HMD (2016) 53:514–529
DOI 10.1365/s40702-016-0237-6

SentiStorm: Echtzeit-Stimmungserkennung von Tweets

Eva Zangerle · Martin Illecker · Günther Specht

Eingegangen: 14. März 2016 / Angenommen: 23. Mai 2016 / Online publiziert: 24. Juni 2016

Zusammenfassung Das automatisierte Erkennen der Stimmung von Texten hat in den letzten Jahren stark an Bedeutung gewonnen. Insbesondere durch die rapide Zunahme der Geschwindigkeit, mit der in sozialen Medien Informationen verbreitet werden, ist eine Echtzeit-Bestimmung der Stimmung von Texten ein herausforderndes Problem. Der Mikroblogging-Dienst Twitter verzeichnet im Durchschnitt über 8000 versendete Nachrichten pro Sekunde. In dieser Arbeit stellen wir mit dem SentiStorm-Ansatz einen Ansatz zur Stimmungserkennung von Tweets vor. Dabei erzeugen wir in einem ersten Schritt Merkmalsvektoren für die Tweets, die sowohl linguistische Informationen über den Tweet (Wichtigkeit der Wörter, Wortarten), wie auch über Sentiment-Lexika gewonnene Stimmungsinformationen beinhalten. In einem zweiten Schritt führen wir mittels der Merkmalsvektoren eine Stimmungsklassifikation durch, die eine Einteilung in positive, negative oder neutrale Tweets ermöglicht. Die durchgeführten Evaluationen zeigen, dass der präsentierte Ansatz bezüglich der Qualität der erkannten Stimmung sehr gute Erkennungsraten garantiert. Weiter zeigen wir, dass der Ansatz mittels der Apache Storm Plattform problemlos für die Echtzeit-Stimmungserkennung von Tweets skaliert werden kann.

Schlüsselwörter Stimmungserkennung · Twitter · Klassifikation · Computerlinguistik · Skalierbarkeit · Text Mining

SentiStorm: Realtime Sentiment Detection of Tweets

Abstract The automatic detection of the sentiment of texts has become more and more important throughout the last years. Particularly, the rapid increase of the

E. Zangerle (✉) · M. Illecker · G. Specht
Datenbanken und Informationssysteme, Institut für Informatik, Universität Innsbruck, Technikerstraße 21A, 6020 Innsbruck, Österreich
E-Mail: eva.zangerle@uibk.ac.at

speed at which information is spread in social media makes real-time sentiment detection a challenging task. On the microblogging platform Twitter, more than 8,000 messages are sent every second. In this work, we present the SentiStorm approach, an approach for sentiment detection within tweets. We base the approach on feature vectors which contain linguistic information about the tweet content (weighting of words, word categories), as well as sentiment information which we gather based on sentiment lexica. Subsequently, we facilitate these feature vectors for a sentiment classification task which allows for distinguishing positive, negative and neutral tweets. Our conducted evaluations show that the proposed approach shows high classification accuracy. At the same time, we show that utilizing the Apache Storm platform we are able to easily scale the approach towards a real-time sentiment classification of tweets.

Keywords sentiment detection · Twitter · classification · computational linguistics · scalability · text mining

1 Einleitung

Das Erkennen von Stimmung in Texten ist zu einer zentralen Forschungsfrage im Bereich der Computerlinguistik geworden. Nicht nur in der Forschung, auch wirtschaftlich hat die Möglichkeit, die Stimmung in Texten zu bestimmen, hohe Relevanz. So basiert Kundenbeziehungsmanagement (engl. Customer Relationship Management) oft auf derartigen Analysen, um beispielsweise die Zufriedenheit von Kunden überwachen zu können. Das Forschungsfeld des „Opinion Mining“ beschäftigt sich mit dem Bestimmen der Meinung, Empfindung oder der Haltung von Menschen über die Analyse von Texten. Speziell in den letzten Jahren hat die Bedeutung derartiger Ansätze stark zugenommen, da über Social Media-Plattformen die Menge an zu analysierenden und analysierbarer Information stark zugenommen hat. Auf Twitter, der derzeit größten Mikroblogging-Plattform weltweit, werden täglich 400 Millionen Tweets gesendet (Twitter 2015). Eine Besonderheit, die Twitter auszeichnet, ist die Beschränkung der Nachrichten auf ein Maximum von 140 Zeichen. Dies macht das Erkennen von Stimmung in diesen Tweets besonders herausfordernd. So soll beispielsweise der Tweet „@Hugo I love you !! <3 :) #love“ als positiv erkannt werden, wohingegen für „#sad Not going to carnival tomorrow :(“ negative Empfindungen erkannt werden soll. Bereits diese zwei verhältnismäßig einfachen Beispiele zeigen, dass die automatische Bestimmung von Stimmung (engl.: Sentiment) in derart kurzen Texten ein komplexes Problem darstellt. Insbesondere das Erkennen der Stimmung in Echtzeit ist aufgrund des hohen Datenaufkommens herausfordernd. Bisherige Ansätze stützen sich entweder auf maschinelles Lernen, auf Sentiment-Lexika, die für alle enthaltenen Wörter einen zugeordneten Sentiment-Wert beinhalten (Liu et al. 2012) oder auf eine Kombination beider Ansätze.

Im Rahmen dieses Artikels befassen wir uns mit den folgenden beiden Forschungsfragen (FF):

- FF1: Wie kann eine Stimmungserkennung von Tweets mittels der Kombination von Sentiment-Lexika und maschinell lernenden Verfahren erreicht werden?
- FF2: Wie kann die Stimmungserkennung von Tweets im Rahmen einer Echtzeitumgebung skaliert werden?

Im vorliegenden Artikel präsentieren wir den sogenannten SentiStorm-Ansatz zur Echtzeit-Stimmungserkennung in Tweets. SentiStorm nutzt dazu einen Ansatz, bei dem jeder Tweet durch einen Merkmalsvektor repräsentiert wird, der den Tweet und dessen Eigenschaften umfassend und bestmöglich charakterisieren soll. Dieser Merkmalsvektor beinhaltet Informationen über die Gewichtung der enthaltenen Wörter, den enthaltenen Wortarten und auch zur Stimmung einzelnen Wörter, die mithilfe von Sentiment-Lexika bestimmt wurde. Diese Merkmalsvektoren dienen anschließend als Input den eigentlichen Sentiment-Klassifizierungsprozess.

