\text{Milch, Brot}\} : 3$ | $\{\text{Cola}\} : 2$ |
| $\{\text{Milch, Bier}\} : 2$ | $\{\text{Windeln, Milch, Brot}\} : 2$ | $\{\text{Milch, Cola}\} : 2$ |
| $\{\text{Milch, Windeln, Bier}\} : 2$ | $\{\text{Windeln}\} : 4$ | $\{\text{Windeln, Cola, Milch}\} : 2$ |
| $\{\text{Bier, Windeln}\} : 3$ | $\{\text{Windeln, Cola}\} : 2$ | |

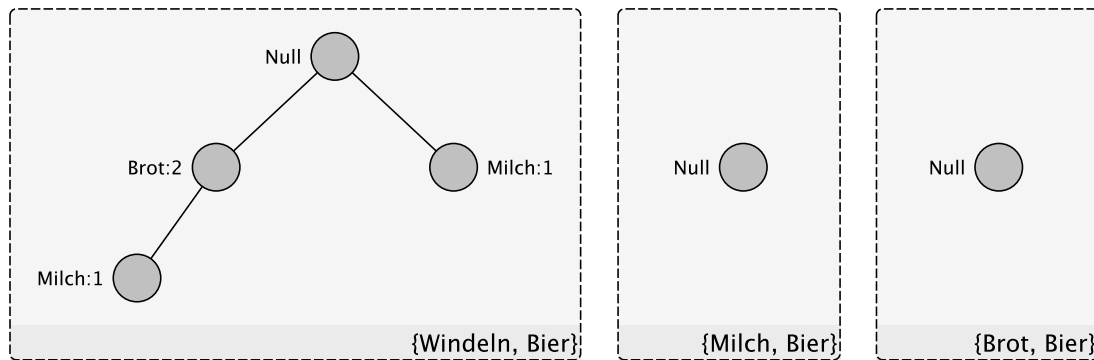


Abbildung 2.7.: Conditional Trees aller rekursiven Aufrufe

und muss automatisiert feststellen, ob es sich dabei um Spam handelt oder nicht. Ein weiteres Anwendungsfeld wäre das automatische Ordnen des eingehenden Briefverkehrs um eine Rechnung der Finanzabteilung oder ein Beschwerdeschreiben dem Kundenservice zuzuordnen.

Ein Klassifizierer benötigt eine Trainingsphase in der Wissen gesammelt wird und wodurch ermöglicht wird, in der Zukunft Objekte eigenständig zu erkennen. Dies wird im späteren Verlauf in Kapitel 4.6 benötigt, da generierte Ergebnisse mit Hilfe eines Textklassifizierers in „interessant“ und „nicht interessant“ aufgeteilt werden sollen.

Eines der Standardverfahren ist das Naive Bayes Klassifizierungsverfahren. Es zeichnet sich dadurch aus, dass es einfach zu realisieren ist, wenig Ressourcen benötigt und trotz wenig Trainingsaufwand gute Ergebnisse liefert. [46] Dieses Verfahren wurde ausgewählt, weil in der Arbeit keine hohe Genauigkeit des Klassifizierers wichtig war, sondern ein robustes, zuverlässiges Verfahren ausgewählt werden sollte, welches ausreichend gute Ergebnisse liefert. Es existieren durchaus andere Ansätze, die im Bereich Textklassifizierung dem Naive Bayes Klassifizierer überlegen sind aber gleichzeitig auch komplexer umzusetzen sind.

2.5.1 Naive Bayes

Naive Bayes ist ein einfaches probabilistisches Verfahren, das auf dem Theorem von Bayes basiert. Man erhält als Klassifizierung nicht nur eine Zuordnung zu einer bestimmten Klasse sondern auch einen Wahrscheinlichkeitswert. Es handelt sich um ein überwachtes Lernverfahren, was eine Menge von Trainingsdaten voraussetzt, die im Vorfeld manuell annotiert werden müssen. Hier kann eine beliebige Anzahl an Klassen erlernt werden. Die späteren Klassifizierungsergebnisse sind abhängig von der Qualität und der Menge der Trainingsdaten. [34] Bevor wir die Funktionsweise genauer betrachten, erfolgt ein kurzer Einschub über die mathematischen Hintergründe.

Bedingte Wahrscheinlichkeit

In der Wahrscheinlichkeitsrechnung bezeichnet die bedingte Wahrscheinlichkeit das Auftreten eines Ereignisses A unter der Bedingung, dass Ereignis B bereits eingetreten ist. Formal ist sie folgendermaßen definiert:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Wir betrachten folgendes Beispiel: Es existiert eine Krankheit und eine speziell für die Krankheit entwickelte Methode um diese zu diagnostizieren. Von einer Millionen Personen haben 0,8% die Krankheit ($P(krank) = 0,008$) und 99,2% haben die Krankheit nicht ($P(gesund) = 0,992$). Im Falle einer vorliegenden Krankheit wird diese zu 98% korrekt diagnostiziert ($P(Pos.|krank) = 0,98$). Das bedeutet in 2% der Fälle wird die Krankheit nicht erkannt obwohl der Patient krank ist ($P(Neg.|krank) = 0,02$). Im Gegenzug schlägt der Test in 3% der Fälle fehl, in denen Patienten gesund sind aber als krank diagnostiziert werden ($P(Pos.|gesund) = 0,03$). In 97% der Fälle wird korrekt erkannt, dass ein Patient gesund ist ($P(Neg.|gesund) = 0,97$). Wie sich dies auf die eine Millionen Patienten auswirkt, ist in Tabelle 2.3 dargestellt. Dabei fällt auf, dass mehr Personen als krank diagnostiziert werden die gesund sind (29760), als es eigentlich Personen mit dieser Krankheit gibt (7840).

| | <i>krank</i> | <i>gesund</i> | Summe |
|-------------------|--------------|---------------|-----------|
| Positive Diagnose | 7.840 | 29.760 | 37.600 |
| Negative Diagnose | 160 | 962.240 | 962.400 |
| Summe | 8.000 | 992.000 | 1.000.000 |

Tabelle 2.3.: Verteilung der Patienten

Nehmen wir an, eine Person erhält eine positive Diagnose. Da die Krankheit in 97% der Fälle richtig diagnostiziert wird, erwartet man, dass die Chancen für diese Person ziemlich schlecht stehen. Hier kommt der Satz von Bayes zum Tragen. Mit ihm kann der Zusammenhang zwischen $P(A|B)$ und $P(B|A)$ berechnet werden. Er lässt sich direkt aus der Definition der bedingten Wahrscheinlichkeit herleiten:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{P(A \cap B)}{P(A)} P(A)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

Wir können dies einsetzen, um zu bestimmen, wie wahrscheinlich es ist, dass der als positiv diagnostizierte Patient wirklich krank ist. Hierzu werden die Wahrscheinlichkeiten für beide Fälle ($P(gesund|Positiv)$ und $P(krank|Positiv)$) berechnet:

$$P(gesund|Pos.) = \frac{P(Pos.|gesund)P(gesund)}{P(Pos.)} = \frac{29760}{37600} \approx 0,79$$

$$P(krank|Pos.) = \frac{P(Pos.|krank)P(krank)}{P(Pos.)} = \frac{7840}{37600} \approx 0,21$$

Man sieht, dass es wahrscheinlicher ist, dass ein Patient obwohl er positiv diagnostiziert wird doch gesund ist.

Lernphase

Da in dieser Arbeit ein Naive Bayes Klassifizierer eingesetzt werden soll, um kurze Texte wie Tweets zu klassifizieren, gehen wir von einem Szenario aus, bei dem einzelne Instanzen unterschiedlicher Länge mit unterschiedlichen Eigenschaften existieren.

Ausgangspunkt ist eine annotierte Menge an Trainingsdaten, die aus einer endlichen Menge von Dokumenten besteht. Jedes Dokument ist bereits manuell einer von mehreren Kategorien zugeordnet worden. Ein Dokument besteht wiederum aus einer Menge unterschiedlicher Wörter, die im Weiteren als Token bezeichnet werden. Anstatt einer Aufteilung eines Dokuments in einzelne Wörter kann auch eine Aufteilung in Wortpaare oder N-Gramme gewählt werden. Bei Naive Bayes handelt es sich um einen *Eager Learning* Algorithmus. Das bedeutet, dass sobald Trainingsdaten vorliegen, daraus ein Modell gebildet wird. Dies hat den Vorteil, dass eine Klassifizierung später schneller durchgeführt werden kann. Das Modell besteht aus den folgenden Wahrscheinlichkeitswerten:

- Häufigkeiten der Instanzen einer Klasse im Vergleich zu den gesamten Trainingsdaten, wobei N die Gesamtanzahl der Instanzen der Trainingsdaten ist und N_{C_i} die Anzahl der Instanz der Klasse C_i :

$$P(C_i) = \frac{N_{C_i}}{N}$$

- Für jeden Token die Häufigkeiten des Auftretens für jede Klasse, wobei N_{T_j, C_i} die Häufigkeit des Tokens T_j in Instanzen der Klasse C_i ist und N_{C_i} die Anzahl der Instanzen der Klasse C_i :

$$P(T_j|C_i) = \frac{N_{T_j, C_i}}{N_{C_i}}$$

Klassifizierung

Ziel ist es zu einem Dokument D die Kategorie C_i zu finden für die die Wahrscheinlichkeit $P(C_i|D)$ maximal ist. Hierfür wird für jede Kategorie C_i ein Wahrscheinlichkeitswert berechnet. Das Ergebnis mit dem höchsten Wert bestimmt die Klasse der das Dokument zugeordnet wird:

$$P(C_i|D) = P(D|C_i) * P(C_i)$$

Die bedingte Wahrscheinlichkeit $P(D|C_i)$, also dass ein Dokument D der Kategorie C_i angehört, berechnet sich aus der Multiplikation der Wahrscheinlichkeiten der einzelnen Token:

$$P(D|C_i) = \prod_{j=1}^d P(T_j|C_i) \tag{2.1}$$

Hierbei muss berücksichtigt werden, dass bei einer Klassifikation Token auftreten können, die nicht in den Trainingsdaten vorhanden waren. In einem solchen Fall wäre das Ergebnis der Multiplikation aus Formel 2.1 gleich 0. Um solche Fälle auszugleichen, wird bei der Berechnung der Wahrscheinlichkeit eines Tokens eine Laplace-Korrektur durchgeführt, wobei d die Anzahl aller unterschiedlichen Token darstellt:

$$\hat{P}(T_j|C_i) = \frac{N_{T_j, C_i} + 1}{N_{C_i} + d}$$

3 Die Streaming API

Eine essenzielle Fragestellung ist, wie die Daten von Twitter auszulesen sind, welche Daten man erhält und welche Qualität diese aufweisen. Dies soll innerhalb dieses Kapitels betrachtet werden. Da das zukünftige System in einem „Live Umfeld“ arbeiten soll, fiel die Wahl auf die kostenlose Variante der Streaming API von Twitter. Bei der Benutzung werden keine einzelnen Anfragen gesendet, sondern es wird eine HTTP-Verbindung aufgebaut, die für die Dauer der Sitzung aufrechterhalten wird. Durch diese Verbindung sendet die API kontinuierlich Daten an den Benutzer. Im Gegensatz zu den REST-basierten APIs können nur Daten gelesen werden und es können keine eigenen Daten an Twitter übermittelt werden.

Bei der Nutzung der Streaming API stehen verschiedene Endpoints zur Verfügung. Nutzt man, wie in dieser Arbeit, den sogenannten Sample Endpoint, so kann man eine allgemeine Stichprobe von ungefähr 1% der kompletten Twitterdaten erhalten. Wie Twitter die Menge von 1% zusammensetzt ist nicht bekannt. Der Dienst gibt an, dass die Menge für alle Benutzer gleich ist, also identische Tweets enthält. Die 1%-Schwelle ist dynamisch und je nach aktuellem Gesamtaufkommen höher oder niedriger.

Nutzt man den sogenannten Filter Endpoint, um die Twitterdaten nach bestimmten Begriffen oder Benutzern zu filtern, besteht die Möglichkeit, nahezu 100% der Tweets zu den angegebenen Parametern zu erhalten. Dies ist der Fall, wenn die Menge der Tweets, die den angegebenen Parametern entsprechen, geringer als 1% des aktuellen Gesamtaufkommens ist. Wird diese Grenze überschritten, so werden Nachrichten unterdrückt und man erhält eine Warnung, die die Anzahl der „abgeschnittenen“ Nachrichten enthält. Morstatter et al. [37] haben untersucht, ob die eingeschränkte Menge an Daten, die über die Streaming API abrufbar ist, die Aktivitäten auf Twitter ausreichend repräsentiert. Hierbei wurden Parameter in Form von Schlüsselwörtern, Geokoordinaten und Benutzern festgelegt. Nach diesen wurden dann Daten über die Firehose API und über den Filter Endpoint der Streaming API gesammelt. Für die verwendeten Parameter konnten über die Streaming API durchschnittlich 43,5% der Daten der Firehose abgerufen werden. Morstatter et al. kommen zu dem Schluss, dass die Streaming API die gesamten Daten ausreichend gut widerspiegelt. Man muss aber bedenken, dass der Filter Endpoint verwendet wurde. Der hier in der Arbeit verwendete Sample Endpoint liefert zu jedem Zeitpunkt maximal 1% des aktuellen Aufkommens auf Twitter und man muss sich darauf verlassen, dass Twitter eine möglichst unabhängige Stichprobe liefert. Möglich wäre also, dass relevante Inhalte in der Stichprobe nicht enthalten sind.

3.1 Inhalt der gesammelten Daten

Neben dem eigentlichen Text eines Tweets liefert die Streaming API noch eine Reihe von anderen Informationen. Tabelle 3.1 zeigt eine Teilmenge der Metadaten, die man zusammen mit einem Tweet im JSON-Format erhält. Insgesamt enthält ein Tweet eine große Menge an Overhead, der für die Zwecke dieser Arbeit nicht relevant ist. Der gesamte JSON-Output beläuft sich für einen Tweet auf ungefähr 7 KB. Das würde bei einer Menge von 50.000 Tweets, was ungefähr dem Aufkommen einer Stunde entspricht, einem Speicherplatz von 350 MB entsprechen. Möchte man für Entwicklungszwecke einen Datensatz über einen größeren Zeitraum abspeichern, ist es daher ratsam, relevante Felder zu identifizieren, damit die Größe in einem handhabbaren Maß bleibt.

Nach der Betrachtung der Daten wurden nur die Felder `tweetId`, `created_at`, `userId`, `retweetId` und `text` gespeichert. Die Felder `coordinates` und `place` sind ebenfalls von Interesse gewesen aber waren in zu wenigen Fällen überhaupt gesetzt. Es existieren Strategien, bei denen auf den Herkunftsort, der im Profil eines Twitternutzers angegeben ist, zurückgegriffen wird. Diese Angaben können aber von den Anwendern frei gewählt werden. Es ist auch nicht sichergestellt, ob der Benutzer den Tweet an seinem

Herkunftsort erstellt hat oder nicht eventuell im Urlaub war. Daher scheidet diese Möglichkeit aus, da sie zu unzuverlässig ist.

| | |
|------------------|---|
| created_at | Der Erstellungszeitpunkt eines Tweets in koordinierter Weltzeit |
| timestamp_ms | Der Erstellungszeitpunkt als Unixzeit |
| id | Der eindeutige Identifier eines Tweets |
| text | Der Text des Tweets |
| user | Ein JSON-Objekt, das Informationen zum Ersteller des Tweets enthält. Dazu zählen unter anderem Name, Herkunft oder Sprache eines Nutzers. |
| coordinates | Ortsangaben in Form von Längen- und Breitengrad |
| geo | Aktuell noch die gleichen Daten wie das coordinates Feld (wird in Zukunft entfernt) |
| place | Informationen über einen Ort auf den sich der Tweet bezieht. Bedeutet nicht, dass der Tweet auch an diesem Ort abgesetzt wurde. |
| retweeted_status | Handelt es sich um einen Retweet, enthält dieses Feld die komplette JSON-Repräsentation des originalen Tweets. |
| entities | Informationen zu einzelnen Bestandteilen, die aus dem Text des Tweets extrahiert wurden. Dazu zählen User Mentions, Hashtags, Urls, aber auch Bilder und Videos, die zusammen mit dem Tweet gepostet wurden. |
| lang | Twitter analysiert eingehende Tweets mit einer Spracherkennungssoftware. In diesem Feld ist das Ergebnis dieses Vorgangs abgelegt. Die Abkürzung einer Sprache ist nach BCP 47 [44] angegeben (Beispiel „en“ für Englisch, „de“ für Deutsch). |

Tabelle 3.1.: Metadaten eines Tweets der Streaming API

3.2 Datenbestand

Um die in der Arbeit entwickelten Ansätze zu testen, wurde ein ausreichend großer Bestand an Twitterdaten gesammelt. Nach der Selektion der relevanten Elemente eines Tweets wurde damit begonnen, möglichst durchgehend alle Daten zu sammeln, die über die Streaming API verfügbar waren. Der Gedanke war, dass es so möglich ist, besondere Ereignisse aufzuzeichnen, die nicht vorhersehbar sind. Zu den Ereignissen, die sich in den Daten besonders widerspiegeln zählen unter anderem die Rosetta Mission [13], die Proteste in Ferguson [51], die Geiselnahme von Sydney [35] und die Anschläge von Paris [59]. Es entstand ein Datensatz von etwa 18 GB über einen Zeitraum vom 29. Oktober 2014 bis zum 28. Januar 2015. Umgerechnet auf einen Tag ergibt das ungefähr 200 MB an Tweets, welche in 24 CSV-Dateien¹ pro Tag aufgeteilt wurden. Eine Speicherung in einfachen Textdateien wurde in diesem Moment gewählt, da die Daten so am flexibelsten weiterzuverarbeiten waren. Jede Zeile enthält die vorher selektierten Felder eines Tweets, die jeweils durch ein Semikolon getrennt sind:

```
TweetId;Erstellungszeitpunkt;UserId;RetweetId;"TweetText"
```

Das Feld RetweetId enthält die ID des originalen Tweets, wenn es sich um einen Retweet handelt oder den Wert 0.

¹ Comma-Separated Values

3.3 Preprocessing

Nach der ersten Betrachtung der Daten der Twitter API wurde der Inhalt der eigentlichen Tweets analysiert. Wie zu Beginn der Arbeit beschrieben, wurde mit dem Sample Endpoint gearbeitet, der eine Stichprobe des gesamten Twitterverkehrs liefert. Zusätzlich wurden nur Tweets in englischer Sprache berücksichtigt; andere Sprachen wurden verworfen. Dies wurde durchgeführt, um die späteren Ergebnisse besser deuten zu können. Man könnte über diese Arbeit hinaus auch Tweets in anderen Sprachen analysieren. Hierzu müssten allerdings die Schritte des Preprocessings angepasst werden.

Die einzelnen Schritte wurden im Laufe der Arbeit immer wieder verändert und angepasst. Das endgültige Ergebnis ist in Abbildung 3.1 zu sehen. Im Folgenden werden die einzelnen Schritte genauer erklärt. Nach jedem Schritt wird die Anzahl der verbleibenden Token geprüft. Alle Tweets, bei denen weniger als zwei Token übrig sind, werden entfernt, da sie keine wirklich nutzbare Information mehr enthalten. Tabelle 3.2 zeigt, wie sich das Preprocessing in den einzelnen Schritten auf die Menge an Tweets eines zufällig ausgewählten Zeitraums von einer Stunde auswirkt.

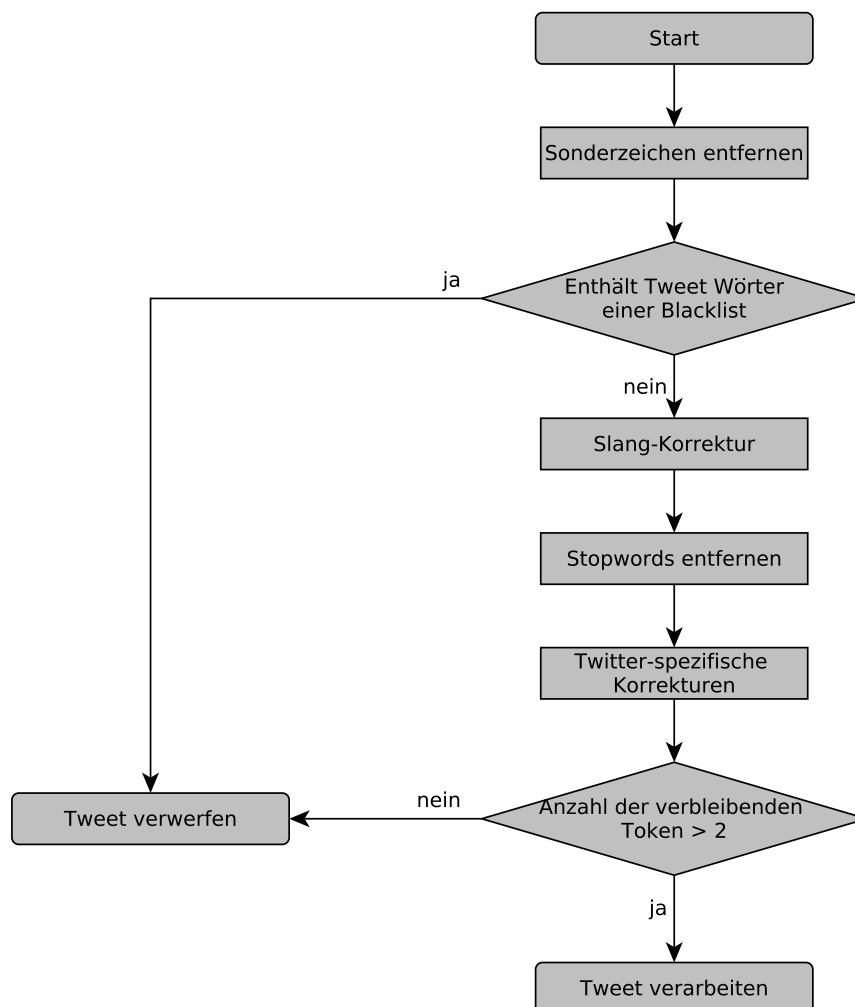


Abbildung 3.1.: Ablauf des Preprocessings

| | Tweets | Token |
|-------------------------------|--------------|---------------|
| Vor Preprocessing | 61773 (100%) | 863330 (100%) |
| Filterung nach Blacklist | 48406 (78%) | 642111 (74%) |
| Filterung nach Sonderzeichen | 48406 (78%) | 533368 (62%) |
| Filterung nach Stopwords | 48243 (78%) | 283614 (33%) |
| Twitter-spezifische Filterung | 48057 (77%) | 253963 (29%) |

Tabelle 3.2.: Auswirkung des Preprocessings

3.3.1 Filterung anhand einer Blacklist

Aus allen gesammelten Daten wurde eine Liste mit den 100 häufigsten Wortpaaren gebildet. Paare wie „happy“ und „birthday“ oder „good“ und „morning“ erzeugen mit ihrem ständigen Vorkommen eine Art Grundrauschen und können auch als eine Art von Stopword gesehen werden. Der Datenstrom an Twiternachrichten wird anhand der Paare gefiltert und alle Tweets, die ein entsprechendes Paar enthalten, werden verworfen. Dieses Vorgehen senkt die Anzahl der Tweets deutlich aber ist in manchen Fällen auch problematisch, wie die folgenden Beispiele zeigen, bei denen es sich nicht um eine wiederholende Diskussion handelt:

1. Wortpaar „person“ und „people“: RT @rantingowl: it's so simple. an unarmed teenager is dead. people want the person who shot him multiple times to have to stand trial. why...
2. Wortpaar „jury“ und „grand“: RT @cletisstump: treating a grand jury like a trial is absurd. indict then present a case & let the defense present theirs. #ferguson

Zusätzlich existiert noch eine Blacklist, die einige Schimpfwörter und pornografische Ausdrücke enthält. Alle Tweets, die einen Begriff beinhalten, der in dieser Liste auftritt, werden direkt am Anfang des Preprocessings entfernt.