Die durchgeführten Evaluationen zeigen, dass durch den vorgestellten Ansatz, der mehrere Sentiment-Lexika kombiniert und auf den so erlangten Daten eine Stimmungsklassifikation durchführt, ähnlich präzise Ergebnisse wie bisherige Ansätze erzielt werden können. Im Gegensatz zu früheren Ansätzen ist SentiStorm auch bezüglich der Skalierbarkeit optimiert. So können wir zeigen, dass durch den Einsatz von Apache Storm, einer skalierbaren Plattform zur Echtzeit-Datenstrom-Verarbeitung, eine Echtzeit-Stimmungserkennung von mehr als 2 Milliarden Tweets pro Tag mittels des SentiStorm-Ansatzes möglich ist, was ein Fünffaches des aktuellen Tweet-Aufkommens pro Tag darstellt (Twitter 2015).

Der vorliegende Artikel ist wie folgt aufgebaut. In Abschn. 2 wird zunächst auf verwandte Arbeiten eingegangen und Hintergrundinformationen zur vorliegenden Arbeit präsentiert. Abschn. 3 stellt im Folgenden den SentiStorm-Ansatz vor. Die Methoden zur Evaluation des vorgeschlagenen Ansatzes werden in Abschn. 4 präsentiert. Abschn. 5 präsentiert die Ergebnisse der Evaluation des SentiStorm-Ansatzes, Abschn. 6 beschließt anschließend den Artikel.

2 Hintergrund und verwandte Arbeiten

Im Bereich der Stimmungserkennung ist die Klassifikation von Texten in positive, negative oder neutrale Stimmung ein sehr ausführlich erforschtes Gebiet (Liu 2016; Pang und Lee 2008). Prinzipiell lässt sich dieses Feld in zwei Bereiche unterteilen: Ansätze, die sich auf unüberwachtes Lernen stützen und solche, denen überwachtes Lernen zugrunde liegt. Im Folgenden möchten wir diese kurz beschreiben und im Anschluss auf verwandte Arbeiten in diesen Bereichen eingehen.

Ansätze, die unüberwachtes Lernen einsetzen, stützen sich auf sogenannte Sentiment-Lexika. Dies sind Verzeichnisse, in denen jedem Wort ein numerischer Wert zugeschrieben wird, der die Stimmung des Wortes repräsentieren soll. So beinhaltet beispielsweise das AFINN-Lexikon für jedes Wort einen Wert zwischen –5 und 5, wobei 5 eine sehr positive Stimmung beschreibt, 0 eine neutrale und –5 eine sehr negative (Nielsen 2011). Mit einem derartigen Ansatz kann direkt die Stimmung von Texten bestimmt werden, indem der Text Wort für Wort mit den Einträgen im Lexikon verglichen werden und aus den einzelnen Stimmungswerten ein Gesamt-

wert für den Text aggregiert wird. Im Gegensatz dazu erfordert ein überwachtes Lernen eine Trainings- und Testphase, in denen mithilfe von bereits klassifizierten Daten ein Regelwerk „gelernt“ wird, um anschließend darauf basierend die Klassifikation von neuen Daten durchführen zu können. Dabei stützt sich die Klassifikation auf Merkmalsvektoren, die Dokumente bestmöglich beschreiben sollen. Diese Vektoren können Informationen über die Wörter (auch Anzahl, Wichtigkeit, etc.), die Wortarten und die Stimmung des Dokuments enthalten. Neben jenen Ansätzen, die eindeutig unüberwachtem Lernen und überwachtem Lernen zuordenbar sind, gibt es auch noch hybride Ansätze, deren Ziel es ist, die Vorteile beider Arten zu kombinieren (Liu 2016; Pang und Lee 2008).

Im Umfeld der Mikroblogging-Plattform Twitter wird die Bestimmung von Stimmung in Tweets verschiedenartig realisiert. Go et al. verwenden dazu Techniken des überwachten Lernens (Naive Bayes, Maximum Entropy oder Support Vector Maschinen (SVM)) (Go et al. 2009). Pak und Paroubek unterscheiden im Gegensatz dazu zwischen objektiven und subjektiven Tweets und verwenden einen eigenen Klassifizierer, der sich auf das Naive Bayes-Verfahren stützt (Pak und Paroubek 2010). Auch Barbosa und Feng präsentieren einen zweistufigen Ansatz, bei dem zunächst in einem ersten Klassifikationsschritt zwischen objektiven und subjektiven Tweets unterschieden wird (Barbosa und Feng 2010). In einem weiteren Schritt wird die Stimmung der subjektiven Tweets klassifiziert. Agarwal et al. verwenden lediglich zwei Klassen und repräsentieren Tweets als Bäume, um darüber verschiedene Merkmale zusammenfassen und dann klassifizieren zu können (Agarwal et al. 2011). Kouloumpis et al. (2011) verwenden – ähnlich zu dem in diesem Artikel präsentierten SentiStorm-Ansatz – erweiterte Merkmale, wie beispielsweise Emoticons, Abkürzungen, etc. Das beste Resultat erreichen die Autoren mit *n*-Grammen (Textfragmente, die aus *n* Wörtern bestehen, z. B. stellt „zum Beispiel“ ein Bigramm dar) und den erweiterten Merkmalen.

Die SemEval-Challenge (Rosenthal et al. 2014) ist eine jährliche Challenge, bei der verschiedene Aufgaben im Umfeld der Semantikanalyse von Texten gelöst werden. Dabei gibt es seit 2013 einen Task, der sich mit der Bestimmung von Stimmung in Tweets befasst und somit sehr relevant für diese Arbeit ist. Dabei sind besonders die Ansätze von Mohammad et al. (2013), Proisl et al. (2014) und Evert et al. (2014) zu erwähnen, da diese mit dem vorgestellten Ansatz verwandt sind. Mohammad et al. stützen ihren Ansatz auf verschiedene Sentiment-Lexika und führen anschließend basierend auf Vektoren, die aus den Wörtern, *n*-Grammen, Wortarten, syntaktischen Merkmale, Stimmung (über Lexika extrahiert), Emoticons und auch Negation, eine Klassifikation durch Mohammad et al. (2013). Der KLUE-Ansatz von Proisl et al. evaluiert verschiedene Klassifikationsalgorithmen und stellen fest, dass ein Maximum Entropy-Klassifizierer die besten Ergebnisse erzielt Proisl et al. (2014). Eine Erweiterung des KLUE-Ansatzes, SemanticKLUE, kann die Ergebnisse nochmals verbessern, indem erweiterte Merkmale verwendet werden (Evert et al. 2014).