3.3.2 Entfernen von Sonderzeichen

Zu Beginn wurden alle Tweets nach einer festgelegten Liste nach Sonderzeichen gefiltert. Da es sich bei Tweets um durch Benutzer erzeugte Texte handelt, können alle Arten und Kombinationen von Sonderzeichen auftreten, was die Pflege einer fest definierten Liste schwierig macht. Oft treten auch exotische Sonderzeichen wie Smileys oder Herzen auf. Daher wurde ein Whitelisting-Ansatz verwendet, der nur Zeichen in bestimmten Bereichen der Ascii-Tabelle zulässt. Es werden nur die Buchstaben des Alphabets, die Zahlen von 0 bis 9, das @-Zeichen und das #-Zeichen akzeptiert.

3.3.3 Korrektur von Umgangssprache

Die Korrektur von umgangssprachlichen Begriffen wie „pls“, „u“ oder „ppl“ erfolgt nach einer kurzen vordefinierten Liste. In den gesammelten Daten wurde ungefähr bei 5% der Tweets eine Korrektur vorgenommen. Eine Verbesserung könnte durch die Erstellung einer umfangreicheren Liste oder die Nutzung eines Dienstes wie Noslang² erzielt werden.

² <http://www.noslang.com>

Beispiele:

1. It seems media will never make this situation clear ppl (= **people**) can be rly (= **really**) stupid! #EbolaInAtlanta
2. Someone pls (= **please**) just sit and tell me their life story
3. I wanna (= **want to**) go out tonight

3.3.4 Entfernen von Stopwords

Stopwords sind Begriffe, die in der Verarbeitung von Texten herausgefiltert werden. In der englischen Sprache sind das Wörter wie „a“, „is“ oder „the“, die in einem Text am häufigsten erscheinen. Ob eine Entfernung von Stopwords sinnvoll ist, hängt immer mit der konkreten Anwendung zusammen. So kann es ein Problem darstellen, zusammengesetzte Hauptwörter zu erkennen, die man eigentlich behalten möchte (Beispiel „The Who“ oder „Take That“). Arbeitet man hingegen mit Häufigkeiten, ist es zu empfehlen, Stopwords zu entfernen, da diese sonst das Gesamtergebnis zu stark beeinflussen. Während der Arbeit wurde eine vordefinierte Liste mit ca. 600 englischen Wörtern verwendet.³

Zusätzlich ist eine Liste entstanden, die Stopwords enthält, die speziell in Twitterdaten häufig vorkommen. Diese Liste besteht aus ca. 20 Wörtern und ist in Anhang A zu finden. Es handelt sich dabei um Wörter, die aus dem Kosmos von Twitter stammen wie „Follower“ oder „Retweet“.

3.3.5 Twitter-spezifische Korrekturen

Innerhalb von Twitter gibt es Themen, die zu bestimmten Zeitpunkten immer in einem erhöhten Maße diskutiert werden. Dazu zählen zum Beispiel Weihnachten oder Halloween. Ein weiteres Muster, das immer wieder auffällt, sind Tweets, die den Namen des aktuellen Monats enthalten. Meistens handelt es sich um Nachrichten, die das aktuelle Datum oder die aktuelle Uhrzeit enthalten. Die Betrachtung einer Menge von Tweets von unterschiedlichen Tagen erweckte den Eindruck, dass die Mehrzahl der Nachrichten ähnlich zu den Beispielen 1 und 2 waren. Trotzdem ist der aktuelle Monatsname nicht ausreichend, um einen Tweet auszusortieren, wie das dritte Beispiel zeigt. Eine Lösung hierfür ist die Betrachtung der verbleibenden Token bei solchen Tweets nach dem Ende des Preprocessings. Enthält ein Tweet den aktuellen Namen des Monats und mehr numerische als nicht-numerische Token, so wird er verworfen. Des Weiteren wurden in diesem Schritt Twitter-spezifische Token entfernt wie URLs und die Kennzeichnung eines Retweets zu Beginn einer Nachricht.

Beispiele:

1. @null http://t.co/aWWM5y57EZ December 14, 2014 at 08:56AM #Phashinal
2. http://t.co/epYNMDKmF1 December 14, 2014 at 08:55AM
3. #EBOLAINATLANTA New Ebola patient came 13 December in Emory University hospital. They knew that there's a city nearby and they didn't care!

³ <https://code.google.com/p/stop-words/>

3.3.6 Handhabung langer Wörter

Bei der Arbeit mit den Daten von Twitter hat sich die Hypothese gebildet, dass Tweets, die einen korrekten Satzbau und korrekt geschriebene Wörter vorweisen, zu einer höheren Wahrscheinlichkeit sinnvolle Informationen enthalten. Dies hat sich auch in der Länge der einzelnen Wörter widerspiegelt. Beim Arbeiten mit den Daten ergab sich eine optimale Maximallänge eines Tokens von 30 Zeichen. Alle längeren Token wurden entfernt. Optional könnte ein Tweet auch komplett verworfen werden, wenn Token enthalten sind, die die Maximallänge überschreiten. Bei langen Token handelt es sich oft um zusammengesetzte Hashtags, wie in den Beispielen 1 und 2. Ab einer Länge von 30 Zeichen war es aber meistens so, dass sich nur noch vereinzelt Hashtag-Begriffe unter den Token befanden.

Beispiele:

1. #christkindlemarketchicago (26 Zeichen)
2. #thehobbithebattleoffivearmies (31 Zeichen)
3. 1/2/32/3/43/4/54/5/65/6/76/7/87/8/98/9/109/10/1110/11/1211/12/13 (70 Zeichen)

Benhardus et al. [6] haben eine Methode beschrieben, bei der alle Tweets als Spam klassifiziert wurden, die einen Token enthielten, der mehr als die Hälfte der Nachricht ausmachte. Dies wurde ebenfalls in den Vorgang des Preprocessings aufgenommen, da es augenscheinlich gute Ergebnisse lieferte.

3.3.7 Säuberung von sonstigem Spam

Eine generelle Erkennung von Spam wurde im Rahmen der Arbeit nicht vorgenommen. Hierbei würde es sich aufgrund des Umfangs um ein eigenes Thema handeln. Es gab aber Überlegungen, wie man die nicht unerhebliche Menge an Spam innerhalb der Daten eindämmen könnte. Mazzia et al. [33] haben in ihrer Arbeit Benutzer, die mehr als 10 Tweets zu einem bestimmten Hashtag posteten, durch einen Faktor herabgestuft, um deren Modell nicht zu stark zu beeinflussen. Dies wurde aufgegriffen und Benutzer betrachtet, die mehrere Tweets pro Stunde erstellten. Eine Betrachtung verschiedener Stichproben aus den gesammelten Daten ergab aber keine zuverlässigen Ergebnisse, so dass dieses Vorgehen wieder verworfen wurde.

4 Erkennen von Trends

Dieses Kapitel geht auf die einzelnen Schritte ein, die im Rahmen dieser Arbeit gemacht wurden, um zum letztendlich entwickelten Ansatz zu gelangen. Als Erstes werden Überlegungen angestellt, was ein Trend ist und wie ein solcher definiert werden kann.

4.1 Was ist ein Trend?

Zu jedem Zeitpunkt des Tages werden die verschiedensten Themen auf Twitter diskutiert. Diese sind sehr heterogen und reichen von Diskussionen über Popstars bis zu Diskussionen über politische Ereignisse. Ein Thema kann durch ein einzelnes Wort oder eine Menge von Wörtern umschrieben werden. Anhand der Häufigkeit kann bestimmt werden, wie populär ein Thema gerade bei den Benutzern ist. Durch die Veränderung der Häufigkeiten kann ermittelt werden, wie sich die Popularität im Laufe der Zeit verändert. Diese Veränderungen können als eine zweidimensionale Funktion gesehen werden, auf deren x-Achse der zeitliche Verlauf und auf deren y-Achse die Häufigkeit dargestellt werden.

Einen Trend könnte man als das Aufkommen und Wachsen eines Themas beschreiben. Alleine daran kann ein Trend aber nicht gemessen werden. So existieren immer wiederkehrende und dauerhaft diskutierte Themen. Die Begriffe „good“ und „morning“ bilden einen täglichen Trend, der sich jeden Tag in einem Zeitfenster am Morgen in den Daten von Twitter abzeichnet. Genauer betrachtet handelt es sich dabei eher um einen Art dauerhaften Trend. Gruhl et al. [22] definieren dieses Verhalten als „Chatter“, eine dauerhaft anhaltende Diskussion über ein Thema. Im Gegensatz dazu existieren sogenannte „Spikes“, die Ausschläge in der Häufigkeit sind, mit der ein Thema diskutiert wird. Sie stellen meist eine Reaktion auf kürzlich stattgefundenere Ereignisse dar. In der Arbeit von Gruhl et al. werden drei Muster beschrieben, nach denen das Auftreten eines Themas unterschieden werden kann:

Mostly Chatter: In Abbildung 4.1 rot dargestellt, beschreibt es allgemeine Themen, die dauerhaft diskutiert werden und im Laufe der Zeit keine großen Veränderungen aufweisen. Als Beispiel wird hier der Begriff „Alzheimer“ aufgeführt. Es handelt sich um ein Thema, das eine bestimmte Gruppe von Personen bewegt aber wahrscheinlich nur zu einem Trend werden würde, wenn ein Heilmittel dagegen gefunden oder eine prominente Person daran erkranken würde.

Spiky Chatter: Das zweite Muster beschreibt Themen die ebenfalls dauerhaft diskutiert werden aber dabei sehr unruhig sind und sehr viele Spitzen aufweisen. Dieses Verhalten ist in Abbildung 4.1 durch den blauen Funktionsgraphen anhand des Begriffs „Microsoft“ dargestellt.

Just Spike: Das letzte Muster beschreibt Themen die überhaupt nicht oder nur in sehr geringem Ausmaß diskutiert werden und deren Häufigkeit in der sie behandelt werden in sehr kurzer Zeit extrem steigt.

In Abbildung 4.1 sieht man sehr deutlich, dass Themen die dauerhaft eine große Varianz in der Häufigkeit aufweisen, in der sie diskutiert werden, leicht andere Themen überlagern können. Folgende Charakteristiken werden im Rahmen dieser Arbeit als Trend angesehen:

1. Themen die nicht zum generellen „Rauschen“ gehören
2. Themen die zu einem vorherigen Zeitpunkt noch nicht oder nur in kleinem Ausmaß aufgetreten sind und deren Häufigkeit stark ansteigt

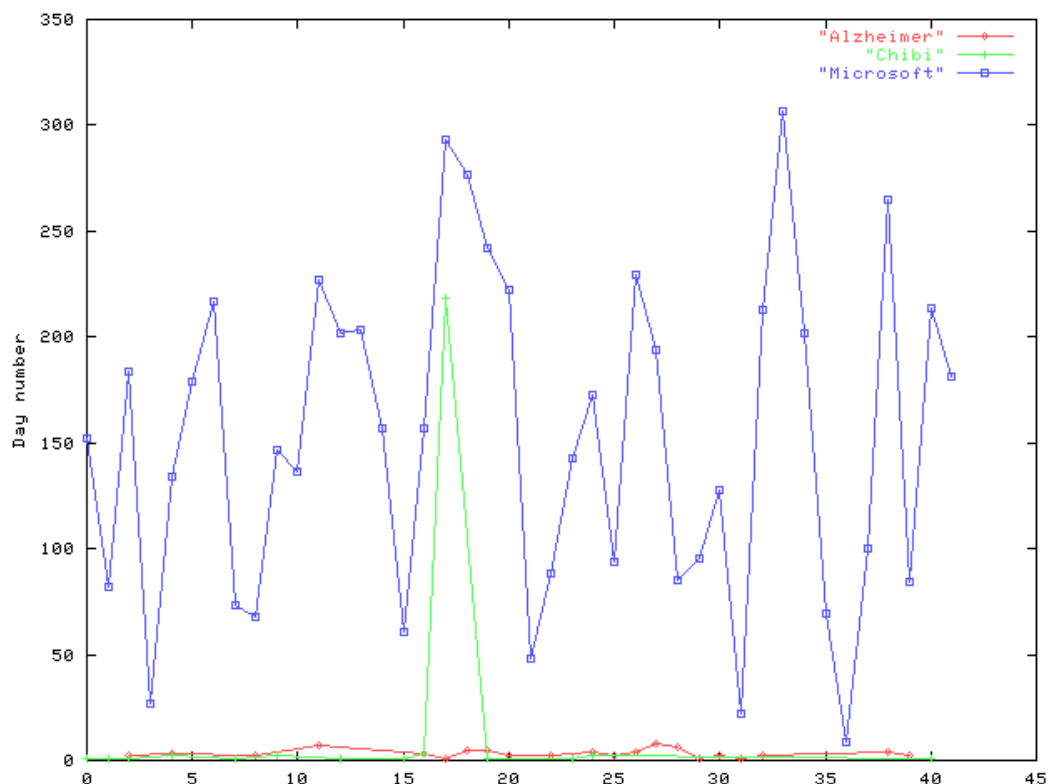


Abbildung 4.1.: Muster nach Gruhl et al. [22]

4.2 Häufigkeitsanalyse anhand von Keywords

Eine erste Idee zu Beginn der Arbeit war das Betrachten der Häufigkeiten einzelner Keywords. Für ein vorher definiertes Zeitintervall wurden alle Tweets zu einer bestimmten Menge von Keywords gesammelt. Nach dem Preprocessing dieser Daten wurden die Häufigkeiten aller enthaltenen Begriffe bestimmt. Durch das Wiederholen dieses Vorgehens konnte die Veränderung der Häufigkeiten von Begriffen pro Zeitintervall erkannt werden. Für das exemplarische Hashtag #Ebola ergaben sich folgende Häufigkeiten: watch (1609), plane (1574), madrid (1455), airport (1407), isolated (1360), people (1318), africa (1231), wobei der Wert innerhalb der Klammer die Anzahl des Auftretens pro Stunde beschreibt. Wie man erkennt, ist das Erahnen einzelner Themen durch die Betrachtung der Begriffe möglich. Die Keywords airport & madrid könnten darauf hinweisen, dass an einem Flughafen in Madrid ein Ebola Fall aufgetreten ist. Die Begriffe isolated & people könnten andeuten, dass Leute in Quarantäne genommen wurden. Dies ist aber nur durch Hintergrundwissen möglich und lässt sich nicht konkret aus den Ergebnissen ableiten. Klare Nachteile dieses Ansatzes sind, dass Zusammenhänge zwischen Begriffen nicht erkennbar sind und dass mehrere Themen vermischt werden. Außerdem sind die Ergebnisse auf die vorher angegebene Menge von Schlüsselwörtern beschränkt.

Dem zweiten Ansatz lag die Idee zu Grunde, dass Häufigkeiten von vorherigen Zeitintervallen genutzt werden könnten, um innerhalb eines aktuellen Zeitfensters neu auftretende Begriffe zu identifizieren. Ein Keyword kommt an einem Tag zu einer Häufigkeit t vor und in einer Stunde zu einer Häufigkeit s . Teilt man beide Werte, erhält man ein Verhältnis, welches das Vorkommen in der Stunde verglichen mit dem Vorkommen des gesamten Tages widerspiegelt. Die Motivation hierfür war, dass bestimmte Begriffe wie love oder people immer sehr häufig vorkommen und damit andere Begriffe verdrängen. In Anhang B befindet sich ein Auszug der Ergebnisse, die nach diesem Verfahren für ein Zeitintervall ermittelt wurden. Aus der Tabelle mit den Ergebnissen lassen sich sowohl die im Zeitintervall neu aufgetretenen Keywords

ablesen, als auch die, die im Vergleich zum Vortag in einer erhöhten Häufigkeit aufgetreten sind. Ein Wert von 0 in der zweiten Spalte gibt an, dass ein Keyword in den Daten des vorherigen Tages nicht enthalten war. Der Ansatz erlaubt, zwischen aktuell häufigen Keywords und „immer“ häufigen Keywords zu unterscheiden. Konkrete Themen sind aber anhand eines einzelnen Begriffs schwer zu erkennen. Außerdem ist es ohne Hintergrundwissen nicht ersichtlich, ob Begriffe zusammen gehören.

4.3 Vergleich von Assoziationen von Keywords

Ausgehend vom vorherigen Ansatz kam die Idee auf, Paare von Begriffen zu bilden, da diese eine größere Aussagekraft besitzen als einzelne Keywords. Paare sind in diesem Fall Wörter, die in einem Tweet gemeinsam auftreten. Die Betrachtung von Wortpaaren hat den Nachteil, dass viele Themen nicht durch zwei Wörter komplett beschrieben werden. Daher wurden auch höherwertige Tupel betrachtet, die aus drei oder vier Wörtern bestanden. Dies führte dazu, dass viele redundante Tupel enthalten waren, die die Menge unnötig vergrößerten. Themen die aus mehr als zwei Wörtern bestehen, wie (Deutschland, gegen, Brasilien) führten zu mehreren Paaren wie (Deutschland, gegen), (Brasilien, gegen) und (Deutschland, Brasilien). Um dem entgegenzuwirken, wurden alle Tupel, die ein Teil eines höherwertigen Tupels waren entfernt. Bei den so ermittelten Tupeln handelt es sich um die in Abschnitt 2.4.1 beschriebenen Frequent Itemsets. Die für eine Stunde der Twitterdaten ermittelten Wortkombinationen können in Anhang C gefunden werden.

Genau wie bei der Betrachtung einzelner Keywords gibt es 2er-Tupel wie (good, morning) oder (happy, birthday) die nahezu immer gemeinsam auftreten. Daher erschien es sinnvoll, solche Kombinationen als eine Art Stopword zu betrachten, die entfernt werden können. Mit dem Skript in Anhang D wurden die häufigen Paare über den kompletten Bestand an Twitterdaten gebildet. Durch das Arbeiten mit den Frequent Itemsets und der Recherche in der Literatur [31] kam der Gedanke auf, aus den vorhandenen Daten Assoziationsregeln zu bilden, worauf im nächsten Abschnitt eingegangen wird.

4.4 Frequent Pattern Mining Ansatz

In Kapitel 2 wurde das Prinzip der Frequent Itemsets und der Assoziationsregeln erläutert. In der Warenkorbanalyse wurden Produktgruppen ermittelt, die häufig in Kombination auftreten. Die dort analysierten Warenkörbe enthalten eine begrenzte Anzahl an Produkten, wobei die Anzahl der unterschiedlichen Produkte sehr hoch sein kann.

Eine Menge von Tweets, von denen jeder Einzelne in einer Bag-of-Words Repräsentation vorliegt, lässt sich sehr gut auf die Warenkorbanalyse übertragen. Ein einzelner Tweet kann durch seine Begrenzung auf 140 Zeichen nur eine beschränkte Menge an Wörtern enthalten. Die Anzahl aller unterschiedlichen Wörter ist dagegen sehr hoch. Ein Wort innerhalb eines Tweets entspricht einem Item und der gesamte Tweet einem einzelnen Warenkorb. Ein Frequent Itemset beschreibt einen Trend als eine Menge von Schlüsselwörtern. Assoziationsregeln ermöglichen es, das Verständnis über Zusammenhänge von Schlüsselwörtern zu verfeinern.

Der hier beschriebene Ansatz wird im weiteren Verlauf der Arbeit als „Frequent Pattern Mining Ansatz“ oder kurz „FPM Ansatz“ bezeichnet. Die Anwendung von Frequent Pattern Mining auf Daten von Twitter wurde bereits in [3] beschrieben. Dort wurden die erzeugten Assoziationsregeln nach Support oder Lift geordnet und die Regeln mit den höchsten Werten bildeten die Trends für das jeweilige Zeitfenster. Es wurde ebenfalls auf verschiedene Strategien eingegangen, um die Menge der Frequent Itemsets zu verringern um bessere Ergebnisse zu erzielen. Dazu zählte unter anderem, dass Itemsets, die eine Teilmenge eines anderen Sets bilden und den gleichen Support-Wert aufweisen, entfernt werden können, da diese zum gleichen Thema gehören. Die daraus resultierende Menge wird in der Literatur als *Closed Frequent Itemsets* bezeichnet und wurde bereits in Abschnitt 2.4.1 eingeführt.

Bei der Anwendung der Assoziationsanalyse werden die Twitterdaten in Zeitfenster aufgeteilt. In dieser Arbeit wurde mit Zeitfenstern von 30 Minuten und einer Stunde experimentiert. In der Arbeit von Petkos

et al. [41] wird von verschiedenen Zeitfenstern gesprochen aber es wird nicht darauf eingegangen, wie man die Informationen zwischen den Zeitfenstern sinnvoll verknüpfen kann. Daher vergleicht der hier vorgestellte Ansatz, wie sich die Ergebnisse zwischen den Zeitfenstern verändern. Eine Veränderung im Wachstum eines Frequent Itemsets oder einer Assoziationsregel spiegelt das Aufkommen und Abklingen eines Themas wider.

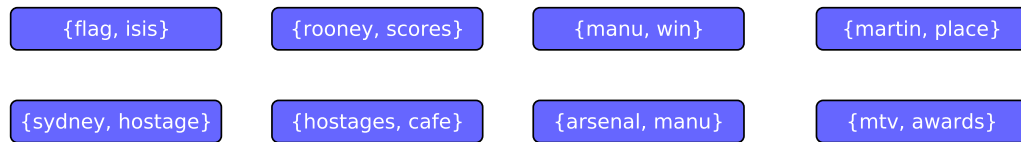


Abbildung 4.2.: Erzeugung des *Time Window Graphen* (Schritt 1)

Zur Ermittlung der Frequent Itemsets wird der FP-Growth Algorithmus aus Kapitel 2.4.3 angewendet. Im Vorfeld werden Parameter wie die Größe der Zeitfenster und ein minimaler Support-Wert festgelegt. Es wird damit begonnen, Tweets von der API zu sammeln. Jeder Tweet wird wie in Abschnitt 3.3 beschrieben vorverarbeitet. Ist das Ende eines Zeitintervalls erreicht, werden die Frequent Itemsets für diesen ermittelt. Die Menge der Frequent Itemsets enthält meistens viele redundante Informationen, wie man im Beispiel in Anhang E sehen kann. Daher werden die maximalen Frequent Itemsets ermittelt und nur diese verwendet. Mit der Menge der maximalen Frequent Itemsets wird nun ein Graph aufgespannt, in den alle Itemsets eingefügt werden. Hierbei wird ein Itemset durch einen Knoten repräsentiert. Das Resultat ist ein Graph, der nur Knoten und noch keine Kanten enthält, wie man in Abbildung 4.2 sieht.