Im Bereich der Echtzeit-Stimmungserkennung von Tweets mittels der Echtzeit-Datenstrom-Plattform Apache Storm (Apache Group 2015) gibt es nach derzeitigem Wissensstand einen weiteren Ansatz. Raina et al. (2014) präsentieren einen Ansatz, bei dem über Sentiment-Lexika die Stimmung von Tweets berechnet wird. Aller-

dings setzt dieser Ansatz – im Vergleich zu SentiStorm – keine weiteren Quellen oder Verfahren zur Verbesserung der Ergebnisse ein.

3 Sentistorm

Im folgenden Abschnitt präsentieren wir den SentiStorm-Ansatz und beschreiben, wie damit die Stimmung von Tweets extrahiert werden kann. Weiter beschreiben wir, wie die Skalierbarkeit des gewählten Ansatzes mithilfe der Apache Storm-Plattform (Apache Group 2015) sichergestellt werden kann.

3.1 Stimmungserkennung

In diesem Abschnitt möchten wir die Funktionsweise der Stimmungserkennung mithilfe von SentiStorm beschreiben. Abb. 1 gibt einen ersten Überblick über den Workflow des SentiStorm-Ansatzes. Der Input für SentiStorm ist eine Menge von Tweets, für die die entsprechende Stimmung erkannt werden soll. Prinzipiell wird bei SentiStorm nach einer Vorverarbeitungsphase (Tokenisierung, Bereinigung) für jeden Tweet ein Merkmalsvektor erzeugt, der sich aus gewichteten Wörtern des Tweets, Informationen über die verwendeten Wortarten und mithilfe der von Sentiment-Lexika extrahierten Stimmung des Tweets zusammensetzt. Mithilfe dieser Vektoren wird anschließend die Sentiment-Klassifikation durchgeführt. Im Folgenden gehen wir auf die einzelnen Schritte näher ein.

3.1.1 Tokenisierung

Im ersten Schritt, der Tokenisierung, werden Tweets in einzelne Wörter (Tokens) aufgesplittet. Dazu ist es zunächst notwendig, Unicode-Zeichen, die auf Twitter häufig für Emoticons verwendet werden, zu ersetzen, um eine richtige Tokenisierung sicherstellen zu können. Dazu ist es beispielsweise auch erforderlich, Emoticons nicht fälschlicherweise zu trennen. So wird beispielsweise „\u002c“ durch ein Komma im Tweet-Text ersetzt. Dies ermöglicht es, in einem nächsten Schritt die in Tweets enthaltenen HTML-Zeichenfolgen mit den entsprechenden Wörtern bzw. Zeichen zu

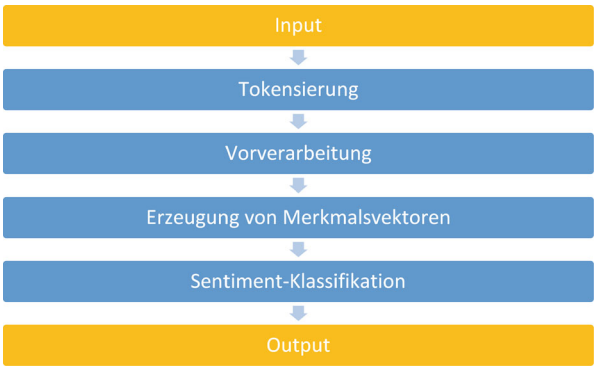


Abb. 1 Workflow Stimmungserkennung mit SentiStorm

Tab. 1 Beispiel Tokenizer

Input-Tweet	Tokenisierter Tweet
@user1 @user2 OMG I just watched Michael's comeback ! U remember him from the 90s ?? yaaaa \uD83Dn\DE09 #michaelcomeback	[@user1, @user2, OMG, I, just, watched, Michael's, comeback, !, U, remember, him, from, the, 90, s, ?, ?, yaaaa, ;), #michaelcomeback]

Tab. 2 Beispiel Vorverarbeitung

Input-Tokens	Vorverarbeitete Tokens
[@user1, @user2, OMG, I, just, watched, Michael's, comeback, !, U, remember, him, from, the, 90, s, ?, ?, yaaaa, ;), #michaelcomeback]	[@user1, @user2, Oh, my, God, I, just, watched, Michael's, comeback, !, you, remember, him, from, the, 90, s, ?, ?, yeah, ;), #michaelcomeback]

ersetzen. So wird beispielsweise „&“ durch ein Ampersand (&) ersetzt. Nach diesen vorbereitenden Arbeiten kann die eigentliche Tokenisierung mittels eines regulären Ausdrucks durchgeführt werden. Ein Beispiel für einen tokenisierten Tweets basierend auf einem gegebenen Input-Tweet ist in Tab. 1 angeführt.

3.1.2 Vorverarbeitung

Der nächste Schritt im SentiStorm-Workflow ist die Vorverarbeitung des bereits tokenisierten Tweets. Dabei wird der Tweet für die nachfolgenden Schritte vorbereitet und optimiert. Dies ist notwendig, da auf Twitter teils durch die Beschränkung auf 140 Zeichen bedingt eine nicht standardisierte Sprache verwendet wird (Go et al. 2009; Barbosa und Feng 2010). In diesem Zuge werden Emoticons auf die im SentiStrength Emoticons-Wörterbuch (Thelwall 2015) verwendeten Emoticons reduziert, da Twitter-Nutzer Emoticons oftmals durch das Wiederholen bzw. Verlängern von Zeichen verstärkt betonen möchten. Im Zuge der Vorverarbeitung wird so beispielsweise :-))) zu :-) transformiert. Auch werden Slang-Wörter und Abkürzungen durch ihre Grundformen ersetzt, dies geschieht mithilfe eines Slang-Wörterbuchs, das sowohl das Slang-Wort, als auch das entsprechende korrekte Wort enthält. Wie in Tab. 2 ersichtlich, wird beispielsweise „OMG“ mit „oh my god“ ersetzt. Im Vorverarbeitungsschritt werden auch für soziale Medien ebenso typische Verlängerungen von Wörtern, die dem Betonen dienen, durch ihre Grundform ersetzt (siehe Beispiel in Tab. 2: „yaaaa“ wird durch „yeah“ ersetzt). Auch werden unvollständige Wörter wie beispielsweise „goin“ durch die entsprechende vollständige Darstellung ersetzt (in diesem Fall mit „going“). Dies wird durch die Verwendung des WordNet-Wörterbuchs realisiert (Miller et al. 1990), mithilfe dessen die vollständigen Wörter gesucht und ersetzt werden.