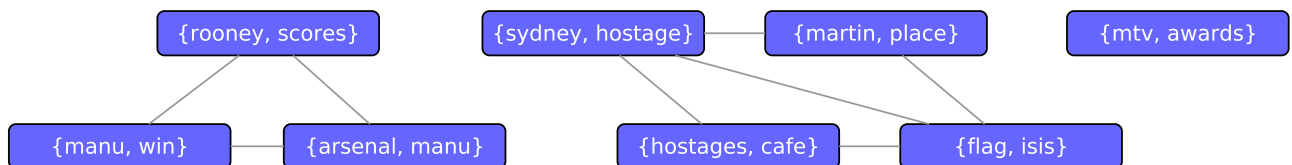


Abbildung 4.3.: Erzeugung des *Time Window Graphen* (Schritt 2)

Im Anschluss werden alle Tweets des Zeitintervalls erneut abgearbeitet und es wird eine Zuordnung erstellt, welche einem Tweet eine Menge von Itemsets zuordnet. Eine Zuordnung wird vorgenommen, wenn alle Objekte eines Itemsets vollständig im jeweiligen Tweet enthalten sind. Werden einem Tweet zwei oder mehr Itemsets zugeordnet, wird für diese im Graphen eine Kante zwischen den Itemsets eingefügt. Abbildung 4.3 zeigt den so entstandenen Graphen, der im Weiteren als *Time Window Graph* bezeichnet wird. Die zusammenhängenden Komponenten innerhalb des Graphen bilden die für das Zeitfenster erkannten Themen. Im Beispiel ist so eine Komponente mit den Itemsets *{rooney,scores}*, *{manu,win}* und *{arsenal,manu}*, eine Komponente mit den Itemsets *{sydney,hostage}*, *{martin,place}*, *{hostages,cafe}* und *{flag,isis}* sowie eine Komponente mit dem einzelnen Itemset *{mtv,awards}* entstanden.

Auf diese Weise werden unterschiedliche Frequent Itemsets, die das gleiche Thema beschreiben zusammengefasst und es treten wenige bis keine Themenfragmentierungen auf. Becker et al. [5] beschreiben in ihrer Arbeit, dass die Fragmentierung von Themen ein Problem ist, mit dem viele Ansätze aus der Forschung umgehen müssen. Tweets mit unterschiedlichen Formulierungen, welche aber inhaltlich das gleiche Thema beschreiben, können von einem System nur schwer als zusammengehörig erkannt werden. Daher stellt der gerade beschriebene Schritt, welcher Itemsets anhand ihres gemeinsamen Auftre-

tens innerhalb eines Tweets zusammenfasst, einen deutlichen Vorteil im Vergleich zu anderen Systemen dar.

Während eines Auswertungsvorgangs wird ein zweiter Graph erstellt, der sogenannte *Main Graph* aus Abbildung 4.4, in den alle *Time Window Graphen* integriert werden. Im *Main Graph* werden neu hinzugefügte Knoten blau dargestellt. Graue Knoten stellen Itemsets aus einem vorherigen Zeitfenster dar. Enthalten verbundene Komponenten ausschließlich blaue Knoten, handelt es sich um ein neu aufgekommenes Thema. Befinden sich blaue und graue Knoten innerhalb einer Komponente, wurden neue Elemente einem bestehenden Thema zugewiesen. In Abbildung 4.4 sieht man, dass die Trends „MTV“ und „Arsenal vs. Manchester United“ neu entstanden sind, der Trend „Sydney“ kam bereits teilweise in einem vorherigen Zeitintervall vor und der Trend „Plane Crash“ stammt vollständig aus einem vorherigen Intervall und war im aktuellen Zeitfenster nicht mehr enthalten.

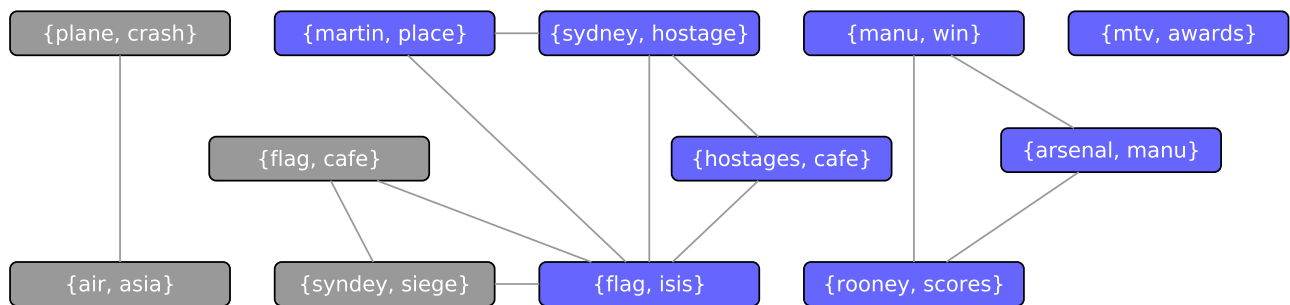


Abbildung 4.4.: Der *Main Graph*

Um zwei Zeitfenster zu kombinieren, werden die zusammenhängenden Komponenten des *Time Window Graphen* ermittelt. Diese können entweder ein neues Thema formen, welches im *Main Graphen* noch nicht enthalten ist oder einem bestehenden Thema aus dem vorherigen Zeitfenster zugeordnet werden. Konkret wird dazu für jeden Knoten einer zusammenhängenden Komponente überprüft ob dieser auch im *Main Graph* existiert. Ist dies der Fall, werden das alte Thema aus dem *Main Graph* und das neue Thema aus dem *Time Window Graph* verbunden. Ein Beispiel dafür bildet die Komponente in Abbildung 4.4, die den blauen Knoten *{sydney, hostage}* enthält. Die grauen Knoten der Komponente stammen aus dem vorherigen Zeitfenster. Die blauen Frequent Itemsets wurden im aktuellen Zeitfenster ermittelt und dem bestehenden Thema zugeordnet. Existiert ein Knoten noch nicht im *Main Graphen*, wird die Komponente übertragen und bildet ein neues Thema.

Die für ein Zeitfenster erkannten Themen werden aus dem *Main Graph* abgeleitet, sobald der *Time Window Graph* integriert wurde. Ein Thema besteht aus einem oder mehreren Frequent Itemsets und aus einer Liste von Tweets. Das Thema wird beschrieben durch die sechs häufigsten Wörter aller in den Itemsets enthaltenen Begriffe. Ein konkretes Beispiel für ein solches Thema sieht folgendermaßen aus:

- Beschreibung: sydney place hostages flag cafe martin
- Zeitfenster: 14. Dezember 2014, 23:46 Uhr bis 15. Dezember 2014, 0:46 Uhr
- Frequent Itemsets: [hostages, cafe], [flag, islamic], [place, martin], [hostages, held], [flag, isis], [sydney, hostage]
- Anzahl ermittelte Tweets: 363
- Beispiel 1: RT @MarketsTicker: Sydney cafe patrons held hostage flag placed in window <http://t.co/9bAnM9XOuw>

- Beispiel 2: RT @BuzzFeedNews: Hostages Held In Sydney Chocolate Shop Forced To Hold Up Islamic Flag <http://t.co/HFOY0ctLNV>
- Beispiel 3: RT @TheMurdochTimes: Sydney airspace shut down, Sydney Opera House evacuated, as Martin Place #cafesiege unfolds. 1 hour in, 13 hostages

Während der Entwicklung des Ansatzes sind unter anderem Tweets mit Wetterinformationen oder Horoskopen aufgetreten, zu denen sich identische Nachrichten sehr oft wiederholten. Der Informationsgehalt dieser Nachrichten ist sehr gering und diese konnten in den meisten Fällen als Spam angesehen werden. Aus diesen Tweets resultierten Themen, die ein Frequent Itemset enthielten und eine Liste von Tweets mit vielen Duplikaten. Durch das Entfernen solcher Themen kann die Gesamtmenge deutlich bereinigt werden.

Der Ablauf des gerade beschriebenen Ansatzes ist in Abbildung 4.5 noch einmal vollständig skizziert. Vorher entwickelte Ideen wie das Bilden von Assoziationsregeln aus den ermittelten Frequent Itemsets oder das Verfolgen des zeitlichen Verlaufs konnten nicht mehr umgesetzt werden. Genauer wird darauf im Abschnitt Future Work in Kapitel 8 eingegangen.

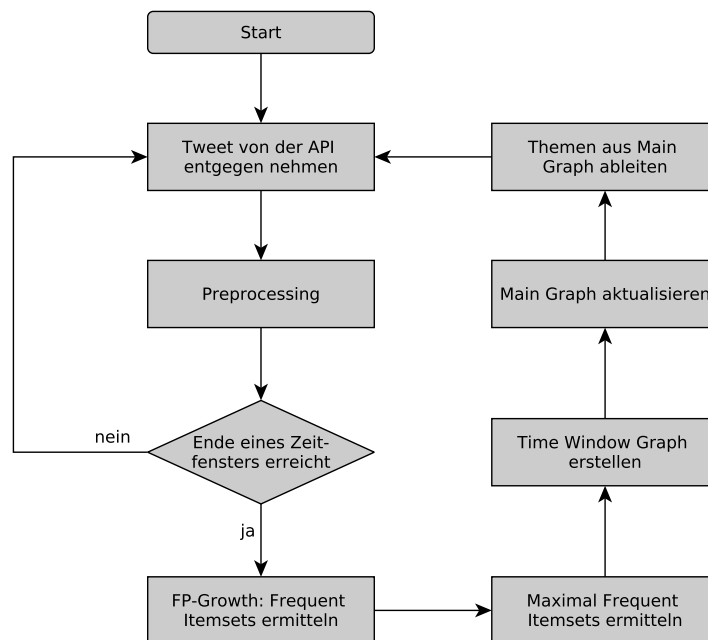


Abbildung 4.5.: Frequent Pattern Mining Ansatz

4.5 Online Clustering Ansatz

Unabhängig zu den bisher beschriebenen Ansätzen kam die Idee auf, ohne Zeitfenster zu arbeiten und Tweets in Echtzeit zu ordnen und in Cluster aufzuteilen. Bei einem ersten Versuch wurde ein eingehender Tweet mit allen bestehenden Clustern verglichen. Hierzu wurde die Kosinus-Ähnlichkeit zwischen den vorverarbeiteten Texten der Tweets verglichen. Lag diese über einer vorher festgelegten Schwelle, wurde der Tweet dem jeweiligen Cluster zugeordnet. Wurde kein ähnlicher Cluster gefunden, bildete der Tweet einen neuen Cluster. Dieses Vorgehen funktionierte für kleine Datenmengen gut aber war nach einer größeren Menge von Tweets nicht mehr effizient nutzbar.

In der Forschung existieren viele Ansätze, die ein ähnliches Vorgehen verwenden. In der Arbeit von Petrović et al. [42] wird zur effizienten Ermittlung ähnlicher Tweets Locality-Sensitive Hashing eingesetzt. Dadurch war es möglich, auch noch in einer großen Menge ähnliche Tweets in angemessener

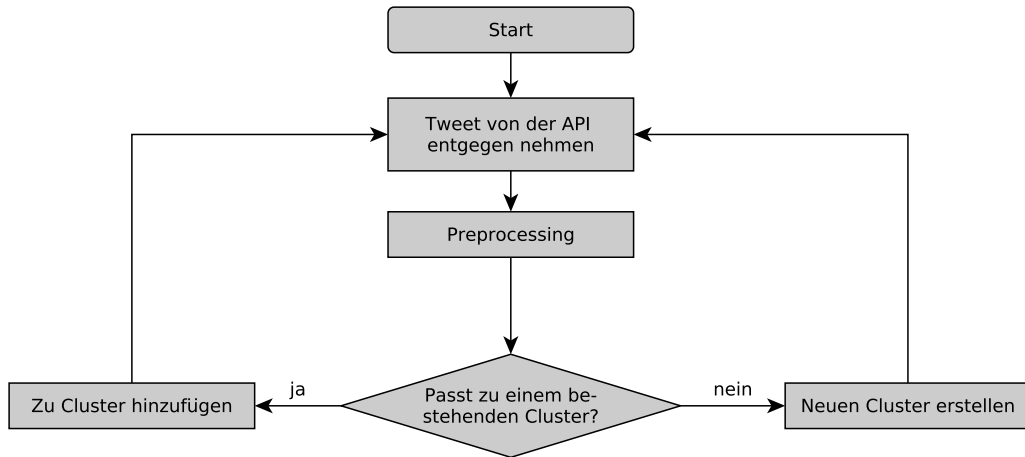


Abbildung 4.6.: Online Clustering Ansatz

Zeit zu identifizieren. Das Verfahren, nachdem in diesem Ansatz Tweets in Clustern organisiert werden, wurde in der Arbeit von Allen et al. [4] als *Single Pass Clustering* zur Erkennung von Ereignissen angewendet. Es findet sich ebenfalls in der Literatur [17] und bezieht sich meist auf das Zusammenfassen von Dokumenten zu Gruppen. Der Ablauf des im Weiteren als „Online Clustering Ansatz“ bezeichneten Verfahrens, ist in Abbildung 4.6 zu sehen. Ein Tweet wird, wie zu Beginn beschrieben, von der API entgegen genommen, vorverarbeitet und dann unter Verwendung von Locality-Sensitive Hashing einem Cluster zugeordnet. Die so entstehenden Cluster weisen verschiedene Merkmale auf, die es erlauben, sie zu gruppieren und zu ordnen:

| | |
|----------------------|---|
| Erster Tweet | Der Text des Tweets durch den ein Cluster erzeugt wurde |
| Erstellungszeitpunkt | Zeitpunkt an dem ein Cluster erstellt wurde |
| Letzte Aktivität | Zeitpunkt an dem der letzte Tweet einem Cluster zugeordnet wurde |
| Aktivitätszeitraum | Der Zeitraum zwischen der Erstellung und dem letzten hinzugefügten Tweet |
| Tweets pro Minute | Anzahl der durchschnittlich pro Minute hinzugefügten Tweets |
| Tweets | Gesamtanzahl der Tweets in einem Cluster |
| Retweets | Anzahl der Retweets in einem Cluster |
| Eindeutige Benutzer | Anzahl der eindeutigen Benutzer in einem Cluster |
| Keywords | Menge und Gewichtung der Keywords die in den Tweets eines Clusters aufgetreten sind |

Tabelle 4.1.: Merkmale eines Clusters

Cluster werden als inaktiv markiert, wenn für einen gewissen Zeitraum kein neuer Tweet hinzugefügt wird. Als Konsequenz sind diese dann abgeschlossen und es ist keine Veränderung mehr möglich. Auf diese Weise werden fortlaufend in die Gesamtmenge neue Cluster hinzugefügt und alte entfernt. Die Anzahl der Cluster pendelt daher innerhalb eines bestimmten Rahmens, was sicherstellt, dass das System auch auf Dauer effizient arbeiten kann.

Einen Vorteil, den der Online Clustering Ansatz bietet, ist die höhere zeitliche Auflösung, da nicht mit festen Zeitfenstern gearbeitet wird. Ein großer Nachteil ist wiederum, dass Themen in verschiedene Cluster fragmentiert werden, da die Unterteilung nur anhand der Kosinus-Ähnlichkeit der Elemente des Textes durchgeführt wird. Dies macht es schwierig, die eigentlichen Themen in der Menge der Cluster zu erkennen. Trotzdem wird durch den Ansatz die Gesamtmenge der Tweets auf ein Maß reduziert, welches es einem menschlichen Betrachter erlaubt, die Daten für einen größeren Zeitraum zu sichten. Die Anzahl der Cluster einer Stunde entspricht ungefähr 0,1% der anfallenden Tweets für denselben Zeitraum.

Die Tabellen 4.2 und 4.3 zeigen zwei Beispiele für Cluster die jeweils einen sinnvollen Trend beschreiben. Eine Idee war es, die Merkmale der Cluster zu nutzen, um einen Klassifizierer anzulernen. Dieser sollte in der Lage sein, bei neuen Clustern zu entscheiden, ob es sich um einen „sinnvollen“ handelt. Eine niedrige Anzahl eindeutiger Benutzer aber eine hohe Anzahl an Tweets könnte eventuell darauf hinweisen, dass es sich um Spam handelt. Ebenfalls könnten Cluster die aus vielen Tweets bestehen aber nur einen kurzen Aktivitätszeitraum aufweisen auf Spam hindeuten. Über Themen aus der echten Welt, die viele Leute beeinflussen, wird in der Regel über einen längeren Zeitraum diskutiert. Dies wurde allerdings nicht umgesetzt, weil der Fokus der Arbeit auf den Frequent Pattern Mining Ansatz gelegt wurde.

| | |
|----------------------|--|
| Erster Tweet | RT @Philae2014: Touchdown! My new address: 67P! #CometLanding |
| Erstellungszeitpunkt | Wed Nov 12 2014 17:04:03 |
| Letzte Aktivität | Wed Nov 12 2014 17:09:07 |
| Aktivitätszeitraum | 4 Minuten |
| Tweets pro Minute | 24 |
| Tweets | 146 |
| Retweets | 141 (97%) |
| Eindeutige Benutzer | 146 |
| Keywords | address, @philae2014, 67p, #cometlanding, @esaoperations, super, awesome, love, #cometlanded, history, @esa, unutterably, #sciencehuman, wow |

Tabelle 4.2.: Beispiel Cluster „European Space Agency“

| | |
|----------------------|---|
| Erster Tweet | RT @PriceSlicerUK: Terrorist Attack #footage today, 11 dead in Paris @Charlie_Hebdo_newspaper http://t.co/5sIQslHCeh |
| Erstellungszeitpunkt | Wed Jan 7 2015 13:58:48 |
| Letzte Aktivität | Wed Jan 7 2015 14:13:25 |
| Aktivitätszeitraum | 14 Minuten |
| Tweets pro Minute | 2 |
| Tweets | 34 |
| Retweets | 34 (100%) |
| Eindeutige Benutzer | 34 |
| Keywords | newspaper, @pricesliceruk, paris, attack, @charlie_hebdo_, today, dead, terrorist, #footage |

Tabelle 4.3.: Beispiel Cluster „Charlie Hebdo“

4.6 Personalisierung von Trends

Die vorgestellten Ansätze dienen dazu, die Menge der Twitterdaten auf ein Maß zu minimieren, welches es einem Benutzer erlaubt, die Daten zu sichten. Die durch die zwei Ansätze generierten Ergebnisse können aber nach wie vor sehr unterschiedliche Trends aus verschiedenen Bereichen wie Sport, Politik oder anderen enthalten. Hinzu kommt, dass jeder Anwender eine eigene Vorstellung davon hat, welche Trends von Interesse sind. Ein Benutzer kann einen Flugzeugabsturz interessant finden und ein anderer Meldungen über die Fußball-Weltmeisterschaft. Deshalb wurde eine zusätzliche Ebene der Filterung eingeführt.

Diese dient als eine Unterstützung für den Benutzer, wodurch für ihn relevante Ergebnisse optisch hervorgehoben werden. Es wird eine binäre Unterscheidung zwischen für den Benutzer *interessant* oder *uninteressant* vorgenommen. Diese wird durch einen Naive Bayes Klassifikator getroffen, der durch den Benutzer während der Arbeit mit dem System angelernt wird.

Im Clustering Ansatz werden die Keywords die einen Cluster beschreiben als Features für den Klassifikator verwendet. Beim Frequent Pattern Mining Ansatz werden die Token aller vorverarbeiteten Tweets verwendet, die einem Frequent Itemset zugeordnet sind.

5 Implementiertes System

Die Ansätze aus den Abschnitten 4.4 und 4.5 wurden in einem konkreten System implementiert, um zu zeigen, dass die theoretischen Überlegungen in der Praxis umsetzbar sind. Des Weiteren soll damit gezeigt werden, dass die Ansätze brauchbare Ergebnisse liefern, was später in Kapitel 6 evaluiert wird. Durch eine praktische Umsetzung und das Arbeiten mit einem System entstehen meist auch neue Ideen und Ansätze, die bei der theoretischen Betrachtung des Problems nicht vorhersehbar waren.

5.1 Allgemeine Architektur

Das implementierte System ist in drei Komponenten aufgeteilt, die in Abbildung 5.1 zu sehen sind. Die zentrale Komponente, gekennzeichnet als „Datensammlung und Trenderkennung“, ist für die Sammlung, Aufarbeitung und Weiterverarbeitung der Twitterdaten zuständig. Hierbei handelt es sich um ein komplett in Java implementiertes System, was die erkannten Trends der beiden Ansätze im JSON-Format zur Verfügung stellt. Die Benutzeroberfläche ist in JavaScript implementiert und ist rein dafür zuständig, die JSON-Daten mit den erzeugten Trends grafisch darzustellen. Es handelt sich um eine Weboberfläche, die sich auf dem lokalen Rechner eines Benutzers befinden kann. Die Daten werden über eine REST Schnittstelle bezogen und durch den Einsatz von Angular JS [27] im Browser dargestellt. Beide Komponenten greifen auf einen Klassifizierer zu, der die selben Trainingsdaten verwendet, um Trends zu klassifizieren. Der Zugriff erfolgt ebenfalls über eine REST Schnittstelle.

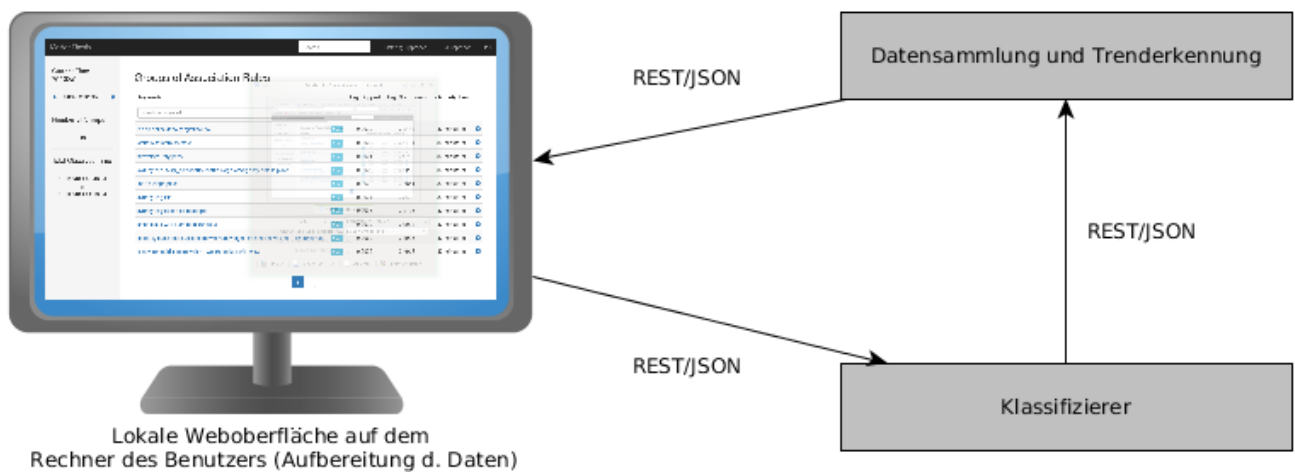


Abbildung 5.1.: Aufbau des Gesamtsystems

5.2 Preprocessing Pipeline

Für die in Abschnitt 3.3 beschriebenen Vorverarbeitungsschritte wurde eine abstrakte Klasse namens „PreprocessingHandler“ eingeführt (Listing 5.1). Diese erlaubt es, für jeden Zwischenschritt einen eigenen Handler anzulegen. Eine beispielhafte Anwendung ist in Listing 5.2 zu finden. Zu Beginn werden alle erstellten Handler in eine Liste eingefügt, die auch die Reihenfolge der Preprocessing-Schritte bestimmt. Ein Tweet wird durch eine gleichnamige Klasse repräsentiert. Durch Aufruf der handle-Methode werden am Tweet Objekt die entsprechenden Manipulationen durchgeführt. Ist der Rückgabewert einer Handler Klasse „null“, wird der Tweet aussortiert und ausstehende Schritte werden übersprungen.

Zum Aufteilen eines Tweets in Token, wurde der Tokenizer des Twitter Natural Language Processing Projekts [54] verwendet.

```
1 public class MyHandler extends PreprocessingHandler {
2
3     ...
4
5     @Override
6     public Tweet handle(Tweet tweet) {
7         // Do something to the Tweet Object
8         ...
9         // If Tweet shall be dropped
10        return null;
11        // Else return Tweet Object
12        return tweet;
13    }
14
15    ...
16
17 }
```

Listing 5.1: Handler Klasse

```
1 // Define Preprocessing Steps to be done
2 List<PreprocessingHandler> preprocessingHandlers = new ArrayList<>();
3 preprocessingHandlers.add(new IrrelevantHandler());
4 preprocessingHandlers.add(new SpecialcharacterHandler());
5 preprocessingHandlers.add(new SlangHandler());
6 preprocessingHandlers.add(new StopwordHandler());
7 preprocessingHandlers.add(new TwitterHandler());
8
9 // A Tweet
10 Tweet tweet = new Tweet();
11
12 // Run Preprocessing
13 for (PreprocessingHandler h : preprocessingHandlers) {
14     if (tweet == null)
15         break;
16     tweet = h.handle(tweet);
17 }
18
19 if (tweet != null) {
20     // Do something
21     ...
22 }
```

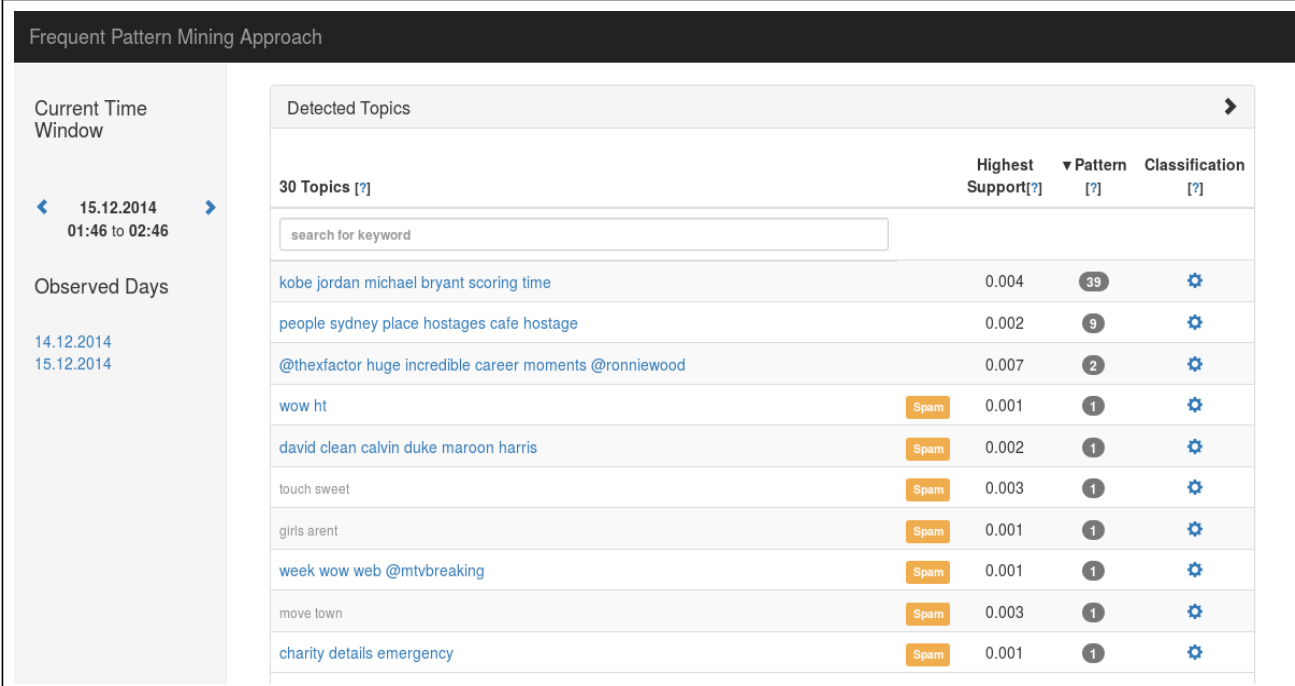
Listing 5.2: Anwendung der Preprocessing Pipeline

5.3 Frequent Pattern Mining Ansatz

Die erzeugten Ergebnisse werden als eine Liste von Topic Objekten repräsentiert. Im Zusammenhang mit der Implementierung werden die Begriffe Topic, Thema und Trend im Weiteren als Synonyme verwendet. Ein Topic Objekt enthält eine eindeutige ID, eine Beschreibung, eine Menge von Tweets, eine Liste zusammengehöriger Frequent Itemsets und eine Liste von Features, die im Klassifizierungsvorgang verwendet werden.

5.3.1 Übersicht

Die Benutzeroberfläche des implementierten FPM Ansatzes ist in Abbildung 5.2 zu sehen. Die graue Spalte auf der linken Seite zeigt das Datum und die Zeitspanne des aktuellen Intervalls an und ermöglicht es dem Benutzer mit Hilfe der Pfeile durch die einzelnen Zeitfenster zu navigieren.



The screenshot shows the 'Frequent Pattern Mining Approach' interface. On the left, there is a sidebar with 'Current Time Window' set to '15.12.2014 01:46 to 02:46' and 'Observed Days' listed as '14.12.2014' and '15.12.2014'. The main area displays 'Detected Topics' with a search bar and a table of 30 topics. The table has columns for 'Highest Support[?]', 'Pattern [?]', and 'Classification [?]'.

| Detected Topics | Highest Support[?] | Pattern [?] | Classification [?] |
|---|--------------------|-------------|--------------------|
| 30 Topics [?] | | | |
| <input type="text" value="search for keyword"/> | | | |
| kobe jordan michael bryant scoring time | 0.004 | 39 | ⚙️ |
| people sydney place hostages cafe hostage | 0.002 | 9 | ⚙️ |
| @thexfactor huge incredible career moments @ronnieewood | 0.007 | 2 | ⚙️ |
| wow ht | Spam 0.001 | 1 | ⚙️ |
| david clean calvin duke maroon harris | Spam 0.002 | 1 | ⚙️ |
| touch sweet | Spam 0.003 | 1 | ⚙️ |
| girls arent | Spam 0.001 | 1 | ⚙️ |
| week wow web @mtvbreaking | Spam 0.001 | 1 | ⚙️ |
| move town | Spam 0.003 | 1 | ⚙️ |
| charity details emergency | Spam 0.001 | 1 | ⚙️ |

Abbildung 5.2.: Themen Übersicht

Unter „Detected Topics“ findet sich die Liste der erkannten Themen für ein Zeitfenster. Die erste Spalte der Liste enthält eine Themenbeschreibung, die aus den sechs häufigsten Begriffen des Themas zusammengesetzt wird. Die Themen lassen sich anhand der Beschreibung nach einer Benutzereingabe filtern, indem das Eingabefeld am Kopf der Tabelle genutzt wird. Die weiteren Spalten der Tabelle enthalten den höchsten Support-Wert der in einem Thema enthaltenen Frequent Itemsets und die Anzahl der Frequent Itemsets, die einem Thema zugeordnet sind. Die Liste kann anhand jeder Spalte sortiert werden.

Per Konfigurationsparameter können potenzielle Spam Themen ein- und ausgeblendet werden. Hat der Benutzer gewählt, dass diese angezeigt werden sollen, erscheint hinter einem Thema eine entsprechende gelbe Kennzeichnung mit der Beschriftung „Spam“.

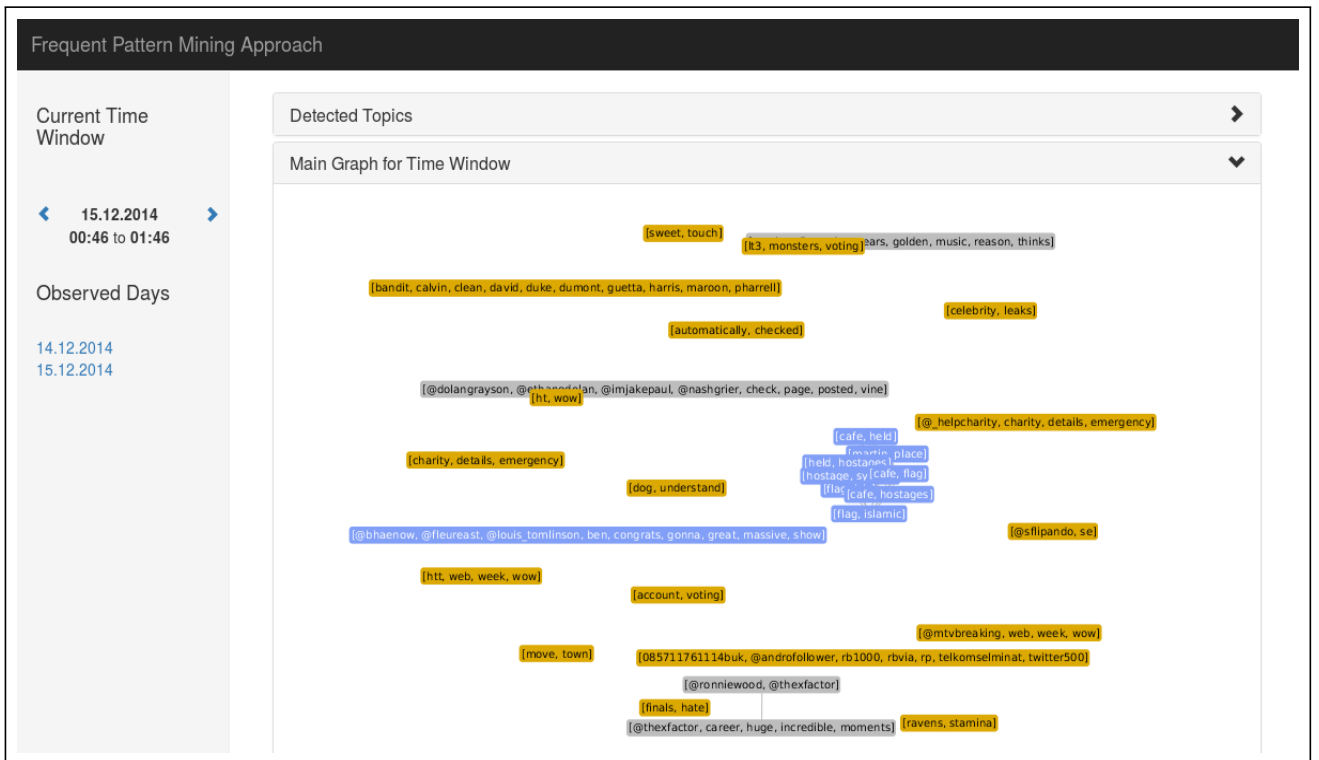


Abbildung 5.3.: Der Main Graph



Abbildung 5.4.: Der Time Window Graph

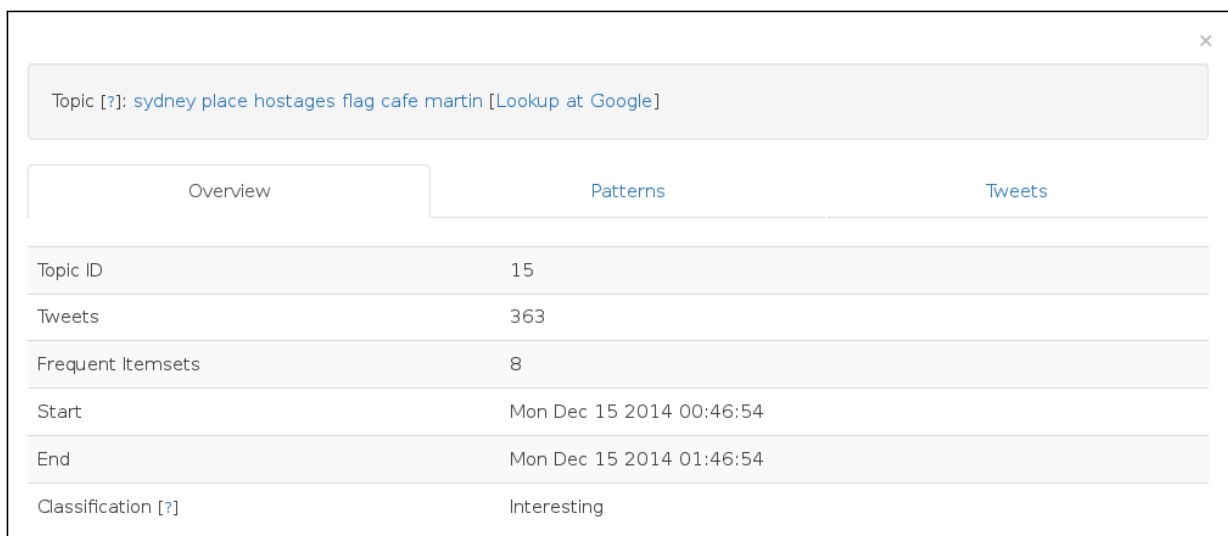
Durch das Zahnrad Symbol in der letzten Spalte der Liste kann der Klassifizierer angelernet werden, indem der Benutzer einzelne Themen als „interessant“ oder „nicht interessant“ kennzeichnet. Vom Klassifizierer als „interessant“ eingestufte Themen werden in der Themenliste in blauer Schrift dargestellt. Umgekehrt werden nicht interessante Themen in grauer Farbe dargestellt. Dies dient als optische Unterstützung für den Benutzer, um für ihn relevante Inhalte hervorzuheben.

Unterhalb der Themenliste befindet sich der aus dem Zeitfenster resultierende Main Graph, zu sehen in Abbildung 5.3. Graue Knoten gehören zu einem der vorherigen Zeitfenster, blaue Knoten wurden im aktuellen Zeitfenster neu hinzugefügt und bei gelben Knoten handelt es sich um als Spam eingestufte Themen. Die vorher beschriebene Liste der erkannten Themen wird aus dem Main Graphen abgeleitet. Jede verbundene Komponente daraus bildet also ein Thema.

Im Abschnitt „New Topics for Time Window“, zu sehen in Abbildung 5.4, ist der Time Window Graph zu finden, welcher alle Themen des aktuellen Zeitfenster enthält. Dieser wird in den Main Graphen eingefügt. Die Screenshots der beiden Graphen dienen ausschließlich der Verdeutlichung. In einem späteren System würde nur die Themenliste angezeigt werden.

5.3.2 Detailansicht

Über einen Klick auf eine der Themenbeschreibungen öffnet sich die entsprechende Detailansicht. Der graue Kasten ganz oben enthält erneut die Themenbeschreibung. Es besteht die Möglichkeit durch einen Link die komplette Beschreibung in einer Suchmaschine nachzuschlagen. Durch Klicken auf einen Begriff, kann nur dieser nachgeschlagen werden. Dies kann sehr hilfreich sein, wenn die Themenbeschreibung auf den ersten Blick nicht sehr aussagekräftig ist.



| Topic [?]: sydney place hostages flag cafe martin [Look up at Google] | |
|---|--------------------------|
| Overview Patterns Tweets | |
| Topic ID | 15 |
| Tweets | 363 |
| Frequent Itemsets | 8 |
| Start | Mon Dec 15 2014 00:46:54 |
| End | Mon Dec 15 2014 01:46:54 |
| Classification [?] | Interesting |

Abbildung 5.5.: Reiter „Overview“

Darunter befinden sich drei verschiedene Karteireiter. Der Reiter „Overview“ aus Abbildung 5.5 enthält verschiedene Informationen zu einem Thema. Die Topic ID ermöglicht es, ein Thema über verschiedene Zeitfenster hinweg eindeutig zu identifizieren, da sich die Beschreibung je nach Intervall verändern kann. „No. of Tweets“ gibt die Anzahl der Tweets an, die ein oder mehrere Frequent Itemsets beinhalten. Es werden jeweils nur die Tweets des aktuellen Zeitintervalls berücksichtigt. Handelt es sich um ein Thema aus einem vorherigen Intervall, so werden für dieses ebenfalls die Tweets des aktuellen Fensters verwendet. Die Zahl in der Reihe mit der Beschriftung „Frequent Itemsets“ gibt die Anzahl der einem Thema zugeordneten Frequent Pattern an. Das Feld „Start“ gibt den Beginn des Zeitfensters an, in dem das Thema zum ersten Mal aufgetreten ist. Das Feld „End“ gibt den Beginn des letzten Zeitfensters an,

in dem das Thema erkannt wurde. Die letzte Zeile gibt an, ob das Thema durch den Klassifizierer als interessant oder nicht interessant eingeordnet wurde.

| Frequent Itemsets | Support | ▼SupportCount |
|-------------------|---------|---------------|
| [cafe hostages] | 0.0017 | 86 |
| [flag islamic] | 0.0014 | 70 |
| [cafe flag] | 0.0013 | 68 |
| [martin place] | 0.0012 | 63 |
| [held hostages] | 0.0012 | 60 |

Abbildung 5.6.: Reiter „Patterns“

Der Reiter „Pattern“, zu sehen in Abbildung 5.6, enthält eine Auflistung aller Frequent Itemsets die dem Thema zugeordnet wurden. Zu jedem Itemset wird zusätzlich der Support-Wert und der Support Count-Wert angegeben.

▼Tweet

- RT @MarketsTicker: Sydney cafe patrons held hostage flag placed in window <http://t.co/9bAnM9XOuw>
- RT @BuzzFeedNews: Hostages Held In Sydney Chocolate Shop Forced To Hold Up Islamic Flag <http://t.co/HFOY0ctLNV> <https://t.co/E4mPEUMATs>
- RT @sunriseon7: PHOTO: #Breaking Confirmation #ISIS flag waved behind Sydney hostages in Martin Place. Live coverage on #TMS7 #sun7 <http://t.co/9bAnM9XOuw>
- RT @TwitchyTeam: Breaking: Report of shotgun wielding man with ISIS flag holding hostages in Sydney, Australia [photos] <http://t.co/YWaRcqt...>
- RT @mediahunter: Hoping for a peaceful resolution to hostage situation in Martin Place, Sydney. <https://t.co/muC17ENSH3>
- @rosierifka @Maggyw519 @AFP Other reports that it's not ISIS flag...
- RT @MadhviPa: guardian reporters @olliemilman @callapilla @heldavidson are tweeting live from the hostage scene in sydney

Abbildung 5.7.: Reiter „Tweets“

Im Reiter „Tweets“, aus Abbildung 5.7, sind alle Nachrichten enthalten, die ein oder mehrere der unter Pattern aufgeführten Frequent Itemsets beinhalten. An dieser Stelle wird der Vorteil des FPM Ansatzes sehr deutlich sichtbar. Die aufgeführten Tweets haben meistens keine bis wenig Ähnlichkeit zueinander aber durch die Beschränkung dass eines der Frequent Itemsets enthalten sein muss, behandeln die Tweets inhaltlich alle das jeweilige Thema.

5.3.3 Klassifizierung

Möchte der Benutzer ein Thema als „interessant“ oder „nicht interessant“ einstufen, erfolgt dies über das Zahnrad-Symbol, welches sich ganz rechts in jeder Zeile befindet. Klickt man darauf, öffnet sich der Dialog aus Abbildung 5.8. Das dort angezeigte Thema wird durch Betätigen der entsprechenden Schaltfläche klassifiziert.

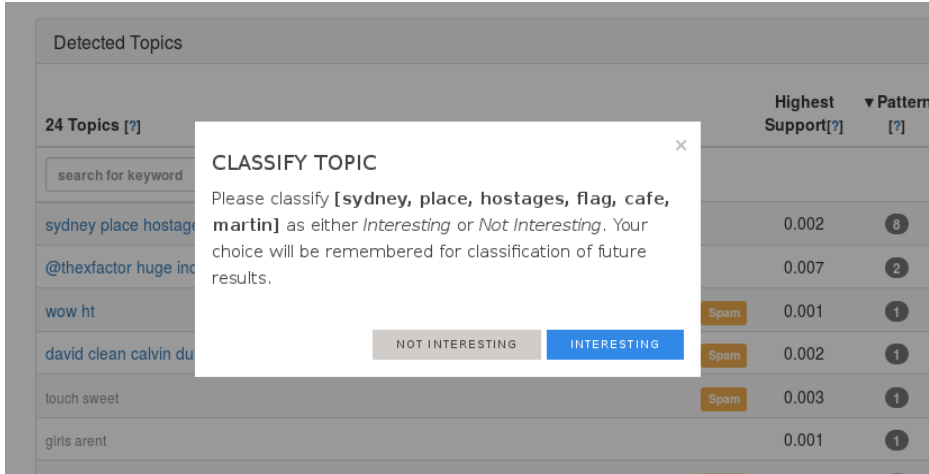


Abbildung 5.8.: Klassifizierung eines Themas

5.4 Online Clustering Ansatz

Im Folgenden wird die Benutzeroberfläche des implementierten Online Clustering Ansatzes beschrieben.