3.1.3 Erzeugung von Merksmalsvektoren

Basierend auf den bisher durchgeführten Vorverarbeitungsschritten wird nun für jeden Tweet ein Merkmalsvektor erzeugt, der die Charakteristika des Tweets bestmöglich beschreiben soll und als Input für die anschließende Sentiment-Klassifikation dient. Wir schlagen vor, in den Merkmalsvektoren Informationen über die verwen-

deten Wortarten, Gewichtungen der Wörter und auch einen durch Sentiment-Lexika ermittelten Stimmungswert zu inkludieren. Bisherige Ansätze haben gezeigt, dass sich das Hinzufügen dieser Merkmale positiv auf die Qualität der Stimmungserkennung auswirkt (Evert et al. 2014). Im Folgenden gehen wir näher auf die Erzeugung der Merkmalsvektoren und die verwendeten Techniken ein.

Zunächst ist anzumerken, dass der Input für die Erzeugung der Merkmalsvektoren die bereits vorverarbeiteten Tweets sind. In einem ersten Schritt wenden wir die traditionelle TF/IDF-Gewichtung (Term Frequency Inverse Document Frequency) auf die in den Tweets enthaltenen Wörter an (Manning et al. 2008). TF/IDF ermöglicht es, Wörter anhand ihrer Relevanz zu gewichten. Dabei wird die Relevanz eines Wortes nicht ausschließlich über die Anzahl des Auftretens (Termfrequenz) berechnet, sondern auch in Bezug auf die Einzigartigkeit der Wörter. So werden beispielsweise Wörter, die in nahezu allen Dokumenten (in unserem Fall Tweets) vorkommen, niedrig gewichtet, da diese wenig charakteristisch für bestimmte Tweets sind und diese daher nur schlecht beschreiben können. Wörter, die hingegen nur in wenigen Tweets vorkommen, werden bezüglich des Tweets hoch gewichtet, da diese stark beschreibend für die Tweets, in denen sie enthalten sind, wirken und damit ein Alleinstellungsmerkmal darstellen. Mit der so errechneten Gewichtung eines Wortes bezüglich des Dokuments, in welchem es enthalten ist, können Dokumente besser charakterisiert werden, als es beispielsweise über die Termfrequenz allein möglich wäre.

Im nächsten Schritt erfolgt nun das Part-of-Speech-Tagging (POS) der vorverarbeiteten Tweets (Kroeger 2005). Dies hat zum Ziel, die Wortart jedes Wortes zu bestimmen und damit beispielsweise alle Nomen und Verben als solche zu kennzeichnen. Aus dieser Verteilung der Wortarten aggregieren wir in weiterer Folge daraus Merkmale für die Stimmungsklassifikation. Auch die Erkennung von Emoticons als solche wird in diesem Schritt durchgeführt. Bitte beachten Sie in Hinblick auf die vorgestellten, extrahierten Merkmale, dass in diesem Schritt nur die Anzahl der Emoticons extrahiert wird und nicht die Stimmung, die durch die Emoticons ausgedrückt wird. Die Stimmung der Emoticons wird im nächsten Schritt über Sentiment-Lexika extrahiert. Für das POS-Tagging wird die von der Carnegie Mellon-Universität entwickelte ARK-Tagging-Software (Owoputi et al. 2013) verwendet. Für die Merkmalsvektoren fügen wir folgende aus den POS-Tags generierten Werte hinzu:

- Anzahl Nomen
- Anzahl Verben
- Anzahl Adjektiven
- Anzahl Adverbien
- Anzahl Satzzeichen
- Anzahl Hashtags
- Anzahl Emoticons

Ein weiteres Merkmal, das im SentiStorm-Ansatz direkt in den Merkmalsvektor eines Tweets einfließt, ist die über Sentiment-Lexika ermittelte Stimmung des Tweets. Dazu bedienen wir uns einer Menge von sieben populären Sentiment-Lexika. Wir argumentieren diese Vielzahl an verwendeten Lexika dahingehend, dass wir

Tab. 3 Verwendete Sentiment-Lexika

Lexikon	Umfang	Sentiment-Wertebereich
AFINN-111 (Nielsen 2011)	2477 Wörter	[–5, 5]
SentiStrength Emotions (Thelwall 2015)	2544 reguläre Ausdrücke	[–5, 5]
SentiStrength Emoticons (Thelwall 2015)	107 Emoticons	[–1, 1]
Human Language Technology (2015) und Guerini, Gatti, Turchi (2013)	147.292 Wörter	[–0,935, 0,8827]
Sentiment140 (Kiritchenko, Zhu, Mohammad 2014)	62.468 Wörter	[–4,999, 5]
Bing Liu (Liu 2016; Liu et al. 2004)	6785 Wörter	[positiv, negativ]
MPQA Subjectivity (MPQA 2015)	6886 Wörter	[positiv, negativ]

mit der Gesamtheit an Einträgen die bestmögliche Erkennungsrate bzgl. der Stimmung von Wörtern bieten können. Auch haben vertiefende Evaluationen gezeigt, dass die Verwendung aller Lexika stets bessere Ergebnisse erzielt als einzelne Lexika oder Teilmengen an Lexika. Die Lexika und deren Charakteristika sind in Tab. 3 aufgezeigt. Aus der Tabelle ist klar erkennbar, dass die Lexika stark in ihrem Umfang und auch der Beschreibung von Sentiment (zwei- bis zehnstufige Skalen oder nur eine binäre Unterteilung in positiv oder negativ) variieren. Daher ist für eine Vergleichbarkeit der Lexika zunächst eine Skalierung der Stimmungsskalen notwendig, bei der wir alle vorhandenen Werteskalen auf den Wertebereich (0,1) normieren. Diese Skalierung führen wir mittels sogenanntem „Feature Scaling“ (Witten et al. 2005) durch. In Bezug auf den Merkmalsvektor schlagen wir vor, mittels der Lexika folgende Merkmale aus jedem Tweets zu extrahieren und dem entsprechenden Vektor hinzuzufügen:

- Anzahl an positiven Wörtern
- Anzahl an neutralen Wörtern
- Anzahl an negativen Wörtern
- Summe aller Sentiment-Werte
- Anzahl an Wörtern, für die ein Sentiment-Wert bestimmt werden konnte
- Maximaler positiver Sentiment-Wert
- Maximaler negativer Sentiment-Wert

Die verwendeten Sentiment-Lexika enthalten auch Überlappungen – Wörter, die in mehreren Sentiment-Lexika einen (möglicherweise unterschiedlichen) Sentiment-Wert zugewiesen bekommen. Wir behalten diese Überlappungen bei und verwenden für die Generierung des Merkmalsvektors alle für ein Wort ermittelbaren Sentiment-Werte, da unsere Experimente gezeigt haben, dass dieses Vorgehen zu verbesserten Resultaten führt.

3.1.4 Klassifikation

Die im vorigen Abschnitt beschriebenen Merkmalsvektoren dienen als Input für den Klassifikationsschritt. Das Ziel der Klassifikation ist, die Tweets anhand der sie beschreibenden Merkmalsvektoren in die drei Klassen positiv, negativ oder neutral einzuteilen. Bei der Wahl der Klassifikationsmethode stützen wir uns auf bisherige Forschungsergebnisse, die gezeigt haben, dass sich Support Vector Maschinen (SVM) (Cristianini und Shawe-Taylor 2000) für einen derartigen Task sehr gut eignen (Barbosa und Feng 2010; Mohammad et al. 2013; Kiritchenko, Zhu, Mohammad 2014). Vereinfacht gesprochen versucht eine Support Vector Maschine, die zu klassifizierenden Objekte so zu unterteilen, dass die Grenzen zwischen den Klassen möglichst klar ausfallen. Für die Berechnung von Tweet-Sentiments muss der Klassifizierer zunächst trainiert werden. D. h. die SVM muss mit ausreichend Beispielen für Merkmalsvektoren und dem damit verknüpften Sentiment versorgt werden, um daraus auch für zukünftig zu klassifizierende Merkmalsvektoren ein Sentiment ableiten zu können (basierend auf dem aus den Trainingsdaten berechneten Regelwerk). Eine SVM erfordert also stets bereits klassifizierte Beispiele, die als Trainingsdatensatz verwendet werden können.

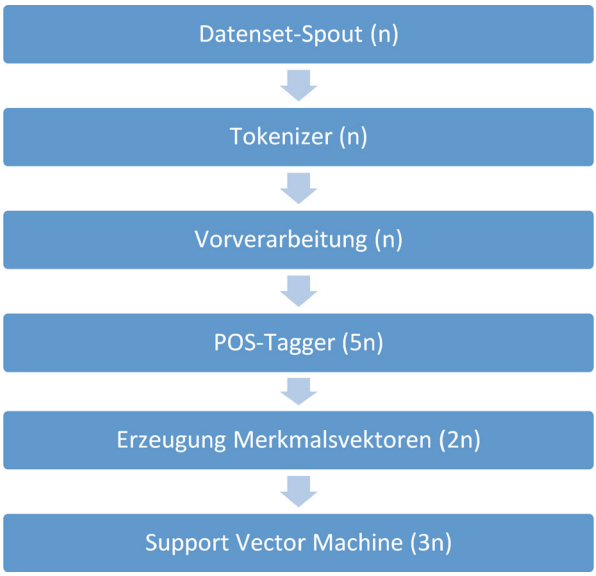
3.2 Skalierung von Sentistorm

Basierend auf dem im vorigen Abschnitt entwickelten Verfahren zur Stimmungsklassifikation möchten wir in einem nächsten Schritt den Ansatz für die Echtzeit-Erkennung von Stimmung in Tweets skalieren. Dazu stützen wir uns auf die Amazon Elastic Compute Cloud (EC2)[1]. Die EC2 ist ein Beispiel für viele verfügbare Angeboten, die es erlauben, in der Cloud flexibel Hardware für eigene Berechnungen und Services anmieten zu können. Für die vorliegende Evaluation kommen in der EC2 Rechner (sogenannte Knoten) mit 32 Kernen und 60 GB RAM zum Einsatz. Für eine optimale Skalierung des SentiStorm-Ansatzes evaluieren wir verschiedene Cluster-Topologien mit unterschiedlicher Anzahl an Cluster-Knoten, wobei eine Cluster-Topologie beschreibt, wie die verschiedenen Arbeitsschritte (Tokenisieren, Vorverarbeitung, POS-Tagger, Merkmalsvektoren extrahieren, SVM) auf die verschiedenen Knoten verteilt werden.

Für die Sentiment-Berechnung mithilfe des SentiStorm-Ansatzes stützen wir uns auf das Apache Storm-Framework, mithilfe dessen Datenströme in Echtzeit verarbeitet werden können. Dabei kontrolliert Storm die Verteilung der Daten, die Skalierung des Gesamtsystems und stellt auch die fehlertolerante Verarbeitung der Datenströme sicher. In Apache Storm legt der sogenannte „parallelism value“ fest, wie die abzuarbeitenden Tasks (im vorliegenden Fall Tokenization, Vorverarbeitung, etc.) unter den zur Verfügung stehenden Knoten im Cluster verteilt werden sollen (Apache Group 2016). So beschreibt beispielweise ein parallelism value von 50 für den POS-Tagger bei einer Topologie mit zehn Knoten, dass pro Knoten 5 Threads (parallelisierbare, leichtgewichtige Prozesse) für den POS-Tagger verwendet werden. Abb. 2 zeigt die für die Evaluation verwendete Topologie. Der sogenannte Datenset-Spout ist dabei

[1] https://aws.amazon.com/ec2/.