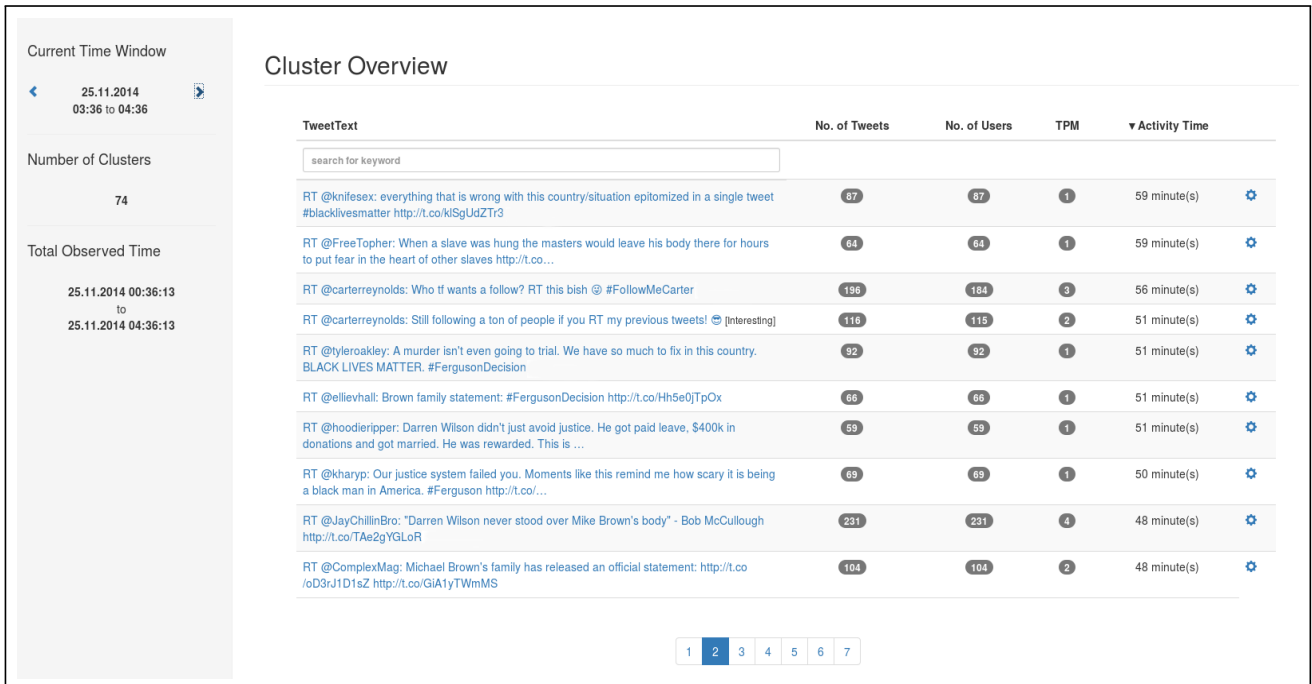
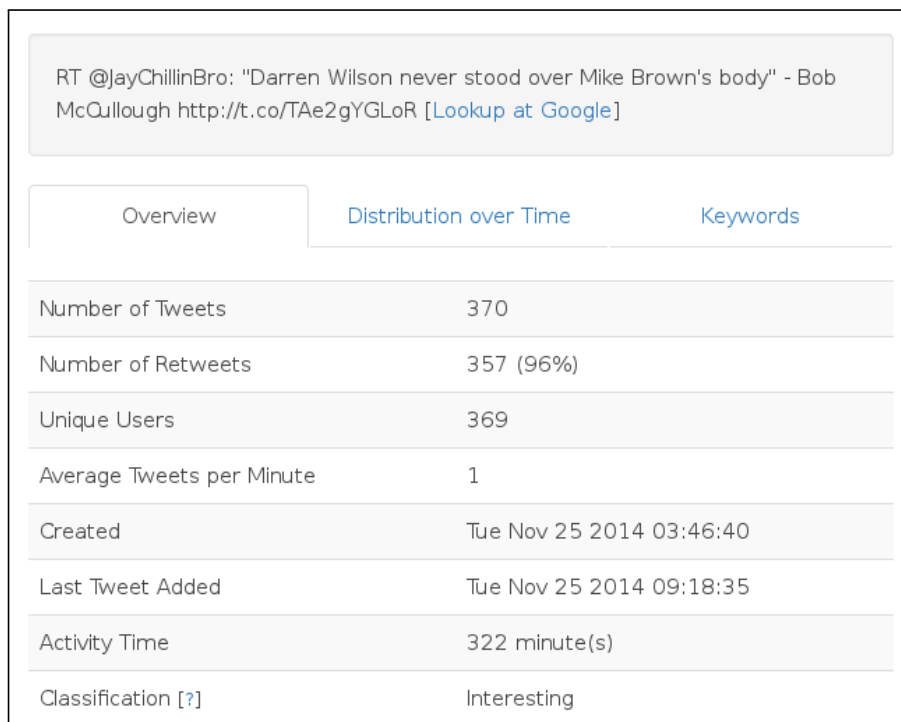


Abbildung 5.9.: Cluster Übersicht

5.4.1 Übersicht

Abbildung 5.9 zeigt die Oberfläche des entwickelten Clustering Ansatzes. Die Tabelle in der Mitte liefert eine Auflistung der Cluster die für ein Zeitintervall ermittelt wurden. Diese lässt sich nach jeder angezeigten Spalte sortieren. In der ersten Spalte finden sich die Beschreibungen der einzelnen Cluster, welche durch den Tweet repräsentiert werden, durch den der Cluster erstellt wurde. Die Beschreibung der Cluster kann durch Eingabe eines Textes gefiltert werden. In den folgenden Spalten findet man die Anzahl der zu einem Cluster zugehörigen Tweets, die Anzahl der verschiedenen Benutzer eines Clusters und die Aktivitätsdauer eines Clusters. Mit Hilfe des Zahnrad Symbols lassen sich wie beim FPM Ansatz einzelne Cluster klassifizieren. Auf dem linken Teil der Oberfläche findet sich das aktuell betrachtete Zeitintervall. Mit den Pfeilen kann ein Benutzer zwischen diesen navigieren. Darunter wird die Anzahl der Cluster des aktuellen Zeitfensters angezeigt, gefolgt vom Gesamtzeitraum der analysiert wurde.

5.4.2 Detailansicht



| | | |
|---|--------------------------|----------|
| RT @JayChillinBro: "Darren Wilson never stood over Mike Brown's body" - Bob McCullough http://t.co/TAe2gYGLoR [Lookup at Google] | | |
| Overview | Distribution over Time | Keywords |
| Number of Tweets | 370 | |
| Number of Retweets | 357 (96%) | |
| Unique Users | 369 | |
| Average Tweets per Minute | 1 | |
| Created | Tue Nov 25 2014 03:46:40 | |
| Last Tweet Added | Tue Nov 25 2014 09:18:35 | |
| Activity Time | 322 minute(s) | |
| Classification [?] | Interesting | |

Abbildung 5.10.: Reiter „Overview“

Durch das Klicken auf einen der Cluster, wird eine Detailansicht geöffnet. Diese ist unterteilt in drei Reiter. Der „Overview“-Reiter aus Abbildung 5.10 enthält die im vorherigen Kapitel beschriebenen Merkmale eines Clusters. Zusätzlich wird dort das Ergebnis der Klassifizierung in „interessant“ oder „nicht interessant“ angezeigt. Abbildung 5.11 zeigt den Reiter „Distribution over Time“, welcher die zeitliche Veränderung der zum Cluster hinzugefügten Tweets pro Minute darstellt. Der Reiter „Keywords“ enthält die Gewichtung der Begriffe die in den Tweets eines Clusters enthalten sind. Dieses Verhältnis wird mit Hilfe einer Wordcloud dargestellt, in der die Häufigkeit von Begriffen anhand von Farbe und Größe des Textes dargestellt wird.

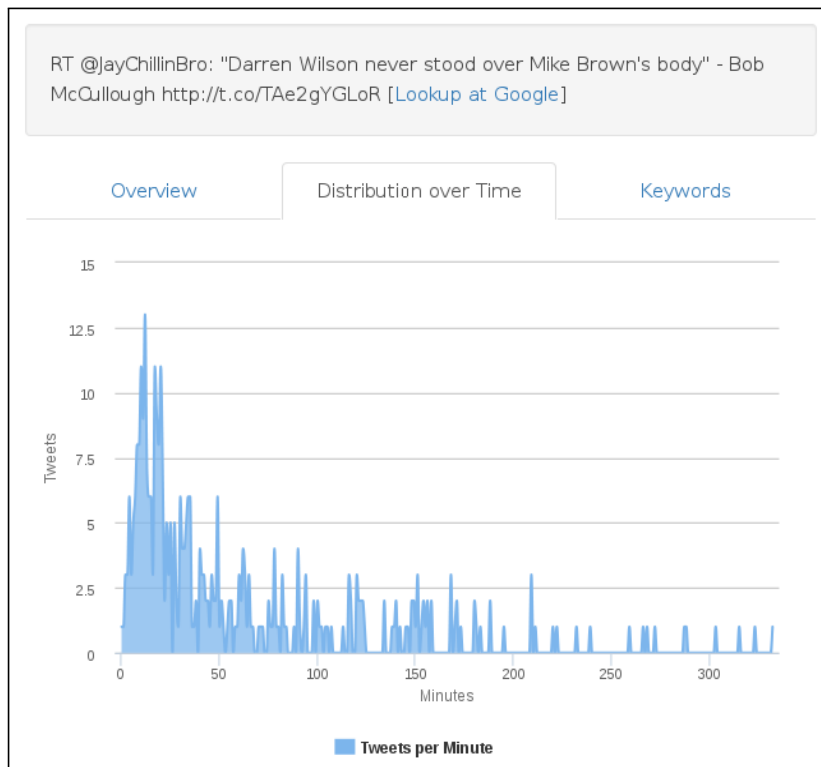


Abbildung 5.11.: Reiter „Distribution over Time“



Abbildung 5.12.: Reiter „Keywords“

5.4.3 Klassifizierung

Die Klassifizierung in „interessant“ und „nicht interessant“ wird auf die gleiche Art wie beim Frequent Pattern Mining Ansatz durchgeführt. Der Benutzer kann über das Symbol auf der rechten Seite einen Dialog öffnen, über den die Einordnung des Clusters vorgenommen werden kann. Es werden dort auch die Schlüsselwörter angezeigt, die zur Klassifizierung verwendet werden.

5.5 TweetGrabber

Zur Kommunikation mit der Streaming API wurde der Hosebird Client [32] verwendet. Es handelt sich hierbei um eine Java Bibliothek, die direkt von Twitter entwickelt wird. Die im Rahmen dieser Arbeit implementierte Komponente zum Sammeln von Twitterdaten liest den Datenstrom der Streaming API, extrahiert aus den erhaltenen JSON-Daten relevante Felder und speichert diese in einzelnen CSV-Dateien. Jede Zeile einer solchen Datei repräsentiert einen Tweet und enthält die in Abschnitt 3.2 beschriebenen Informationen. Die gesammelten Twitterdaten werden automatisch in mehrere Dateien aufgeteilt, welche jeweils die Daten einer Stunde beinhalten.

Es kann sowohl eine allgemeine Stichprobe über den Sample Endpoint bezogen werden als auch nach einer Liste von Begriffen gefiltert werden. Alle gesammelten Tweets werden mit Hilfe einer Spracherkennungsbibliothek¹ analysiert, um nicht englische Tweets herauszufiltern.

¹ <http://code.google.com/p/language-detection/>

6 Evaluierung

Sogenannte „Ground Truth“ Daten sind Daten, deren Klassifikation vollständig bekannt ist. Meist werden diese durch Menschen manuell annotiert und eingesetzt, um Algorithmen oder Systeme zu trainieren und zu evaluieren. Der Ground Truth für Twitterdaten lässt sich nur schwer ermitteln, da jeder Tweet dazu einzeln annotiert werden muss. Arbeitet man wie in dieser Arbeit mit einer allgemeinen Stichprobe an Twitterdaten ist dies nahezu unmöglich, da alleine eine Stunde ca. 50.000 (englische) Tweets enthält. In der Forschung werden verschiedene Ansätze verwendet, um dieses Problem zu umgehen.

Eine Möglichkeit besteht darin, Daten für einen festgelegten Zeitraum zu sammeln und dann über einen zweiten Kanal herauszufinden, was innerhalb des Datensatzes diskutiert wird. Die Annahme hier ist, dass Ereignisse, die viele Menschen beeinflussen, sich innerhalb der Twitterdaten widerspiegeln. Dies konnte auch bei der Sammlung der Daten im Rahmen dieser Arbeit festgestellt werden.

Eine weitere Möglichkeit ist eine Vorfilterung der Daten anhand bestimmter Kriterien wie in [10]. Sind nur Daten zu einem bestimmten Thema wie beispielsweise Politik relevant, werden die Twitterdaten anhand von Begriffen, die mit Politik in Zusammenhang stehen gefiltert und nur die verbleibenden Tweets manuell annotiert. Das Problem beider Ansätze ist, dass sie nicht den vollständigen Ground Truth liefern und in den Gesamtdaten viele weitere Themen enthalten sind.

Eine dritte Möglichkeit ist das Verwenden eines bestehenden Corpus, der auch schon in anderen Forschungsarbeiten eingesetzt wurde. Dies bietet den Vorteil, dass der eigene Ansatz direkt mit anderen verglichen werden kann. Twitter hat seit 2011 seine Nutzungsrichtlinien so eingeschränkt, dass die Bereitstellung von Tweets durch Dritte untersagt ist [56]. Dies hat zur Folge, dass Corpora nur in Form von Tweet Identifiern veröffentlicht werden dürfen. Diese müssen dann manuell mit Hilfe der Twitter Search API extrahiert werden, was durch die Limitierung der API Zugriffe je nach Menge der Tweets sehr lange dauern kann. Folgende Twitter Corpora sind in anderen Forschungsarbeiten entstanden und können für eigene Forschungszwecke verwendet werden:

- Der TDT5 Topics and Annotations Datensatz [20] enthält Meldungen von Nachrichtenagenturen in den Sprachen Englisch, Chinesisch und Arabisch. Es wurden 250 verschiedene Themen annotiert. Der Datensatz wurde in verschiedenen Forschungsarbeiten eingesetzt, die sich mit dem Schwerpunkt Themenerkennung befassen. Der Datensatz besteht nicht aus Twitterdaten und muss zudem gegen eine Gebühr erworben werden.
- Der Tweets2011 Corpus aus [39] enthält Tweets aus dem Jahr 2011 über einen Zeitraum von 15 Tagen. Enthalten sind die Tweet Ids und der Nutzernamen des Erstellers. Die Inhalte müssen also wie bei den anderen Datensätzen auch nachträglich mit Hilfe der Search API bezogen werden. Es sind sowohl „wichtige“ als auch „spam“ Nachrichten enthalten und es existiert keine Annotierung. Der Corpus unterscheidet sich also nicht von dem in der Arbeit erstellten Corpus.
- Der Edinburgh Twitter Corpus von Petrović et al. [43] enthält die Identifier von 52 Millionen Tweets. Von diesen wurden ca. 3.000 manuell annotiert und jeweils einem von 27 Real-World Ereignissen zugewiesen. Dies ist natürlich nur eine kleine Teilmenge aller Ereignisse innerhalb der Gesamtmenge an Twitterdaten und somit ebenfalls nicht als Ground Truth verwendbar. Gedacht ist der Datensatz, um Systeme aus dem Bereich Event Detection zu überprüfen.

-
- Der Twitter Topic Detection Datensatz [3] umfasst Tweets von drei Real-World Ereignissen, dem FA Cup Finale, dem Super Tuesday for US Elections und den US Wahlen 2012. Die Tweets sind in Form deren Identifier angegeben. Zusätzlich sind verschiedene Ereignisse in Form von Schlüsselwörtern und dem Zeitpunkt des Auftretens angegeben. Die Ereignisse wurden anhand verschiedener Nachrichtenseiten ausgewählt.

6.1 Evaluierung der Trenderkennung

Keiner der aufgeführten Corpora konnte für die Evaluierung in dieser Arbeit direkt genutzt werden. Lediglich der Edinburgh Twitter Corpus enthielt eine Menge annotierter Tweets zu verschiedenen Real-World Ereignissen. Da die Definition eines Trends in dieser Arbeit sehr weitläufig ist, hätte eine Evaluierung mit diesem Datensatz auch keine zuverlässige Aussage über die Funktionsweise des erstellten Systems geliefert.

Eine eigene Annotierung wäre nicht zielführend gewesen, da das System dann nur auf einem kleinen Datensatz und damit einem kurzen Zeitraum evaluiert werden könnte. Außerdem war es nicht Ziel der Arbeit, ein System zu entwerfen, was einem anderen konkreten Ansatz überlegen ist. Daher wird im weiteren Verlauf auf die Evaluierung der Trenderkennung verzichtet. Der Fokus wird darauf gelegt, zu beschreiben, welcher Ansatz intuitiver ist, beziehungsweise welcher Ansatz ausdrucksstärkere und qualitativ bessere Ergebnisse liefert.

6.1.1 Frequent Pattern Mining Ansatz

Der Frequent Pattern Mining Ansatz verwendet einen Graphen, um die Itemsets zu identifizieren, die zu einem gleichen Thema gehören und verhindert so Themenfragmentierungen sehr zuverlässig. In Einzelfällen kann es allerdings auftreten, dass einzelne Themen über mehrere Zeitfenster hinweg nicht als zusammengehörig erkannt werden können. Tritt in einem Zeitfenster beispielsweise ein Itemset $\{Wort_1, Wort_2, Wort_3\}$ und im folgenden Zeitfenster ein Itemset $\{Wort_2, Wort_3, Wort_4\}$ auf, werden diese nicht als zusammengehörend erkannt. Dies liegt daran, dass die Strategie zur Verknüpfung einzelner Trends rein anhand des Itemsets entscheidet.

Bei der Verknüpfung einzelner Itemsets wird darauf geachtet, ob diese innerhalb eines Tweets gemeinsam vorkommen. Ist dies der Fall, werden diese innerhalb des Graphen verbunden. Zusätzlich werden Itemsets verbunden, wenn diese den selben Benutzernamen enthalten. Dies führt in bestimmten Fällen dazu, dass zwei getrennte Trends zusammengefasst werden. Würden ein Itemset $\{China, Earthquake, @tagesschau\}$ und ein Itemset $\{Washington, attack, @tagesschau\}$ auftreten, die beide unterschiedliche Themen beschreiben, würden diese vom System wegen des Vorkommens des Benutzers *@tagesschau* zusammengefasst werden.

Wie zuvor beschrieben, wurde eine Liste mit den gängigen Wortkombinationen erstellt, die als eine Stopwordliste dient. Diese Wortpaare sind wichtig, um dauerhaft diskutierte Themen zu erkennen, die ausgeblendet werden sollen. Außerdem ist das Entfernen dieser Paare wichtig, damit die eigentlichen Trends innerhalb des Graphen voneinander abgegrenzt werden können. Entfernt man die Paare nicht, treten Fälle auf, in denen unterschiedliche Trends innerhalb des Graphen zusammengefasst werden, wie man in Abbildung 6.1 sehen kann. Hier sorgen die Paare (*happy, birthday*), (*good, luck*) und (*video, music*) für eine Verbindung verschiedener Themen.

Die Liste der häufigen Wortkombinationen wurde über einen Datenbestand von drei Monaten erstellt. Aus dieser werden die 1.000 häufigsten Kombinationen verwendet. Trotzdem gibt es Fälle, in denen alltägliche Wortkombinationen auftreten, die nicht in der Liste enthalten sind. Das Paar (*cut, hair*) ist eine solche Kombination die innerhalb eines Zeitintervalls in erhöhter Form aufgetreten ist. Dies führte zu einem Trend, der viele Tweets zum Thema Frisuren und Schneiden von Haaren enthielt. Es wäre also zu überlegen, ob die Liste verlängert werden sollte oder eventuell nach jedem Tag neu dynamisch erstellt werden sollte.

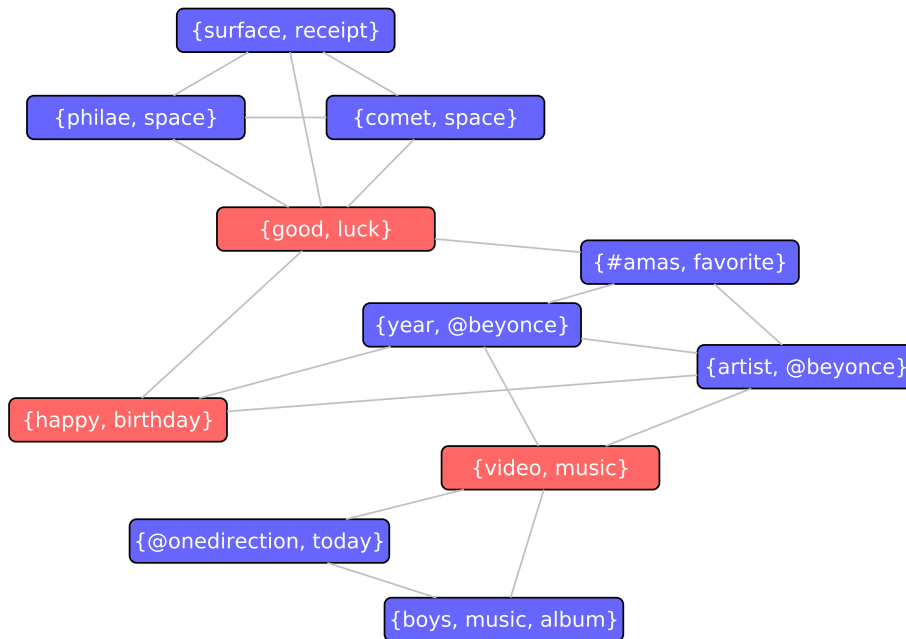


Abbildung 6.1.: Trends ohne Entfernung der häufigen Wortkombinationen

Bei der Betrachtung der generierten Ergebnisse ist aufgefallen, dass Trends besser erkannt und verfolgt werden können, wenn viele Leute darüber diskutieren. Durch unterschiedlich formulierte Tweets entstehen viele verschiedene Itemsets, die es wiederum erleichtern die Komponenten des Graphen über die Zeitfenster hinweg zu verbinden. Ein Fall, in dem dies besonders gut sichtbar wurde, war die Geiselnahme von Sydney [35]. Dieser Trend konnte über mehrere Zeitfenster hinweg zuverlässig verfolgt werden. Alleine durch die Themenbeschreibung, welche sich über die Zeitfenster hinweg veränderte, konnte die Entwicklung der Geiselnahme gut verfolgt werden. Tabelle 6.1 zeigt die zeitliche Veränderung der Trendbeschreibung am 15. Dezember 2014.

| Zeitfenster | Beschreibung des Trends |
|-------------------------|---|
| 7:46 Uhr bis 8:46 Uhr | Sydney siege: hostages held in central cafe - watch live |
| 9:46 Uhr bis 10:46 Uhr | 3 people freed from cafe in Sydney under siege. Not clear how many hostages remain. #sydnneysiege |
| 10:46 Uhr bis 11:46 Uhr | 5 people have been able to get out of Sydney cafe during hostage situation |
| 12:46 Uhr bis 13:46 Uhr | Hostages held in darkness as Sydney cafe siege passes 12th hour |
| 15:46 Uhr bis 16:46 Uhr | Sydney Siege: Hostage Taker is Iranian Refugee Man Haron Monis, Say Police Source: An Iranian refugee convicte... |
| 16:46 Uhr bis 17:46 Uhr | BBC News - Heavily armed police storm Sydney cafe ending siege by Iranian refugee gunman who took dozens hostage |

Tabelle 6.1.: Veränderung des Trends „Sydney Siege“

Trends, die ausschließlich gleiche oder sehr ähnlich formulierte Tweets enthalten, können vom System zuverlässig entfernt werden. Stichprobenartig wurden die als „Spam“ markierten Trends aus verschiedenen Tagen innerhalb der gesammelten Daten betrachtet. Dies ergab, dass diese auch wirklich als Spam gesehen werden konnten. Eine Ausnahme bildete ein Trend, welcher eine Liste von Last.fm¹ Statusmeldungen von verschiedenen Benutzern enthielt:

My Top 3 #lastfm Artists: Within Temptation (15), TV on the Radio (13) & Metric (11)

My Top 3 #lastfm Artists: Ayreon (46), Anathema (38) & Within Temptation (30)

My Top 3 #lastfm Artists: Goo Goo Dolls (2), Matchbox Twenty (1) & Jon Bon Jovi & Richie Sambora (1)

My Top 3 #lastfm Artists: Automatic Sam (10), Florence + the Machine (8) & Stevie Nicks (5)

My Top 3 #lastfm Artists: Kate Bush (3), Jethro Tull (3) & Julie Covington (3) #lastfm

Dieser Trend konnte nicht als Spam erkannt werden, da viele unterschiedliche Künstlernamen auftraten und somit die Anzahl der unterschiedlichen Wörter des Trends über der festgelegten Schwelle lag.

6.1.2 Online Clustering Ansatz

Im Online Clustering Ansatz werden Tweets mit Hilfe eines Ähnlichkeitsmaßes und einer festgelegten Schwelle bestehenden Clustern zugeordnet. Dies birgt gleichzeitig auch den größten Nachteil dieses Ansatzes. Dadurch, dass nur die textuelle Ähnlichkeit zwischen den Tweets herangezogen wird, ist es nicht möglich, Tweets mit unterschiedlichen Formulierungen zum gleichen Thema als zusammengehörig zu erkennen.

Durch Festlegung der Ähnlichkeitsschwelle kann bestimmt werden ob die Abgrenzung zwischen den Clustern sehr streng ist, was zu vielen verschiedenen Clustern führt oder umgekehrt zu wenigen Clustern, die dann einzelne Themen vermischen.

Der Online Clustering Ansatz bietet den Vorteil, dass er nicht an feste Zeitfenster gebunden ist, wie der Frequent Pattern Mining Ansatz. Da Tweets direkt verarbeitet werden, sobald sie empfangen werden, kann der Erstellungszeitpunkt eines Clusters sekundengenau festgelegt werden und die zeitliche Auflösung ist damit deutlich besser. Die Veränderung eines Trends, der durch ein Cluster repräsentiert wird, lässt sich als Funktion zwischen der Anzahl der hinzugefügten Tweets zum Cluster und der Zeit darstellen. Eine Veränderung in der Funktion spiegelt das Aufkommen und das Abklingen des Themas wider.

Die Aufteilung in Cluster alleine ist meist nicht sehr aussagekräftig. Die Anzahl der Twitterdaten wird dadurch zwar deutlich reduziert aber es wäre ein weiterer Schritt nötig, in dem Cluster entweder zusammengefasst oder gefiltert werden, wie es auch in den meisten Forschungsarbeiten der Fall ist. [5]

6.1.3 Zusammenfassung

Beim FPM Ansatz treten wenige bis gar keine Themenfragmentierungen auf. Durch die unterschiedlichen Itemsets, die einem Trend zugeordnet sind, ist es möglich, unterschiedliche Formulierungen zu erkennen, die das gleiche Thema diskutieren. So werden nahezu alle Tweets identifiziert, die zu einem Trend gehören. Im Online Clustering Ansatz ist dies nicht möglich und es treten sehr viele unterschiedliche Cluster auf, die zu einem Trend gehören.

Zusätzlich kann der FPM Ansatz Trends unterscheiden, die immer gleich oder ähnlich formulierte Tweets enthalten oder sich durch unterschiedlich formulierte Tweets auszeichnen. Auf diese Weise können viele als „Spam“ identifizierte Trends automatisch entfernt werden. Dies ist im Online Clustering Ansatz nicht möglich, weshalb die Ergebnisse dort deutlich mehr mit Spam behaftet sind.

¹ <http://www.lastfm.de>

6.2 Evaluierung der Personalisierungsfunktion

Um dem Benutzer eine zusätzliche Unterstützung zu bieten, wurde wie in Abschnitt 4.6 beschrieben eine Personalisierungsfunktion eingeführt. Diese erlaubt es, die vom System ermittelten Trends in die zwei Klassen „interessant“ und „nicht interessant“ zu unterteilen. Da die Interessantheit von Benutzer zu Benutzer unterschiedlich ist, kam nur eine Evaluierung mit mehreren Personen in Frage, deren Ablauf im weiteren beschrieben wird.

| | Erkannter Trend | Klassifizierung | |
|---|--|-----------------|-----------------|
| 1 | plot twist: bangtan is taking pictures of PH ARMYs inside the car or filming them FOR BANGTAN BOMB 😊 | Interesting | Not Interesting |
| 2 | (Warning: graphic photos) Literally just saw someone get shot at the corner of Hollywood & highland. I'm in shock. | Interesting | Not Interesting |
| 3 | *@Madonna: Even with No Light.....we're Gonna Shine like Gold in this mad mad World #ghostown ♥#rebelheart | Interesting | Not Interesting |
| 4 | Packers take 14-0 lead on Lions on Aaron Rodgers TD pass to Randall Cobb. Rodgers is helped off field with apparent injur... | Interesting | Not Interesting |
| 5 | Panthers win NFC South! Carolina destroys Atlanta, 34-3.Panthers have won 2 straight division titles. | Interesting | Not Interesting |

Abbildung 6.2.: Manuelle Annotation durch die Versuchsperson

Um herauszufinden, ob die Personalisierungsfunktion einen Anwender darin unterstützt, für ihn relevante Trends zu erkennen, wurde eine Benutzerstudie mit acht Personen durchgeführt. Nach einer kurzen Einführung in die Oberfläche wurden jeder Versuchsperson 120 verschiedene vom System ermittelte Trends vorgelegt, die dann manuell von ihnen als „interessant“ oder „nicht interessant“ markiert werden mussten. Die dazu verwendete Oberfläche ist in Abbildung 6.2 zu sehen. Im Anschluss wurden 120 weitere Trends präsentiert inklusive deren Einordnung durch den Klassifizierer (Abbildung 6.3). Die Versuchspersonen mussten nun entscheiden, ob der präsentierte Trend und das Ergebnis der Klassifizierung übereinstimmten.

| | Erkannter Trend | Klassifizierung | | |
|---|---|-----------------|---------|--------|
| 1 | Hong Kong protesters clash with police near heart of financial district via @josephjett #news | Uninteresting | Richtig | Falsch |
| 2 | Woww it is December 1st tomorrow where has the time gone ? 🤔 | Uninteresting | Richtig | Falsch |
| 3 | FACT: Roberto Soldado, Chris Smalling, Joe Cole & Glen Johnson have all scored a Premier League goal this weekend. | Interesting | Richtig | Falsch |
| 4 | GOAL Southampton 0-2 Man City (80 mins).. Frank Lampard lashes in from the edge of the box after James Milner's pass #SO... | Interesting | Richtig | Falsch |
| 5 | Today marks one year since Paul Walker died in a car crash. RIP Paul. | Uninteresting | Richtig | Falsch |

Abbildung 6.3.: Auswertung der Ergebnisse des Klassifikators

Es wurde eine Liste mit erkannten Trends über einen Zeitraum von zwei Monaten ermittelt. Aus dieser Liste wurden 240 zufällige Trends ausgewählt. Aus diesen wurde wiederum ein Trainings- und ein Testdatensatz erstellt. Diese Aufteilung wurde gewählt, damit in beiden Datensätzen ähnliche Themen auftreten können, die der Klassifizierer unterscheiden kann. Würden beide Datensätze völlig unterschiedliche Themen abbilden, könnte nur schwer eine Unterscheidung getroffen werden. Interessiert sich ein Benutzer beispielsweise für das Thema Sport und bewertet Inhalte mit „FC Barcelona“ als interessant, kann es vorkommen, dass das System „Real Madrid“ nicht als interessant einstuft, da der Klassifizierer „Real Madrid“ noch nicht mit Sport in Verbindung bringen kann.

Als Baseline wurde der Fall gewählt, dass das System über keine Personalisierungsfunktion verfügt und alle durch die Trenderkennung erkannten Trends als interessant angesehen werden. Das Ziel der Evaluierung ist es, dass eine Klassifizierung der Trends in „interessant“ und „nicht interessant“ bessere Ergebnisse liefert als die definierte Baseline.

| | Precision | Recall | F1-Score | Accuracy |
|----------------------|--------------|------------|--------------|--------------|
| Baseline | 0,369 | 1,0 | 0,517 | 0,369 |
| Mit Personalisierung | 0,897 | 0,676 | 0,728 | 0,859 |

Tabelle 6.2.: Auswertung der Benutzerevaluierung

Tabelle 6.2 enthält die Ergebnisse der Benutzerevaluierung in Form der durchschnittlichen Werte über alle Versuchspersonen. Der Precision- oder Genauigkeits-Wert gibt das Verhältnis zwischen korrekt als interessant klassifizierten Objekten und der Gesamtmenge aller als interessant klassifizierten Objekte an. Der Recall-Wert gibt die Trefferquote, also das Verhältnis der als interessant klassifizierten Objekte zu allen tatsächlich interessanten Objekten an. Die Werte in der Spalte Accuracy geben den Anteil der korrekt klassifizierten Trends an der Anzahl aller präsentierten Trends an. Beim F1-Score handelt es sich um ein kombiniertes Maß, welches sowohl den Precision- als auch den Recall-Wert berücksichtigt und als die Genauigkeit des Tests gesehen werden kann:

$$F_1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

Bei Betrachtung von Tabelle 6.2 fällt auf, dass sowohl der durchschnittliche Accuracy Wert, wie auch der durchschnittliche F1-Wert deutlich höher ist als der entsprechende Wert der Baseline. Der Recall-Wert von 1,0 aus der Baseline kommt dadurch zustande, da alle präsentierten Trends als interessant gewertet werden und somit alle tatsächlichen Trends korrekt erkannt werden. Im Gegenzug resultiert durch die hohe False Positive-Rate daraus ein niedriger Genauigkeitswert. In Anhang F findet sich eine detailliertere Übersicht über die Ergebnisse der einzelnen Versuchspersonen.

Wie zuvor angenommen, war die Einordnung der Trends in interessant oder nicht interessant von Person zu Person sehr unterschiedlich. Eine Versuchsperson hat eine sehr strenge Einteilung vorgenommen, indem nur Trends, die von persönlichem Interesse waren als „interessant“ annotiert wurden. So wurde ein Trend zum Tod von Joe Cocker vom 22. Dezember 2014 als „interessant“ und der Tod von einem Cricketspieler² vom 27. November 2014 als „nicht interessant“ eingeordnet, da die Versuchsperson die Person nicht kannte. In solchen Situationen ist das System nicht in der Lage, eine ausreichend gute Abschätzung zu treffen. Ein ähnlicher Fall ist bei einer anderen Versuchsperson aufgetreten, die Trends zum Thema Football als „interessant“ markierte aber nicht Trends zum Thema Basketball. Beide Trends enthalten ähnliche Schlagworte wie „win“, „loose“, „score“, „coach“ oder „player“, mit denen der Klassifizierer angelernt wird. Von daher ist bei der gewählten Implementierung nur eine Unterscheidung in grobe Themenkomplexe möglich. Es kann zum Beispiel gut zwischen Sport und Politik unterschieden werden aber nicht zwischen einzelnen Sportarten.

² http://de.wikipedia.org/wiki/Phillip_Hughes#Tod

7 Aktuelle Forschung

Seit dem ersten Tweet im Jahr 2006 [11] existiert eine Vielzahl an Forschung im Bereich Social Media, die sich direkt oder indirekt mit Twitter beschäftigt. Dies könnte an mehreren Faktoren liegen: Zum einen bietet Twitter mehrere APIs für einen guten und einfachen Zugriff. Zum anderen sind Tweets mit einer maximalen Länge von 140 Zeichen zu Auswertungszwecken leicht zu verarbeiten.

Twitter gibt die Anzahl der monatlich aktiven Benutzer auf ca. 284 Millionen an, von denen 80% von einem Smartphone aus den Dienst nutzen. [28] Durch die hohe Anzahl der Benutzer und die Einfachheit der Nutzung des Dienstes, ist Twitter eine Quelle, die auch immer das aktuelle Geschehen mehr oder weniger deutlich widerspiegelt.

Die Forschung beschäftigte sich in der Vergangenheit unter anderem mit der Erkennung von Nachrichten [45, 50], mit der Analyse und Vorhersage politischer Stimmungen [53], mit der Vorhersage von für den Benutzer möglicherweise interessanten Hashtags oder Themen [33], mit der Erkennung oder nachträglichen Analyse von Naturkatastrophen [49] oder mit der Analyse der Verbindungen der Benutzer untereinander durch das Aufspannen von Graphen [12].

Die vorliegende Arbeit versucht Arbeiten aus der Forschung zu betrachten, die möglichst aus den vergangenen vier Jahren stammen und sich auf die Bereiche Themen- und Trenderkennung beziehen. Da es viele Forschungsarbeiten auf diesem Gebiet gibt, wurde versucht den Fokus dadurch auf die neuere Forschung zu legen. Obwohl die Arbeiten von Allan et al. [4] und Petrović et al. [42] nicht in den ausgewählten Zeitraum fallen, sollten diese trotzdem erwähnt werden, da sie für viele aktuelle Arbeiten eine Grundlage bilden. Es wurden zwei Klassen von Ansätzen der Themenerkennung unterschieden, welche im Weiteren beschrieben werden. Beim Online Clustering Ansatz in dieser Arbeit handelt es sich um einen Document-Pivot Ansatz. Der Frequent Pattern Mining Ansatz gehört zur Gruppe der Feature-Pivot Ansätze.

7.1 Document-Pivot Ansätze

Diese Art von Ansätzen fasst Dokumente anhand eines definierten Ähnlichkeitsmaßes zu Clustern zusammen. Es existieren zwei Strategien, wie Cluster gebildet werden können:

1. Für jeden Cluster wird ein Prototyp erstellt, der alle Dokumente des Clusters möglichst gut repräsentiert. Für ein neues Dokument wird die Ähnlichkeit zu den Prototypen der Cluster bestimmt und der Cluster mit der höchsten Ähnlichkeit ausgewählt.
2. Es wird für ein neues Dokument das ähnlichste Dokument bestimmt und dann der Cluster ausgewählt, dem dieses angehört.

Bei beiden Strategien bildet ein Dokument einen neuen Cluster, wenn die Ähnlichkeit zu allen bestehenden Clustern unterhalb einer vorher definierten Schwelle liegt. Es muss eine adäquate Schwelle ausgewählt werden, die sich je nach eingesetztem Ähnlichkeitsmaß unterscheidet. Wird die Schwelle zu niedrig angesetzt, entstehen zu allgemeine Cluster, die verschiedene Themen vermischen. Wird die Schwelle zu hoch gewählt, entstehen Fragmentierungen und gleiche Themen werden in verschiedene Cluster verteilt.

Die Arbeit von Allan, Papka und Lavrenko [4] war eine der Ersten, die sich mit dem Erkennen von Ereignissen innerhalb eines Stroms von Nachrichten beschäftigte. Gearbeitet wurde mit Nachrichtenbeiträgen, die sequenziell entsprechend ihrem zeitlichen Auftreten verarbeitet wurden. Das vorgeschlagene System nutzt eine modifizierte Version des Single Pass Clusterings [17], bei dem jede eingehende Nachricht nur einmalig betrachtet wird. Ein Event wird als eine Menge von Nachrichtenartikeln definiert, die das gleiche Ereignis diskutieren. Das Ziel ist es, mit einem Artikel zu beginnen und alle anderen zu finden, die das gleiche Event diskutieren. Durch die folgenden Schritte wird entschieden, ob ein Artikel ein bereits behandeltes Event beschreibt oder ob ein neues Event vorliegt:

- Extraktion relevanter Features und Aufarbeitung des textuellen Inhalts, um eine verkürzte „Query“-Darstellung des Nachrichtenartikels zu erzeugen.
- Vergleich des Artikels mit vorher erkannten Artikeln anhand der Query-Darstellung.
- Wenn die Ähnlichkeit des Nachrichtenartikels nicht über dem definierten Schwellwert liegt, wird der Artikel als neues Event markiert.
- Ist das Gegenteil der Fall, wird der Artikel nicht als neues Event markiert.
- Optional wird der Artikel nun dem bestehenden Event zugeordnet und die Query-Darstellungen werden neu berechnet.

Das von Allan et al. beschriebene Vorgehen ist in mehreren Forschungsarbeiten angewendet und auf Twitter übertragen worden. Hierzu zählen unter anderem die Arbeiten von Becker et al. [5] und Petrović et al. [42].

Beyond Trending Topics: Real-World Event Identification on Twitter

Das in [5] von Becker et al. beschriebene System arbeitet in zwei Schritten. Zu Beginn werden aus dem Strom der Twiternachrichten einzelne Cluster gebildet. Diese werden anhand eines Klassifikators in *Real-World Events* - Themen mit Bezug zu einem Ereignis aus der wahren Welt - und *Twitter-Centric Topics* - Themen die außerhalb von Twitter wenig bis keine Bedeutung haben - unterteilt.

Für die Erstellung der Cluster wird ein Online Clustering verwendet, welches ähnlich zu dem aus [4] arbeitet. Dies bietet bei einem Strom von Nachrichten den Vorteil, dass Cluster in Echtzeit gebildet werden können und im Vorfeld kein Wissen über die Anzahl der Cluster notwendig ist, wie es bei klassischen Clustering-Verfahren manchmal vorausgesetzt wird. Um einen eingehenden Tweet einem der Cluster zuzuordnen zu können, wird eine Centroid-Darstellung des Clusters generiert, die aus den durchschnittlichen Häufigkeiten aller Begriffe über alle Nachrichten des Clusters besteht.

Um zu bestimmen, ob es sich um einen Cluster mit Bezug zu einem Real-World Event handelt, haben Becker et al. eine Reihe von Features definiert:

- *Temporal Features*: Beschreiben Veränderungen der Häufigkeiten einzelner Begriffe und das Wachstum der Nachrichten eines Clusters über die Zeit hinweg.
- *Social Features*: Beschreiben die Arten der Interaktion die in den Tweets eines Clusters verwendet wurden (Retweets, User Mentions, Replies). Eine hohe Anzahl an Retweets könnte laut Becker et al. auf Informationen hindeuten, die kein Real-World Ereignis widerspiegeln.
- *Topical Features*: Beschreiben, wie unterschiedlich die Nachrichten eines Clusters zu dessen Schwerpunkt sind. Der Gedanke dabei ist, dass Tweets über ein Event eine höhere Ähnlichkeit zueinander aufweisen, wobei Diskussionen über Alltagsthemen sehr unterschiedlich sind. Diese These steht im Widerspruch zu den Beobachtungen die in dieser Arbeit gemacht wurden.

-
- *Twitter-Centric Features*: Beschreiben, ob es sich bei einem Cluster um ein Thema handelt, welches auf den Kosmos von Twitter beschränkt ist. Hier wird als Indikator die Analyse der Verwendung von Hashtags herangezogen.

Die Features wurden benutzt, um einen Klassifikator auf der Basis einer Support Vector Machine zu trainieren. Als Baseline diente ein Naive Bayes Klassifikator, welcher anhand des textuellen Inhalts der Cluster eine Klassifizierung in Event- oder Nicht-Event-Cluster vornahm.

Streaming First Story Detection with Application to Twitter

First Story Detection oder auch *New Event Detection* beschäftigt sich damit, die erste Nachricht innerhalb eines Datenstroms zu erkennen, die ein bestimmtes Ereignis diskutiert. Petrović, Osborne und Lavrenko [42] beschäftigen sich in ihrer Arbeit damit, wie dieses Problem auf Twitterdaten übertragen werden kann.

Das von ihnen vorgeschlagene System arbeitet wie die zwei zuvor beschriebenen Systeme aus [4] und [5], indem es nur einen Lauf über die bis dahin erkannten Events benötigt und direkt entscheidet, ob es sich um ein neues oder ein bestehendes Event handelt. Durch den Einsatz von Locality-Sensitive Hashing ist es möglich, dass das System effizient auf den Streaming Daten von Twitter arbeitet. Es wird für jeden eingehenden Tweet eine Anzahl an Kandidatenpaaren gebildet, aus denen dann das ähnlichste Dokument bestimmt wird. Die Autoren geben an, dass dies alleine zu keinen guten Ergebnissen führt. Daher wird zusätzlich eine *Variance Reduction Strategie* eingeführt. Kann mit Hilfe des Locality-Sensitive Hashings kein ausreichend ähnliches Dokument ermittelt werden, wird ein Tweet noch zusätzlich mit einer Liste der 2.000 neuesten Dokumente verglichen.

Petrović et al. beschreiben die so entstehenden Gruppen von Tweets als Thread. Pro Zeitintervall werden nur die am schnellsten wachsenden Threads ausgegeben, da davon ausgegangen wird, dass wenn etwas Bedeutendes passiert, viele Leute darüber diskutieren.

Zur Anordnung der ermittelten Threads wurden verschiedene Strategien verwendet. Eine zufällige Anordnung, eine Anordnung nach der Anzahl von Tweets und eine Anordnung nach unterschiedlichen Benutzern innerhalb einer Threads führte zu keinen signifikanten Unterschieden. Eine Anordnung nach Anzahl der unterschiedlichen Benutzer, bei der Threads mit einem Entropie-Wert kleiner 3,5 ans Ende der Liste gesetzt wurden, führte zu den besten Ergebnissen.

Die Autoren kommen zu dem Schluss, dass das vorgeschlagene System in der Lage ist durch konstanten Speicherverbrauch und konstanter Bearbeitungszeit große Mengen von Twitterdaten über einen längeren Zeitraum zu verarbeiten.

Clustering Microtext Streams for Event Identification

Yin beschreibt in seiner Arbeit [57] ein Verfahren, welches im Strom der Twitterdaten sogenannte Event-Based Cluster identifiziert. Er legt zu Beginn drei Anforderungen an sein System fest: Es soll möglich sein, eine große Menge an Twitterdaten zu verarbeiten. Zeitliche Informationen sollen in den Prozess des Clusterings einfließen, da es wahrscheinlicher ist, dass Nachrichten die zeitlich nicht zu weit auseinander sind ein gleiches Thema beschreiben, als Nachrichten, die zeitlich sehr weit auseinander liegen. Außerdem soll das System Cluster zusammenfassen können, falls diese ähnliche Inhalte beschreiben.

Der beschriebene Ansatz ist in zwei Phasen unterteilt. In der *Online Discovery Phase* werden Cluster inkrementell gebildet, ähnlich wie bei den vorher beschriebenen Document-Pivot Verfahren. Jede Nachricht wird nur einmal betrachtet und es wird entschieden, ob die Nachricht zu einem bestehenden Cluster gehört oder ein neues Ereignis abbildet. Die so entstehenden Base Cluster enthalten jeweils eine Menge von ähnlichen Tweets. Zur Zuordnung wird ein kombiniertes Ähnlichkeitsmaß verwendet, welches einen neuen Tweet und die bestehenden Cluster anhand der Kosinus-Ähnlichkeit und einem Time Similarity Maß vergleicht. Letzteres berechnet sich aus dem zeitlichen Abstand eines Tweets und eines Clusters.

Neue Tweets werden nur mit einer Liste aktiver Cluster verglichen. Wird eine längere Zeit zu einem Cluster kein neuer Tweet hinzugefügt, wird er als inaktiv angesehen und von der Liste entfernt. Dieses Vorgehen wurde auch im Clustering Ansatz umgesetzt, der in dieser Arbeit beschrieben wurde.

In der *Offline Cluster Merging Phase* wird versucht, die vorher generierten Base Cluster in sogenannte Event-Based Cluster zusammenzufassen. Benutzer, die über ein gleiches Thema twittern, verwenden meist unterschiedliche Formulierungen. Dies ist ein Problem bei Document-Pivot Ansätzen und führt dort zu Themenfragmentierungen. Durch das nachträgliche Zusammenfassen der Base Cluster versucht Yin die Menge der erkannten Ereignisse zu verbessern. Vor und nach jedem Zusammenfassen einzelner Cluster wird ein durchschnittlicher Inter-Cluster Similarity Wert gebildet, welcher Auskunft darüber gibt, wie gut die Cluster voneinander getrennt sind. Es werden solange Cluster zusammengefasst, bis die Veränderung dieser Wertes oberhalb einer Schwelle liegt.

Der Autor nennt die vom Ansatz generierten Ergebnisse Event-Based Cluster. Es wird aber nicht genau definiert, was unter einem Event zu verstehen ist. In der Evaluierungsphase wird ein Datensatz verwendet, welcher ausschließlich aus Twitterdaten mit Bezug zu Ereignissen aus der echten Welt besteht. Wie gut das System von Yin mit allgemeinen Twitterdaten funktioniert wäre eine interessante Erweiterung der Arbeit.

7.2 Feature-Pivot Ansätze

Feature-Pivot Ansätze versuchen Begriffe zusammenzufassen, indem das gemeinsame Auftreten analysiert wird. Ergebnisse werden durch Worte oder Gruppen von Worten repräsentiert, welche ein Thema innerhalb einer Menge von Dokumenten beschreiben.

A soft frequent pattern mining approach for textual topic detection

Petkos et al. [41] beschreiben in ihrer Arbeit einen *Soft Frequent Pattern Mining* Ansatz. Sie argumentieren, dass die normale Anwendung von Frequent Pattern Mining zwar Themen in Form von gemeinsam auftretenden Begriffen liefert aber hierbei zu unflexibel vorgegangen wird. Es wird verlangt, dass alle Begriffe in einem Frequent Itemset über einer vorher definierten Supportschwelle liegen. Beim Ansatz von Petkos et al. soll es ermöglicht werden, dass auch Mengen als ein zusammenhängendes Thema erkannt werden, die nicht häufige Begriffe enthalten.

Der Ausgangspunkt ist eine Menge von Twitterdaten eines bestimmten Zeitfensters für die eine Menge von Themen ermittelt werden soll. Der beschriebene Ansatz arbeitet in drei Phasen. Einer *Term Selection Phase*, einer *Cooccurrence-vector formation Phase* und einer *Post-processing Phase*.