Abb. 2 SentiStorm-Topologie



jener Storm-Dienst, der Daten in die Storm-Pipeline eingibt und damit eine Datenquelle simuliert – im vorliegenden Fall werden eingehende Tweets in die Pipeline eingespeist. Die anderen Dienste wurden bereits im vorigen Abschnitt beschrieben. Dabei ist jeweils in runden Klammern der parallelism value angeführt. Bei einer Topologie, die *n* Knoten beinhaltet, läuft also auf jedem Knoten ein Datenset Spout-Thread, ein Tokenization-Thread, ein Vorverarbeitungs-Thread, fünf POS-Tagging-Threads, zwei Threads zu Erzeugung der Merkmalsvektoren und drei Threads für die Klassifikation mittels SVM. Diese Verteilung begründen wir damit, dass sich während unserer Experimente der POS-Tagger und auch die Support Vector Machine als Bottlenecks bezüglich des Durchsatzes herausgestellt haben. Dieses Problem können wir lösen, indem wir diesen Prozessen mehr Ressourcen (und damit Threads) zur Verfügung stellen.

4 Evaluation

Im folgenden Kapitel stellen wir die Methode, die zur Evaluation des präsentierten Ansatzes verwendet wurde, vor. Dabei konzentrieren wir uns zunächst auf die Qualität der berechneten Stimmung und in einem weiteren Evaluationsschritt auf die Skalierbarkeit und den Durchsatz des Ansatzes in einer Apache Storm-Implementierung.

Für die Evaluation verwenden wir ein Datenset, das bereits für die SemEval-Challenge im Jahre 2013 verwendet wurde (Nakov et al. 2013). Diese Challenge stellt einen Task bereit, der sich mit der Erkennung von Stimmung in Tweets befasst, bei dem jährlich die besten Ansätze zur Sentiment-Analyse von Tweets präsentiert und verglichen werden. Damit können wir bestmögliche Vergleichbarkeit des Sen-

Tab. 4 Datenset SemEval 2013

	Positiv	Negativ	Neutral	Gesamt
Training	3660	1466	4602	9729
Development	575	340	739	1654
Test	1572	601	1640	3813

tiStorm-Ansatzes mit bisherigen und aktuellen Ansätzen sicherstellen. Die Eckdaten des Datensets sind in Tab. 4 aufgelistet. Dabei verwenden wir sowohl das Training- als auch das Development-Set zum Trainieren des Klassifizierers und verwenden das Test-Set wird zur Evaluation des Ansatzes (siehe auch Beschreibung der Klassifikation des SentiStorm-Ansatzes). Für alle drei Datensätze (Training, Development und Test) werden die Tweet-Inhalte und auch die Sentiment-Klasse bereitgestellt. Aus der Tabelle ist gut ersichtlich, dass die Verteilung der drei Klassen unausgewogen ist.

Zur Evaluation des vorgeschlagenen Ansatzes stützen wir uns auf die weit verbreiteten Metriken Recall und Precision (Theodoridis and Koutroumbas 2008). Dabei beschreibt der Recall-Wert (im Deutschen auch oft Trefferquote genannt), wie viele der positiven, negativen und neutralen Tweets wirklich als solche klassifiziert wurden. Precision (dt. Genauigkeit) hingegen beschreibt, wie viele der als positiv, negativ oder neutral klassifizierten Tweets auch wirklich der entsprechenden Klasse angehören. Diese Metriken können sowohl für alle Klassen gemeinsam, als auch für jede Klasse einzeln evaluiert werden. Oftmals werden Recall und Precision auch in einen Wert zusammengefasst: die sogenannte F-Metrik ergibt sich aus dem harmonischen Mittelwert von Precision und Recall (Nakov et al. 2013; Witten et al. 2011). Wir verwenden die F-Metrik für die Evaluation, da die SemEval Challenge den Durchschnitt der F-Metrik für die positiven und negativen Klasse zum Vergleich der eingereichten Ansätze verwendet. Da wir die von SentiStorm erreichten Ergebnisse mit den Ergebnissen der SemEval vergleichen möchten, berechnen wir diesen Durchschnitt wie folgt (Nakov et al. 2013):

$$F_{p/n} = \frac{F_{pos} + F_{neg}}{2}$$

Um auch die Skalierbarkeit bezüglich der Echtzeitanforderung evaluieren zu können, testen wir den SentiStorm-Ansatz mit verschiedenen Topologien. Als Maßzahl verwenden wir bei der Evaluation den Durchsatz an Tweets und damit die Anzahl an Tweets, für die pro Sekunde die Sentiment-Klasse bestimmt werden kann. Als Input für diese Evaluation verwenden wir wiederum das bereits beschriebene Datenset der SemEval-Challenge 2013. Dieses beinhaltet allerdings lediglich 15.196 Tweets und muss daher vergrößert werden. Daher verwenden wir das SemEval-Datenset mehrfach als Input. D. h., wir fügen den Inhalt des Datensets mehrfach zusammen, um so zu einer größeren Anzahl an Testdaten zu gelangen und den erreichten Durchsatz realistisch evaluieren zu können.

Tab. 5 Wahrheitsmatrix SentiStorm-Ansatz

		Ermittelte Klasse			
		Positiv	Negativ	Neutral	Gesamt
Klasse	Positiv	1033	146	393	1572
	Negativ	56	412	133	601
	Neutral	257	151	1232	1640
	Gesamt	1346	709	1758	3813

Tab. 6 SemEval Top-5 Ergebnisse der Jahre 2013 und 2014

2013	$\mathbf{F}_{p/n}$	**2014**	$\mathbf{F}_{p/n}$
NRC-Canada	0,6902	TeamX	0,7212
GU-MLT-LT	0,6527	NRC-Canada	0,7075
Teragram	0,6486	Coooolll	0,7040
BOUNCE	0,6353	RTRGO	0,6910
KLUE	0,6303	SentiKLUE	0,6906

5 Resultate

Im folgenden Abschnitt präsentieren wir die Resultate der durchgeführten Evaluation des SentiStorm-Ansatzes. Die dazu verwendeten Evaluationsmethoden sind in Abschn. 4 beschrieben.

Tab. 5 zeigt die Wahrheitsmatrix nach der Evaluation des SentiStorm-Ansatzes basierend auf dem Datensatz der SemEval 2013, das bereits in Abschn. 4 vorgestellt wurde. Darin sind die korrekte Sentiment-Klasse der Tweets (Zeilen) und die durch den SentiStorm-Ansatz ermittelte Klasse (Spalten) angeführt. Die Anzahl der korrekt klassifizierten Tweets ist für alle drei Klassen (positiv, negativ und neutral) grau hinterlegt. So ist zum Beispiel beispielsweise erkennbar, dass der SentiStorm-Ansatz 1033 der gesamt 1572 positiven Tweets korrekt klassifiziert und 393 der positiven Tweets fälschlicherweise als neutral eingestuft werden.