In der *Term Selection Phase* wird ein Referenzkorpus herangezogen, welcher aus zufällig gesammelten Twitterdaten besteht. Für jeden Begriff wird die Wahrscheinlichkeit des Auftretens innerhalb des Referenzkorpus berechnet. Für jeden Begriff aus den Twitterdaten des aktuell betrachteten Zeitfensters wird ebenfalls die Auftrittswahrscheinlichkeit berechnet. Diese wird dann durch die Wahrscheinlichkeit des Auftretens im Referenzkorpus geteilt. Dadurch erhält man das Verhältnis zwischen dem aktuellen Auftreten und dem Auftreten im Referenzkorpus. Der Wert soll einen Hinweis darauf geben, ob der jeweilige Begriff im aktuellen Zeitintervall gehäuft auftritt oder ob es sich um einen themenneutralen Begriff, wie „people“ oder „day“ handelt. Eine Auflistung nach diesem Verhältnis liefert die Begriffe, die zum Bilden der Themen verwendet werden.

Die *Cooccurrence-vector formation Phase* bildet den Kern des vorgestellten Ansatzes. Für jeden der in der vorherigen Phase ermittelten Terme t wird ein Vektor D_t der Länge n gebildet, wobei n die Anzahl aller Dokumente beschreibt. Ein Eintrag des Vektors gibt an, ob der Begriff im jeweiligen Dokument enthalten ist. Anschließend wird von jedem Term t ausgehend eine Menge S gebildet, welche eines der späteren Themen repräsentiert. Hierzu wird ein Vektor D_S ebenfalls mit der Länge n gebildet. Hier gibt ein Eintrag des Vektors an, wie viele Terme aus S in dem jeweiligen Dokument enthalten sind. Eine Menge S wird sukzessive um einzelne Terme erweitert, indem die Kosinus-Ähnlichkeiten der Vektoren

D_s und D_t berechnet werden. Liegt diese über einer definierten Schwelle wird der jeweilige Term in die Menge S aufgenommen. Da die Themen ausgehend von einzelnen Termen gebildet werden, ist es möglich, dass die resultierende Menge an Themen Duplikate enthält. In der *Post-Processing Phase* werden diesen am Ende entfernt.

Detecting Newsworthy Topics in Twitter

Van Canneyt et al. [55] verwenden in ihrer Arbeit den Begriff „Newsworthy Topic“ ohne genau zu definieren, welche Arten von Ereignissen damit gemeint sind. Aus dem Kontext der Arbeit geht hervor, dass es sich um Themen aus den Nachrichtenwelt handelt. Es werden beispielsweise die Ukrainekrise aber auch das Thema Bitcoins aufgeführt.

Ziel der Arbeit ist das Erkennen von „Newsworthy Topics“ innerhalb des Datenstroms von Twitter. Hier wird mit festen Zeitintervallen gearbeitet für die jeweils Themen erkannt werden. Der vorgeschlagene Ansatz arbeitet in vier Schritten:

1. *News Publisher Detection*: Ein Klassifizierer wurde mit 10.000 Benutzern von Twitter trainiert, welche zuvor manuell in „News Publisher“ oder „Other“ unterteilt wurden. Aus allen Benutzern, die im aktuellen Zeitintervall einen Tweet erstellt haben, werden mit Hilfe des Klassifizierers relevante Benutzer ermittelt. Es werden dann nur noch Tweets dieser Benutzer zur Erstellung der Themen berücksichtigt.
2. *Topic Detection*: Die Tweets werden in eine Vektor-Darstellung umgewandelt. Es werden bei der Gewichtung der Wörter vorherige Zeitintervalle miteinbezogen. Eigennamen und Verben erhalten eine höhere Gewichtung als andere Wörter. Anschließend werden durch Einsatz des DBSCAN Algorithmus [14] Cluster gebildet.
3. *Topic Ranking*: Aus den Clustern werden jene ermittelt, die ein Thema aus der Nachrichtenwelt beschreiben. Hierzu wurde ein Klassifizierer mit 116 Clustern trainiert, indem manuell zwischen „newsworthy“ und „not newsworthy“ unterschieden wurde. Die identifizierten Cluster werden dann anhand der ermittelten Wahrscheinlichkeit sortiert, dass es sich um einen relevanten Cluster handelt.
4. *Topic Enrichment*: Die ermittelten Themen werden mit einer Überschrift, einer Liste von Schlagwörtern, einer Liste von Tweets und einer Liste von Bildern versehen.

Die Autoren haben den vorgestellten Ansatz nicht evaluiert aber führen verschiedene generierte Ergebnisse auf. Sie beschreiben, dass bei unterschiedlichen Formulierungen Fragmentierungen von Themen auftreten. Wenn im ersten Schritt Benutzer fälschlicherweise als „News Publisher“ identifiziert werden, kann es vorkommen, dass auch nicht relevante Themen in der Ergebnismenge auftauchen.

7.3 Sonstige Forschung

Die folgenden zwei Arbeiten können thematisch nicht den Document-Pivot oder den Feature-Pivot Ansätzen zugeordnet werden. Inhaltlich sind sie trotzdem interessant, weshalb sie im Weiteren betrachtet werden.

Biber no more: First Story Detection using Twitter and Wikipedia

Ein Schlüsselproblem bei der Eventerkennung auf Twitterdaten ist die große Anzahl an Ereignissen, die für niemanden von Interesse sind oder die Dinge aus dem Kosmos von Twitter beschreiben. In ihrer Arbeit beschäftigen sich Osborne et al. [38] mit der Frage, wie man die Anzahl dieser Ereignisse sinnvoll

reduzieren kann. Dafür wird untersucht, ob Wikipedia als Informationsquelle herangezogen werden kann. Um Events innerhalb des Stroms von Twitter zu erkennen, wird der Ansatz von Petrović, Osborne und Lavrenko [42] verwendet.

In einem ersten Schritt werden die Anzahlen der Aufrufe aller englischen Wikipedia-Artikeln ausgewertet. Eine Stunde enthält ungefähr 10 Millionen Aufrufe. Für jeden Artikel wird eine Statistik erstellt, welche die Anzahl der stündlichen Aufrufe enthält. Mit einem Ausreißertest nach Grubbs [21] werden die Artikel ermittelt, welche innerhalb einer Stunde eine besonders hohe Veränderung in den Zugriffen aufweisen. Für ein Fenster von 48 Stunden konnten ca. 625.000 Wikipedia-Artikel ermittelt werden, die Ausreißer aufwiesen. Diese bilden zusammen mit dem Zeitpunkt des Ausschlags einen Strom von Wikipedia-Artikeln.

Die Annahme ist, dass ein relevantes Ereignis sich mit einem Ausreißer innerhalb des Wikipedia Streams in Verbindung bringen lässt. Nicht relevante Themen, können hingegen keinem der Wikipedia-Artikel zugeordnet werden. Ein Beispiel, welches sich mit dem Wikipedia-Stream in Verbindung bringen lässt, war der Tod von Amy Winehouse. Ihr Tod führte dazu, dass sehr viele Menschen auf Wikipedia nach Informationen über die Sängerin suchten.

Die Autoren kamen zu dem Ergebnis, dass Twitter und Wikipedia gemeinsam genutzt werden können, um Ergebnisse von Eventerkennungssystem zu verbessern. Das Widerspiegeln in den Zugriffszahlen von Wikipedia scheint dabei aber um ungefähr zwei Stunden verzögert zu sein. Aus diesem Grund eignet sich das vorgeschlagene Vorgehen auch nur eingeschränkt, um Ereignisse in Echtzeit zu erkennen.

Detecting Life Events in Feeds from Twitter

Die Arbeit von Di Eugenio, Green und Subba [10] beschäftigt sich mit dem Erkennen sogenannter „Life Events“. Damit beschreiben die Autoren Ereignisse, die das Leben einer Person beeinflussen und die über soziale Medien geteilt werden. Hierzu wurden als Beispiele eine Hochzeit, die Geburt eines Kindes oder eine berufliche Veränderung aufgeführt.

In der Arbeit werden exemplarisch die Ereignisse „Heirat“ und „Veränderung des Arbeitsverhältnisses“ betrachtet. Die Autoren haben dazu mit Hilfe der Streaming API einen eigenen Corpus an Twitterdaten aufgebaut. Aus der Gesamtmenge wurden anhand bestimmter Schlüsselwörter Tweets zu beiden Themen extrahiert und nicht relevante Nachrichten verworfen.

Zum Thema Arbeitsverhältnis wurden die Begriffe „job“, „laid off“, „interview“ und „job offer“ gewählt und davon ausgegangen, dass eine Filterung nach diesen das Thema ausreichend gut repräsentiert. Beim Thema Heirat wurde ähnlich verfahren, mit dem Unterschied, dass Schlüsselwörter anhand von Webseiten ermittelt wurden, die der Domäne zugeordnet sind. Es wurden aus verschiedenen Webseiten wie www.brides.com oder www.weddings-magazine.com Inhalte extrahiert, aus denen dann die häufigsten Begriffe ermittelt wurden. Es wurde dann davon ausgegangen, dass diese das Thema besser beschreiben. Bei den ermittelten Begriffen handelte es sich um „engaged“, „married“ und „wedding“.

Aus diesem Schritt resultierte eine Menge von ungefähr 4.000 Tweets. Jeder Tweet wurde mit einer externen Bibliothek auf dessen Rechtschreibung geprüft und ein Ranking erstellt, an dessen Ende sich die Tweets mit den meisten Fehlern befanden. Die ersten 2.250 Tweets der entstandenen Liste wurden manuell verschiedenen Klassen zugeordnet. Für das Thema Arbeit wurden die Klassen „hat Einfluss auf den Ersteller“, „hat Einfluss auf eine andere Person“, „Relevant aber allgemeine Aussage“ und „keine Relevanz zum Thema“, entschieden. Das Thema Heirat wurde in sieben Klassen unterteilt.

Mit Hilfe der Software Weka [24] wurden verschiedene Klassifizierungsverfahren mit den annotierten Tweets getestet. Im ersten Schritt wurde ein Unigram-Modell verwendet, welches die Tweets als Liste von Worttoken darstellte. Zusätzlich wurden Variationen getestet, die verschiedene Natural Language Processing Techniken einsetzen. Gegen die Erwartung der Autoren hat sich gezeigt, dass die Präzision bei der Verwendung des einfachen Unigram-Modells am höchsten war.

Bei der Entwicklung des FPM und des Clustering Ansatzes hat sich gezeigt, dass die Ergebnisse sehr weit gefächert sind und es manchmal besser wäre, eine thematische Eingrenzung vorzunehmen. Daher

wäre es interessant, das Prinzip und die Vorgehensweise der Arbeit von Eugenio et al. auf andere Themen zu übertragen.

7.4 Zusammenfassung der aktuellen Forschung

Ein Problem bei allen Ansätzen ist, dass man kein Gefühl dafür bekommt, wie gut oder schlecht die Ansätze funktionieren. Dadurch, dass Daten von Twitter schwierig zu evaluieren sind, kann man schwer einschätzen, wie gut oder schlecht ein Ansatz im Vergleich zu anderen arbeitet.

Probleme welche im Rahmen dieser Arbeit aufgetreten sind, finden sich teilweise auch in den Arbeiten aus der Forschung wieder. In vielen Arbeiten wird beispielsweise beschrieben, dass Fragmentierungen oder Duplikate einzelner Themen auftreten.

Insgesamt existieren bisher wenige Arbeiten, die sich mit der Analyse und der Veränderung von Häufigkeiten beschäftigen. Es existiert keine wissenschaftliche Arbeit, die sich direkt mit der Anwendung von Assoziationsanalyse auf Twitterdaten beschäftigt. In der Arbeit von Petkos et al. [41] wird dies zwar oberflächlich beschrieben aber es handelt sich dabei nicht um das eigentliche Thema. Daher bietet dies sicher Raum für weitere Forschung.

8 Zusammenfassung

Im ersten Schritt dieser Arbeit wurde betrachtet, wie und welche Daten von Twitter bezogen werden können. Es wurde die Entscheidung getroffen, mit dem Sample Endpoint der Streaming API zu arbeiten, welcher eine zufällige Stichprobe des aktuellen Verkehrs auf Twitter liefert. Außerdem wurde die Einschränkung getroffen, nur mit Tweets in englischer Sprache zu arbeiten. So ist ein Datenbestand von Tweets über einen Zeitraum von drei Monaten entstanden, welcher ungefähr 18 GB groß ist. Unter Berücksichtigung der kompletten gesammelten Daten wurde eine Liste mit Wortpaaren erstellt. Die Wortpaare geben an, welche Wörter oft in Kombination innerhalb einzelner Tweets aufgetreten sind. Die entstandene Liste kann unter Umständen auch für andere Forschungsarbeiten von Interesse sein, da sie die Themen widerspiegelt, welche über den kompletten Zeitraum von drei Monaten dauerhaft präsent waren.

Es wurden zwei aus der Forschung motivierte Ansätze implementiert und getestet. Aus einem davon entwickelte sich der letztendlich vorgeschlagene Ansatz. Die Twitterdaten werden dabei in Zeitfenster aufgeteilt, auf die jeweils die Assoziationsanalyse angewendet wird. Die resultierenden Frequent Itemsets werden durch eine Zuordnung zwischen Itemsets und Tweets unter Verwendung eines Graphen zu verschiedenen Trends gruppiert. Im Anschluss werden einzelne Zeitfenster verbunden, um festzustellen, ob ein Trend neu aufgetreten ist oder bereits im Vorfeld vorkam. Auf diese Weise kann die Entwicklung eines Trends auch über einen längeren Zeitraum verfolgt werden.

Viele Ansätze aus der Forschung müssen sich in diesem Zusammenhang mit der Fragmentierung einzelner Themen auseinandersetzen. Dabei werden von einem System verschiedene Ergebnisse generiert, die zu einem Themenkomplex gehören. Das neuartige Vorgehen in dieser Arbeit sorgt dafür, dass Frequent Itemsets, die zu einem gleichen Thema gehören, als zusammengehörig erkannt werden und so wenige bis keine Themenfragmentierungen auftreten.

Eine Schwierigkeit stellte die hohe Anzahl an Spam innerhalb der Tweets dar. Durch die Arbeit mit der Stichprobe der Twitterdaten war die Menge dieser Tweets besonders hoch. Die vom implementierten System generierten Trends werden deshalb automatisch auf Spam gefiltert. Dies wird anhand der Unterschiedlichkeit der Tweets eines Trends zueinander vorgenommen. Zusätzlich wurde eine Personalisierungsfunktion integriert. Sie erlaubt es dem Anwender das System zu trainieren, sodass dieses für den Benutzer relevante Trends hervorhebt.

Durch eine Benutzerstudie mit verschiedenen Versuchspersonen wurde gezeigt, dass im Gegensatz zur Ausgangssituation, in der dem Anwender alle generierten Trends präsentiert werden, die Personalisierungsfunktion zu einer signifikanten Verbesserung führt, indem nur die für den Anwender interessanten Trends präsentiert werden.

8.1 Ausblick

Es gibt eine Vielzahl an Möglichkeiten, die Daten von Twitter zu verarbeiten und zu analysieren. Dies spiegelt sich auch in den vielen Forschungsarbeiten wider, die es zum Thema Twitter gibt. So sind im Laufe dieser Arbeit immer wieder Ideen aufgekommen, die wieder verworfen wurden, weil sie teilweise in andere Richtungen gehen oder weil der zeitliche Rahmen zu eng war, um sich damit ausführlicher zu beschäftigen.

Zur Bildung der Frequent Itemsets wurde in dieser Arbeit wie beschrieben der FP-Growth Algorithmus eingesetzt. Nach einer gewissen Zeit stellte sich heraus, dass das System nur mit der Menge der maximalen Frequent Itemsets arbeiten wird. Hierzu gibt es spezielle Algorithmen wie MAFIA von Burdick et al. [7]. Da die Laufzeit des Systems keine zentrale Rolle gespielt hat, war es nicht nötig einen anderen Algorithmus zu verwenden und der Umweg über die Filterung der Ergebnisse des FP-Growth Algorithmus war ausreichend.

An der Arizona State University wurde ein Analyseprogramm namens TweetTracker [30] entwickelt. Dieses ermöglicht es, interaktiv durch einen Bestand an Twitterdaten zu navigieren und die Daten individuell zu analysieren, indem eigene Filter definiert werden oder konkrete Zeiträume angegeben werden können. Eine interaktive Selektion und Auswertung der Daten ist mit dem in dieser Arbeit erstellten System bisher nicht möglich. Ein Feature, bei dem Zeiträume über eine Benutzeroberfläche ausgewählt werden könnten und die dazugehörigen Trends per Knopfdruck erstellt werden würden, wäre eine gute Verbesserung seitens der Usability.

TweetTracker erlaubt es, benutzerdefinierte Jobs zu erstellen, die aus einer Liste von Keywords, Geodaten oder Twitter Benutzern bestehen. Das System sammelt dann im Hintergrund gezielt Tweets zu den angegebenen Kriterien. Mit dieser Herangehensweise, unter Verwendung des Filter Endpoints, kann die Flut von nicht relevanten Tweets reduziert werden. Man wäre so auch in der Lage mehr Daten zu einem konkreten Thema zu erhalten als es bei der Verwendung des Sample Endpoints möglich ist.

Viele Arbeiten aus der Forschung beschäftigen sich damit, Tweets auch inhaltlich zu analysieren. Hierzu werden Natural Language Processing Techniken, wie Part of Speech Tagging [19] oder Named Entity Recognition [47] eingesetzt, um Satzstrukturen zu analysieren oder Personen und Orte zu extrahieren. Dieser Bereich der Forschung wurde in dieser Arbeit nicht berücksichtigt und der vorgeschlagene Ansatz arbeitet rein mit der Auswertung von Häufigkeiten. Es wäre aber auch eine vielversprechende Option, zu überlegen, wie man mit Natural Language Processing die generierten Ergebnisse verbessern könnte.

A Twitter-spezifische Stopwordliste

Stopwords

Im Laufe der Arbeit ergab sich die Liste mit folgenden Wörtern, die sich als Twitter-spezifische Stopwords herausstellten. Zusätzlich zu der Filterung der sprachbezogenen Stopwords wurden alle Tweets, die eines der folgenden Wörter enthielten entfernt, was die Resultate der implementierten Ansätze deutlich verbesserte.

followers, follower, follow, followed, followersrt, tweet, twitter, unfollowers, unfollower, unfollow, unfollowed, backfollow, followersrt, rtsgain, rts, rt, #rt, #retweet, retweet, #retweets, retweets

Häufige Wortkombinationen

Des Weiteren stellte sich heraus, dass es eine immer wiederkehrende Anzahl an Begriffen gibt, die oft in Kombination auftreten. So gibt es zum Beispiel täglich Tweets mit den Begriffen „happy“ und „birthday“. Bei folgenden Wortkombinationen handelt es sich um die Top 100 der Paare über den ganzen Datenbestand aller gesammelter Tweets über drei Monate:

(stats, today), (birthday, happy), (#mtvstars, direction), (gaga, lady), (swift, taylor), (#mtvstars, lady), (christmas, merry), (#mtvstars, gaga), (bieber, justin), (#mtvstars, coldplay), (photo, posted), (#mtvstars, taylor), (#mtvstars, swift), (happy, year), (del, lana), (@youtube, video), (#mtvstars, lana), (#mtvstars, del), (del, rey), (lana, rey), (#mtvstars, rey), (happy, love), (facebook, posted), (ariana, grande), (lana, taylor), (del, taylor), (lana, swift), (del, swift), (good, morning), (rey, taylor), (rey, swift), (minaj, nicki), (azalea, iggy), (#mtvstars, justin), (facebook, photo), (#mtvstars, beyonce), (#mtvstars, bieber), (pack, starter), (brown, chris), (#mtvstars, paramore), (@harry_styles, love), (#mtvstars, iggy), (#mtvstars, ariana), (people, today), (#mtvstars, azalea), (#mtvstars, grande), (person, today), (hope, love), (#mtvstars, nicki), (#mtvstars, minaj), (person, stats), (day, happy), (day, love), (people, stats), (day, good), (day, today), (love, people), (beyonce, nicki), (love, world), (artist, year), (day, hope), (beyonce, minaj), (life, love), (#mtvstars, brown), (video, watch), (update, weather), (#mtvstars, chris), (#amas, artist), (good, luck), (hemmings, luke), (day, great), (chance, win), (@real_liam_payne, love), (hate, people), (high, school), (birthday, love), (nicole, scherzinger), (#mtvstars, nicole), (#mtvstars, scherzinger), (christmas, love), (#amas, year), (coins, gold), (guys, love), (#mtvstars, avicii), (brown, nicki), (brown, minaj), (coins, collected), (collected, gold), (avicii, coldplay), (beyonce, brown), (happy, hope), (cyrus, miley), (hey, love), (life, people), (beyonce, paramore), (chris, nicki), (#mtvstars, mix), (chris, minaj), (beyonce, chris), (lady, paramore)

B Häufigkeitsanalyse Keywords

| Aktuelle Stunde | Vorheriger Tag | Verhältnis | Keyword |
|-----------------|----------------|------------|---------------------|
| 230 | 0 | 230.0 | #lionheart |
| 230 | 0 | 230.0 | rise!jake |
| 185 | 0 | 185.0 | cardigan |
| 181 | 0 | 181.0 | hopped |
| 151 | 0 | 151.0 | @caspar_lee |
| 124 | 0 | 124.0 | @towlerluke |
| 115 | 0 | 115.0 | @theweeknd |
| 98 | 0 | 98.0 | picking |
| 96 | 0 | 96.0 | @vevo |
| 79 | 0 | 79.0 | #luketowersoccer6 |
| 76 | 0 | 76.0 | @barsandmelody |
| 75 | 0 | 75.0 | cam |
| 73 | 0 | 73.0 | @madisonellebeer |
| 68 | 0 | 68.0 | beth |
| 67 | 0 | 67.0 | recap |
| 66 | 0 | 66.0 | @danisnotonfire |
| 61 | 0 | 61.0 | @katyperry |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| 237 | 539 | 0.43970317 | miller |
| 245 | 624 | 0.3926282 | mtv |
| 257 | 688 | 0.3735465 | #votejakemiller |
| 207 | 557 | 0.37163374 | lax |
| 313 | 1017 | 0.30776796 | @themattspinosa |
| 204 | 711 | 0.28691983 | plane |
| 465 | 1669 | 0.27860993 | @michael5sos |
| 264 | 970 | 0.27216494 | deserves |
| 276 | 1059 | 0.26062322 | @luke5sos |
| 251 | 970 | 0.2587629 | hemmings |
| 161 | 631 | 0.25515056 | lyrics |
| 132 | 544 | 0.24264705 | demilovato |
| 328 | 1398 | 0.23462088 | @camerondallas |
| 195 | 956 | 0.2039749 | clifford |
| 177 | 884 | 0.20022625 | @5sos |
| 364 | 1956 | 0.18609408 | luke |
| 91 | 528 | 0.17234848 | screen |
| 815 | 4866 | 0.1674887 | stats |
| 274 | 1705 | 0.16070381 | #rollersmusicawards |