Basierend auf dieser Tabelle können wir nun die Qualität des SentiStorm-Ansatzes weiter analysieren. In einem ersten Schritt berechnen wir Recall und Precision der drei angeführten Klassen und anschließend die F-Metrik. Daraus ergeben sich für die Klasse der positiven Tweets ein Recall von 65,71 %, Precision von 76,75 % und eine F1-Metrik von 70,80 %. Für die Klasse der negativen Tweets ergeben sich ein Recall von 68,55 %, eine Precision von 58,11 % und eine F1-Metrik von 62,90 %. Für die neutrale Klasse liegt der Recall bei 75,12 %, Precision bei 70,08 % und F1 bei 72,51 %.

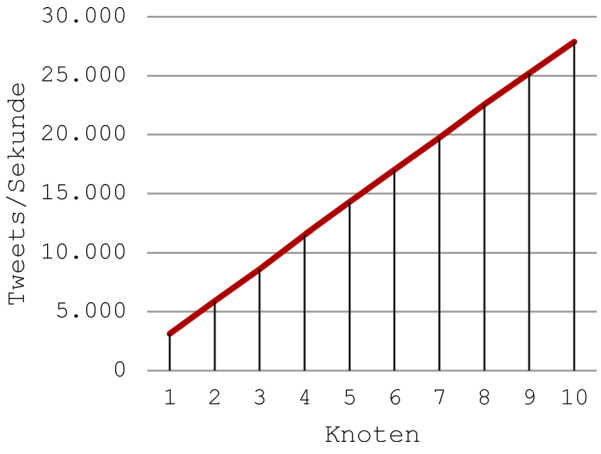
Für die $F_{p/n}$-Metrik ergibt sich ein Resultat von 70,12 % für den SentiStorm-Ansatz. Im Vergleich zu den besten fünf Teams der SemEval Competitions der Jahre 2013 (Rosenthal et al. 2014; Nakov et al. 2013), die in Tab. 6 aufgeführt sind, stellen wir fest, dass der SentiStorm-Approach mit den erreichten 66.85 % im Jahre 2013 den zweiten Rang und im Jahre 2014 den sechsten Rang belegt hätte.

Die Ergebnisse der Skalierbarkeitsevaluation möchten wir nun im folgenden Abschnitt präsentieren. Um das Ziel der Echtzeit-Sentiment-Analyse gerecht zu werden, ist es notwendig, bei durchschnittlichem Tweet-Aufkommen die Stimmung

Tab. 7 Durchsatz SentiStorm auf Amazon EC2

Knoten	Durchschnitt Tweets/Sekunde
1	3133
2	5920
3	8599
4	11.528
5	14.295
6	17.025
7	19.735
8	22.579
9	25.207
10	27.876

Abb. 3 Skalierbarkeit



der Tweets in Echtzeit berechnen zu können. Im August 2013 war die von Twitter bekannt gegebene durchschnittliche Anzahl an Tweets pro Sekunde 5700 und die maximale Spitze betrug 143.199 Tweets pro Sekunde (Erstausstrahlung der Serie „Castle in the Sky“ in Japan) (Twitter 2013). 2015 erhöhte sich die durchschnittliche Anzahl von Tweets pro Sekunde auf 8793 (Internetlivestats 2015).

Tab. 7 zeigt die Skalierungsmöglichkeit des SentiStorm-Ansatzes. Dabei listen wir die Anzahl der Knoten, die für die Berechnung der Stimmung verwendet wurden. Die Evaluation des SentiStorm-Ansatzes auf der Amazon Elastic Compute Cloud zeigt, dass die zuvor beschriebene Echtzeit-Anforderung bereits mit dem Einsatz von vier Knoten erfüllt werden kann, diese Konfiguration verfügt über einen Durchsatz von 11.528 Tweets pro Sekunde. Bereits mit einem einzigen Knoten können pro Sekunde 3133 Tweets verarbeitet werden. Wir können also feststellen, dass der vorgestellte SentiStorm-Ansatz mittels einer Verteilung auf der Amazon EC2 (als ein Stellvertreter für Cloud-Computing-Plattformen) linear skaliert werden kann. Lineare Skalierbarkeit bedeutet, dass die Anzahl an zusätzlich bestimmbaren Tweets mit jedem neu hinzugefügten Knoten konstant bleibt. Diese lineare Skalierbarkeit ist zusätzlich nochmals in Abb. 3 dargestellt.

Die durchgeführten Evaluationen des in diesem Artikel präsentierten Ansatzes zeigen, dass die gewählte Methode mit den erfolgreichsten Ansätzen der SemEval-Challenge aus den Jahren 2013 und 2014 konkurrieren kann. Zugleich konnten wir zeigen, dass dieser Ansatz über Apache Storm derart skaliert werden kann, dass eine Echtzeit-Stimmungserkennung von Tweets problemlos durchführbar wäre. Hinsichtlich Forschungsfrage FF1 können wir also festhalten, dass mithilfe der präsentierten Merkmalsvektoren, die TF/IDF-gewichtete Wörter, POS-Tags und auch über Sentiment-Lexika gewonnene Stimmungsinformation, eine präzise Klassifizierung von Tweets durchführbar ist. Bezüglich Forschungsfrage FF2 haben unsere Experimente gezeigt, dass sich ein derartiger Ansatz linear skalieren lässt und bereits mit zehn Knoten weit über die Echtzeit-Anforderung hinaus skalierbar ist. Mit dem Einsatz von vier Knoten können bereits 11.528 Tweets pro Sekunde verarbeitet werden. In einer Konfiguration mit zehn Knoten können somit 1,6 Millionen Tweets pro Minute, und 2,4 Milliarden Tweets pro Tag verarbeitet werden. Zum Vergleich: im Jahre 2015 wurden täglich etwa 500 Milliarden Tweets pro Tag gesendet (Internetlivestats 2015). Natürlich unterliegt das Tweetaufkommen Schwankungen, so liegt – wie schon beschrieben – der Spitzenwert bei über 143.000 Tweets pro Sekunde. Derartige Spitzen sollen in der vorliegenden Konfiguration jedoch auch kein Problem sein, da die nicht direkt verarbeitbaren Tweets gecached werden und danach entsprechend der verfügbaren Ressourcen verzögert verarbeitet werden können.