C Häufigkeitsanalyse n-Tupel

2-Tupel:

| Häufigkeit | Tupel |
|------------|--------------------------------|
| 271 | #caniffollowme, taylor |
| 272 | college, football |
| 279 | dream, realize |
| 280 | prove, wrong |
| 282 | sam, smith |
| 290 | dream, true |
| 295 | #talktomematt, matt |
| 298 | galaxy, samsung |
| 300 | @taylorcaniff, taylor |
| 310 | life, save |
| 311 | #movie, #music |
| 324 | @themattspinosa, matt |
| 333 | hard, work |
| 333 | great, time |
| 333 | feet, ground |
| 338 | bad, feel |
| 338 | download, free |
| 370 | cream, ice |
| 374 | girl, happiest |
| 380 | school, tomorrow |
| 385 | morning, sunday |
| 393 | bless, god |
| 397 | app, download |
| 410 | night, saturday |
| 425 | high, school |
| 443 | girl, steal |
| 461 | call, credit |
| 492 | media, social |
| 506 | gain, thismust |
| 542 | long, time |
| 554 | good, time |
| 586 | live, stream |
| 627 | good, luck |
| 690 | food, harvested |
| 889 | good, night |
| 987 | #talktomematt, @themattspinosa |
| 1009 | #caniffollowme, @taylorcaniff |
| 1597 | good, morning |

3-Tupel:

| Häufigkeit | Tupel |
|------------|---|
| 251 | completed, game, quest |
| 262 | @jacobwhitesides, buy, words |
| 276 | coming, health, important |
| 306 | berlin, fall, wall |
| 398 | leo, meaning, seeking |
| 406 | #mybandquestion, #myfourquestion, #mytourquestion |
| 619 | #mpoints, earning, rewards |
| 1420 | coins, collected, gold |
| 1927 | facebook, photo, posted |

4-Tupel:

| Häufigkeit | Tupel |
|------------|--|
| 82 | #fourhangout, @onedirection, excited, wait |
| 358 | day, potential, strength, today |
| 363 | dis, expect, friends, today |

D Skript zur Ermittlung der häufigen Paare

Mit diesem Apache Pig! [16] Skript wurde die Liste der häufigen Paare über den Gesamtzeitraum der gesammelten Daten aus Anhang A erstellt.

```
1 register './libs/jsonic-1.2.0.jar';
2 register './libs/langdetect.jar';
3 register './libs/myudfs.jar';
4
5 DEFINE pp_tweet(in_relation, id_field, text_field) RETURNS out_relation {
6
7     /* Filter out non english tweets */
8     tweets = FILTER $in_relation BY myudfs.LanguageFilter($text_field);
9
10    /* Filter out all Tweets that contain words like 'porn' */
11    tweets = FILTER $in_relation BY myudfs.PornFilter($text_field);
12
13    /* Convert all text to lowercase */
14    tweets = FOREACH tweets GENERATE $id_field, myudfs.LowerEvalFunc($text_field) AS
15        $text_field;
16
17    /* Return result */
18    $out_relation = FOREACH tweets GENERATE $id_field, $text_field;
19};
```

Listing D.1: preprocessing-tweet-macro.pig

```
1 register './libs/myudfs.jar';
2
3 DEFINE pp_word(in_relation, id_field, word_field) RETURNS out_relation {
4
5     /* Filter Special Characters */
6     words = FOREACH $in_relation GENERATE $id_field, myudfs.SpecialCharEvalFunc($word_field
7         ) as $word_field;
8
9     /* Convert slang to english */
10    words = FOREACH words GENERATE $id_field, myudfs.SlangEvalFunc(word) as $word_field;
11
12    /* Filter out Stopwords */
13    words = FILTER words BY myudfs.StopwordFilter($word_field);
14
15    /* Remove Empty words */
16    words = FILTER words BY word != '';
17
18    /* Return result */
19    $out_relation = FOREACH words GENERATE $id_field, $word_field;
20};
```

Listing D.2: preprocessing-word-macro.pig

```

1 import 'preprocessing-word-macro.pig';
2 import 'preprocessing-tweet-macro.pig';
3 register './myudfs.jar';
4 register './datafu-1.2.0.jar';
5 define UnorderedPairs datafu.pig.bags.UnorderedPairs();
6
7 /* Load Tweets from Directory */
8 twitterData = LOAD 'tweets-2015-01-10/' USING PigStorage(';')
9             AS (id, timestamp, userId, retweetId:bigint, text);
10
11 /* Preprocessing Step 1 */
12 twitterData = pp_tweet(twitterData, 'id', 'text');
13
14 tweets = FOREACH twitterData GENERATE
15         id,
16         myudfs.TwoTokenize(LOWER(text)) AS words;
17
18 wordsByTweet = FOREACH tweets {
19     orderedWords = DISTINCT words;
20     GENERATE
21         id AS tweet,
22         orderedWords AS words;
23 };
24
25 wordTweetList = FOREACH wordsByTweet GENERATE
26     tweet,
27     FLATTEN(words) AS word;
28
29 /* Preprocessing Step 2 */
30 wordTweetList = pp_word(wordTweetList, 'tweet', 'word');
31
32 tweetsByWords = FOREACH (GROUP wordTweetList BY word) {
33     GENERATE
34         group AS word,
35         COUNT(wordTweetList) AS count,
36         wordTweetList.tweet AS tweets;
37 };
38
39 tweetsByFrequentWords = FILTER tweetsByWords BY count >= 500;
40
41 wordTweetList = FOREACH tweetsByFrequentWords GENERATE
42     FLATTEN(tweets) AS tweet,
43     word;
44
45 wordsByTweet = FOREACH (GROUP wordTweetList BY tweet) {
46     words = DISTINCT wordTweetList.word;
47     GENERATE
48         group AS tweet,
49         words;
50 };
51
52 tupleTweetList = FOREACH wordsByTweet GENERATE
53     FLATTEN(UnorderedPairs(words)) AS (base, extension),
54     tweet;
55
56 tweetsByTuples = FOREACH(GROUP tupleTweetList BY (base, extension)) {
57     GENERATE
58         group.base.word AS base,
59         group.extension.word AS extension,

```

```
60     COUNT(tupleTweetList) AS count ,
61     tupleTweetList.tweet AS tweets;
62 };
63
64 /* Keep only Tuples with a Support > 500 */
65 tweetsByTuples = FILTER tweetsByTuples BY count > 500;
66
67 /* Store Tuples and Support Value in 'output/' */
68 orderedTweetsByTuples = FOREACH tweetsByTuples GENERATE count, CONCAT( CONCAT( base, ' ' )
69     , extension ) AS words;
70 orderedTweetsByTuples = ORDER orderedTweetsByTuples BY count;
71 RMF output/;
72 STORE tmp INTO 'output/';
```

Listing D.3: frequentPairs.pig

E Frequent und maximal Frequent Itemsets

Die folgenden Abbildungen sollen den mengenmäßigen Unterschied zwischen den Frequent Itemsets, die mit Hilfe des FP-Growth Algorithmus ermittelt wurden und der reduzierten Menge der maximal Frequent Itemsets verdeutlichen. Bei den Frequent Itemsets in Abbildung E.1 handelt es sich um Teilmengen und Variationen der drei maximal Frequent Itemsets aus Abbildung E.2.



Abbildung E.1.: Alle Frequent Itemsets (minSupport = 0,001)

[@feel, @onedirection, happening, hope, feeling, believer, theretheres, here]

[#cometlanding, @philae2014, touchdown, address, 67p]

[@girl, week, @real_liam_payne, youre, itunes, almighty, enjoying, tracks]

Abbildung E.2.: Nur maximal Frequent Itemsets (minSupport = 0,001)

F Ergebnisse der Benutzerevaluierung

TP = True Positives, FP = False Positives, TN = True Negatives, FN = False Negatives

| Versuchsperson | TP | FP | TN | FN | Precision | Recall | F1-Score | Accuracy |
|-------------------|----|----|----|----|-----------|--------|----------|----------|
| 1 | 67 | 53 | 0 | 0 | 0,558 | 1,000 | 0,716 | 0,558 |
| 2 | 28 | 92 | 0 | 0 | 0,233 | 1,000 | 0,378 | 0,233 |
| 3 | 29 | 91 | 0 | 0 | 0,241 | 1,000 | 0,389 | 0,241 |
| 4 | 31 | 89 | 0 | 0 | 0,258 | 1,000 | 0,410 | 0,258 |
| 5 | 30 | 90 | 0 | 0 | 0,250 | 1,000 | 0,400 | 0,250 |
| 6 | 65 | 55 | 0 | 0 | 0,541 | 1,000 | 0,702 | 0,541 |
| 7 | 80 | 40 | 0 | 0 | 0,666 | 1,000 | 0,800 | 0,666 |
| 8 | 25 | 95 | 0 | 0 | 0,208 | 1,000 | 0,344 | 0,208 |
| Mittelwert | 44 | 76 | 0 | 0 | 0,369 | 1,000 | 0,517 | 0,369 |

Tabelle F.1.: Baseline

| Versuchsperson | TP | FP | TN | FN | Precision | Recall | F1-Score | Accuracy |
|-------------------|----|----|----|----|-----------|--------|----------|----------|
| 1 | 64 | 19 | 34 | 3 | 0,771 | 0,9552 | 0,816 | 0,853 |
| 2 | 22 | 1 | 91 | 6 | 0,956 | 0,785 | 0,862 | 0,941 |
| 3 | 5 | 0 | 91 | 24 | 1,000 | 0,172 | 0,294 | 0,800 |
| 4 | 19 | 0 | 89 | 12 | 1,000 | 0,612 | 0,760 | 0,900 |
| 5 | 17 | 2 | 88 | 13 | 0,894 | 0,566 | 0,693 | 0,870 |
| 6 | 57 | 15 | 40 | 8 | 0,791 | 0,876 | 0,832 | 0,808 |
| 7 | 77 | 14 | 26 | 3 | 0,846 | 0,962 | 0,900 | 0,858 |
| 8 | 12 | 1 | 94 | 13 | 0,923 | 0,480 | 0,631 | 0,883 |
| Mittelwert | 34 | 7 | 69 | 10 | 0,897 | 0,676 | 0,728 | 0,859 |

Tabelle F.2.: Mit Personalisierungsfunktion

Abbildungsverzeichnis

| | | |
|------|---|----|
| 2.1 | Beispiel eines Tweets | 7 |
| 2.2 | Knowledge Discovery in Databases (KDD) [15] | 8 |
| 2.3 | Beziehung der Arten von Itemsets [52] | 14 |
| 2.4 | Generierung FP-Tree Schritte 1 bis 3 | 15 |
| 2.5 | Generierung FP-Tree Schritte 4 und 5 | 16 |
| 2.6 | Teilbaum des Items Bier | 17 |
| 2.7 | Conditional Trees aller rekursiven Aufrufe | 18 |
| 3.1 | Ablauf des Preprocessings | 23 |
| 4.1 | Muster nach Gruhl et al. [22] | 28 |
| 4.2 | Erzeugung des <i>Time Window Graphen</i> (Schritt 1) | 30 |
| 4.3 | Erzeugung des <i>Time Window Graphen</i> (Schritt 2) | 30 |
| 4.4 | Der <i>Main Graph</i> | 31 |
| 4.5 | Frequent Pattern Mining Ansatz | 32 |
| 4.6 | Online Clustering Ansatz | 33 |
| 5.1 | Aufbau des Gesamtsystems | 36 |
| 5.2 | Themen Übersicht | 38 |
| 5.3 | Der <i>Main Graph</i> | 39 |
| 5.4 | Der <i>Time Window Graph</i> | 39 |
| 5.5 | Reiter „Overview“ | 40 |
| 5.6 | Reiter „Patterns“ | 41 |
| 5.7 | Reiter „Tweets“ | 41 |
| 5.8 | Klassifizierung eines Themas | 42 |
| 5.9 | Cluster Übersicht | 42 |
| 5.10 | Reiter „Overview“ | 43 |
| 5.11 | Reiter „Distribution over Time“ | 44 |
| 5.12 | Reiter „Keywords“ | 44 |
| 6.1 | Trends ohne Entfernung der häufigen Wortkombinationen | 48 |
| 6.2 | Manuelle Annotation durch die Versuchsperson | 50 |
| 6.3 | Auswertung der Ergebnisse des Klassifikators | 50 |
| E.1 | Alle Frequent Itemsets (minSupport = 0,001) | 67 |
| E.2 | Nur maximal Frequent Itemsets (minSupport = 0,001) | 67 |

Tabellenverzeichnis

| | | |
|-----|--|----|
| 2.1 | Warenkorbbeispiel [52] | 13 |
| 2.2 | Häufigkeiten der Items | 16 |
| 2.3 | Verteilung der Patienten | 19 |
| 3.1 | Metadaten eines Tweets der Streaming API | 22 |
| 3.2 | Auswirkung des Preprocessings | 24 |
| 4.1 | Merkmale eines Clusters | 33 |
| 4.2 | Beispiel Cluster „European Space Agency“ | 34 |
| 4.3 | Beispiel Cluster „Charlie Hebdo“ | 35 |
| 6.1 | Veränderung des Trends „Sydney Siege“ | 48 |
| 6.2 | Auswertung der Benutzerevaluierung | 51 |
| F.1 | Baseline | 68 |
| F.2 | Mit Personalisierungsfunktion | 68 |

Literaturverzeichnis

- [1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining Association Rules Between Sets of Items in Large Databases. In ACM SIGMOD Record, volume 22, pages 207–216. Association for Computing Machinery (ACM), 1993.
- [2] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), volume 1215, pages 487–499, 1994.
- [3] Luca Maria Aiello, Georgios Petkos, Carlos Martin, David Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Goker, Ioannis Kompatsiaris, and Alejandro Jaimes. Sensing Trending Topics in Twitter. IEEE Transactions on Multimedia, 15:1268–1282, 2013.
- [4] James Allan, Ron Papka, and Victor Lavrenko. On-line New Event Detection and Tracking. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 37–45. Association for Computing Machinery (ACM), 1998.
- [5] Hila Becker, Mor Naaman, and Luis Gravano. Beyond Trending Topics: Real-World Event Identification on Twitter. International AAAI Conference on Web and Social Media (ICWSM), 11:438–441, 2011.
- [6] James Benhardus and Jugal Kalita. Streaming Trend Detection in Twitter. International Journal of Web Based Communities (IJWBC), 9:122–139, 2013.
- [7] Douglas Burdick, Manuel Calimlim, and Johannes Gehrke. MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases. In Proceedings of the 17th International Conference on Data Engineering (ICDE), pages 443–452. Institute of Electrical and Electronics Engineers (IEEE), 2001.
- [8] Moses Charikar. Similarity Estimation Techniques from Rounding Algorithms. In Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pages 380–388. Association for Computing Machinery (ACM), 2002.
- [9] TU Darmstadt. Merkblatt Externe Abschlussarbeiten. http://www.intern.tu-darmstadt.de/media/dezernat_ii/referat_iig/fuer_pruefende/merkblaetter/Info_externeAbschlussarbeiten.pdf. (Zugegriffen am 11. Februar 2015).
- [10] Barbara Di Eugenio, Nick Green, and Rajen Subba. Detecting Life Events in Feeds from Twitter. In International Conference on Semantic Computing (ICSC), pages 274–277, 2013.
- [11] Jack Dorsey. just setting up my twttr. <https://twitter.com/jack/status/20>, 2006. (Zugegriffen am 3. März 2015).
- [12] David Ediger, Karl Jiang, Jason Riedy, David Bader, Courtney Corley, Rob Farber, and William Reynolds. Massive Social Network Analysis: Mining Twitter for Social Good. In 39th International Conference on Parallel Processing (ICPP), pages 583–593. Institute of Electrical and Electronics Engineers (IEEE), 2010.
- [13] European Space Agency (ESA). Touchdown! Rosetta’s Philae Probe Lands on Comet. http://www.esa.int/Our_Activities/Space_Science/Rosetta/Touchdown!_Rosetta_s_Philae_probe_lands_on_comet, 2014. (Zugegriffen am 24. Februar 2015).

-
- [14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), volume 96, pages 226–231, 1996.
- [15] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. AI magazine, 17:37, 1996.
- [16] Apache Software Foundation. Apache Pig! <http://pig.apache.org>. (Zugegriffen am 21. Dezember 2014).
- [17] William B. Frakes. Introduction to Information Storage and Retrieval Systems. Space, 14:10, 1992.
- [18] Henri Gilbert and Helena Handschuh. Security Analysis of SHA-256 and Sisters. In Selected Areas in Cryptography, pages 175–193. Springer, 2004.
- [19] Kevin Gimpel, Nathan Schneider, Brendan O’Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah Smith. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies: short papers-Volume 2, pages 42–47. Association for Computational Linguistics (ACL), 2011.
- [20] Meghan Glenn, Stephanie Strassel, Junbo Kong, and Kazuaki Maeda. TDT5 Topics and Annotations. Linguistic Data Consortium (LDC), 2006.
- [21] Frank Grubbs. Procedures for Detecting Outlying Observations in Samples. Technometrics, 11:1–21, 1969.
- [22] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. Information Diffusion through Blogspace. In Proceedings of the 13th International Conference on World Wide Web (WWW), pages 491–501. Association for Computing Machinery (ACM), 2004.
- [23] Cornelia Györödi, Robert Györödi, and Stefan Holban. A Comparative Study of Association Rules Mining Algorithms. In 1st Romanian-Hungarian Joint Symposium on Applied Computational Intelligence (SACI), pages 213–222, 2004.
- [24] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11:10–18, 2009.
- [25] Jiawei Han, Jian Pei, and Yiwen Yin. Mining Frequent Patterns without Candidate Generation. In ACM SIGMOD Record, volume 29, pages 1–12. Association for Computing Machinery (ACM), 2000.
- [26] Philip Howard, Aiden Duffy, Deen Freelon, Muzammil Hussain, Will Mari, and Marwa Mazaid. Opening Closed Regimes: What was the Role of Social Media during the Arab Spring? 2011.
- [27] Google Inc. Angular JS - Superheroic JavaScript MVW Framework. <https://angularjs.org>, 2015. (Zugegriffen am 3. Januar 2015).
- [28] Twitter Inc. About Twitter. <https://about.twitter.com/company>, 2015. (Zugegriffen am 3. März 2015).
- [29] Benjamin Kuhlhoff. Wie ein Schüler via Twitter Sportjournalisten auf der ganzen Welt narrete. <http://www.11freunde.de/interview/wie-ein-schueler-twitter-sportjournalisten-auf-der-ganzen-welt-narrte>. (Zugegriffen am 11. Februar 2015).

-
- [30] Shamanth Kumar, Geoffrey Barbier, Mohammad Ali Abbasi, and Huan Liu. TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief. In International AAAI Conference on Web and Social Media (ICWSM), 2011.
- [31] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. Mining of Massive Datasets. Cambridge University Press, 2014.
- [32] Steven Liu and Kevin Oliver. Twitter Hosebird Client (hbc). <https://github.com/twitter/hbc>, 2015. (Zugegriffen am 17. März 2015).
- [33] Allie Mazzia and James Juett. Suggesting Hashtags on Twitter. EECS 545m, Machine Learning, Computer Science and Engineering, University of Michigan, 2009.
- [34] Andrew McCallum, Kamal Nigam, et al. A Comparison of Event Models for Naive Bayes Text Classification. In AAAI-98 workshop on learning for text categorization, volume 752, pages 41–48. Citeseer, 1998.
- [35] Daniel Meers, Taylor Auerbach, Lema Samandar, Alicia Wood, and David Meddows. Martin Place siege: Hostages taken in Lindt Chocolate shop by armed robber. <http://www.dailytelegraph.com.au/news/martin-place-siege-hostages-taken-in-lindt-chocolate-shop-by-armed-robber/story-fni0cx4q-1227156245751?nk=0b24d1ba900bc58748595f461b377016>, 2014. (Zugegriffen am 24. Februar 2015).
- [36] Super Monitoring. Twitter 2012 - Facts and Figures (infographic). <http://www.supermonitoring.com/blog/twitter-2012-facts-and-figures-infographic/>. (Zugegriffen am 11. Februar 2015).
- [37] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen Carley. Is the Sample Good Enough? Comparing Data from Twitter’s Streaming API with Twitter’s Firehose. arXiv preprint arXiv:1306.5204, 2013.
- [38] Miles Osborne, Saša Petrović, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Bieber no more: First story detection using Twitter and Wikipedia. In Proceedings of the Workshop on Time-aware Information Access (TAIA), volume 12, 2012.
- [39] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the TREC-2011 Microblog Track. In Proceedings of the 20th Text REtrieval Conference (TREC 2011), 2011.
- [40] peerreach.com. 4 ways how Twitter can keep growing. <http://blog.peerreach.com/2013/11/4-ways-how-twitter-can-keep-growing/>. (Zugegriffen am 11. Februar 2015).
- [41] Georgios Petkos, Symeon Papadopoulos, Luca Aiello, Ryan Skraba, and Yiannis Kompatsiaris. A soft frequent pattern mining approach for textual topic detection. In Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), page 25. Association for Computing Machinery (ACM), 2014.
- [42] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pages 181–189. Association for Computational Linguistics (ACL), 2010.
- [43] Saša Petrović, Miles Osborne, and Victor Lavrenko. The Edinburgh Twitter Corpus. In Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pages 25–26, 2010.

-
- [44] Phillips and Davis. RFC 5646: Tags for Identifying Languages. <https://tools.ietf.org/html/bcp47>, 2009. (Zugegriffen am 19. Februar 2015).
- [45] Swit Phuvipadawat and Tsuyoshi Murata. Breaking news detection and tracking in twitter. In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), volume 3, pages 120–123. Institute of Electrical and Electronics Engineers (IEEE), 2010.
- [46] Irina Rish. An Empirical Study of the Naive Bayes Classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence, volume 3, pages 41–46. IBM New York, 2001.
- [47] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named Entity Recognition in Tweets: An Experimental Study. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1524–1534. Association for Computational Linguistics (ACL), 2011.
- [48] Ronald Rivest. The MD5 message-digest algorithm. 1992.
- [49] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In Proceedings of the 19th International Conference on World Wide Web (WWW), pages 851–860. Association for Computing Machinery (ACM), 2010.
- [50] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: News in Tweets. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 42–51. Association for Computing Machinery (ACM), 2009.
- [51] Jon Swaine, Paul Lewis, and Dan Roberts. Grand jury decline to charge Darren Wilson for killing Michael Brown. <http://www.theguardian.com/us-news/2014/nov/24/ferguson-police-darren-wilson-michael-brown-no-charges>, 2014. (Zugegriffen am 24. Februar 2015).
- [52] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, et al. Introduction to Data Mining, volume 1. Pearson Addison Wesley Boston, 2006.
- [53] Andranik Tumasjan, Timm Oliver Sprenger, Philipp Sandner, and Isabell Welp. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. International AAAI Conference on Web and Social Media (ICWSM), 10:178–185, 2010.
- [54] Carnegie Mellon University. Twitter Natural Language Processing. <http://www.ark.cs.cmu.edu/TweetNLP/>, 2015. (Zugegriffen am 16. März 2015).
- [55] Steven Van Canneyt, Matthias Feys, Steven Schockaert, Thomas Demeester, Chris Develder, and Bart Dhoedt. Detecting Newsworthy Topics in Twitter. In Second Workshop on Social News on the Web (SNOW), pages 25–32, 2014.
- [56] Audrey Watters. How Recent Changes to Twitter’s Terms of Service Might Hurt Academic Research. http://readwrite.com/2011/03/03/how_recent_changes_to_twitter_terms_of_service_mi. (Zugegriffen am 31. März 2015).
- [57] Jie Yin. Clustering Microtext Streams for Event Identification. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 719–725, Nagoya, Japan, October 2013. International Joint Conference on Natural Language Processing (IJCNLP), Asian Federation of Natural Language Processing.
- [58] Ron Zacharski. A Programmer’s Guide to Data Mining. <http://guidetodatamining.com>, 2012. (Zugegriffen am 14. Februar 2015).

[59] Frankfurter Allgemeine Zeitung. Terroranschlag auf „Charlie Hebdo“ in Paris - 12 Tote. <http://www.faz.net/-gpf-7yanw>, 2015. (Zugegriffen am 24. Februar 2015).