6 Ausblick

Im vorliegenden Artikel haben wir den SentiStorm-Ansatz zur Echtzeit-Extraktion der Stimmung in Tweets vorgestellt. Die Evaluationsergebnisse zeigen zum einen, dass der präsentierte Ansatz in Bezug auf die Qualität der Stimmungserkennung akkurat arbeitet und zum anderen, dass der SentiStorm-Ansatz auf der Amazon Elastic Compute Cloud bereits unter Verwendung von vier Knoten die Echtzeit-Anforderung erfüllen kann.

Zukünftige Herausforderungen und Aufgaben können einerseits in Bezug auf die Qualität der Sentiment-Klassifikation und andererseits bezüglich der Skalierbarkeit und Skalierung definiert werden. In Hinblick auf die Qualität der Sentiment-Klassifikation gilt es, an zwei Punkten anzusetzen: dem eigentlichen Klassifikationsverfahren und dem Berücksichtigen von sprachlichen Merkmalen. In Bezug auf die Klassifikationsmethode gilt es, auch weitere Methoden, die sich als passend und performant erwiesen haben (z. B. Maximum Entropy-Klassifikation), für dem SentiStorm-Ansatz zu evaluieren und zu optimieren. Bezüglich der linguistischen Verarbeitung liegen die Herausforderungen im Erkennen und Abbilden von Negationen und Sarkasmus, um eine verbesserte Extraktion von Stimmung zu ermöglichen. Hinsichtlich der Skalierung des Systems zeigen die vorliegenden Evaluationen, dass sowohl das POS-Tagging, wie auch die Support Vector Maschine noch Potential für Verbesserungen bezüglich der Verarbeitungsdauer bieten, die es auszunützen gilt.

Open access funding provided by University of Innsbruck and Medical University of Innsbruck

Literatur

Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R (2011) Sentiment analysis of Twitter data. In: Proceedings of the workshop on languages in social media, S 30–38. Konferenzjahr: 2011, Konferenzort: Portland, Oregon, USA

Apache Group (2015) Apache storm. https://storm.apache.org/documentation/Concepts.html. Zugegriffen: 14-Mar-2016

Apache Group (2016) Understanding the parallelism of a storm topology. http://storm.apache.org/documentation/Understanding-the-parallelism-of-a-Storm-topology.html. Zugegriffen: 13-Mar-2016

Barbosa L, Feng J (2010) Robust sentiment detection on Twitter from biased and noisy data. In: Data Proceedings of the 23rd International Conference on Computational Linguistics: Posters, S 36–44. Konferenzjahr: 2010, Konferenzort: Peking, China.

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Evert S, Proisl T, Greiner P, Kabashi B (2014) SentiKLUE: updating a polarity classifier in 48 hours. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), S 551–555

Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. Processing 150(12):1–6

Guerini M, Gatti L, Turchi M (2013) Sentiment analysis: how to derive prior polarities from SentiWordNet. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, S 1259–1269

Human Language Technology (2015) SentiWords. https://hlt.fbk.eu/technologies/sentiwords Accessed: 10-Mar-2016

Internetlivestats (2015) 1 second – internet live stats. http://www.internetlivestats.com/one-second/#tweets-band. Zugegriffen: 10. März 2016

Kiritchenko S, Zhu X, Mohammad S (2014) Sentiment analysis of short informal texts. J Artif Intell Res 50:723–762

Kouloumpis E, Wilson T, Moore J (2011) Twitter sentiment analysis: the good the bad and the OMG! ICWSM 11:538–541

Kroeger PR (2005) Analyzing grammar: an introduction. Cambridge University Press, Cambridge

Liu B (2012) Sentiment analysis and opinion mining. Morgan & Claypool, San Rafael, California, USA

Liu B (2016) Bing Liu sentiment lexicon. http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon. Zugegriffen: 04. März 2016

Liu B, Li X, Lee WS, Yu PS (2004) Text classification by labeling words. In: Proceedings of the 19th National Conference on Artifical Intelligence, S 425–430

Manning CD, Raghavan P, Schütze H (2008) Scoring, term weighting, and the vector space model. In: Introduction to Information Retrieval. Cambridge University Press, Cambridge, S 100–123

Miller GA, Beckwith R, Fellbaum C, Gross D, Miller KJ (1990) Introduction to WordNet: an online lexical database. Int J Lexicogr 3(4):235–244

Mohammad S (2015) Sentiment140. http://www.saifmohammad.com/WebPages/lexicons.html. Zugegriffen: 11. März 2016

Mohammad S, Kiritchenko S, Zhu X (2013) NRC-Canada: building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of the seventh International Workshop on Semantic Evaluation Exercises (SemEval-2013)

MPQA (2015) MPQA subjectivity lexicon. http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/. Zugegriffen: 10-Mar-2016

Nakov P, Rosenthal S, Kozareva Z, Stoyanov V, Ritter A, Wilson T (2013) SemEval-2013 Task 2: sentiment analysis in Twitter. In: Second joint conference on lexical and computational semantics (*SEM)

proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) Bd. 2, S 312–320
Nielsen FÅ (2011) AFINN. In: Informatics and mathematical modelling. Technical University of Denmark, Lyngby
Owoputi O, O'Connor B, Dyer C, Gimpel K, Schneider N, Smith AN (2013) Improved part-of-speech tagging for online conversational text with word clusters. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, S 380–390
Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC'10). Jahr: 2010, Ort: Valletta, Malta
Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135
Proisl T, Evert S, Greiner P, Kabashi B (2014) SemantiKLUE: robust semantic similarity at multiple levels using maximum weight matching. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), S 532–540. Konferenzort: Dublin, Irland, Konferenzjahr: 2014
Raina I, Gujar S, Shah P, Desai A, Bodkhe B (2014) Twitter sentiment analysis using apache storm. Int J Recent Technol Eng 3(5):23–26
Rosenthal S, Ritter A, Nakov P, Stoyanov V (2014) SemEval-2014 Task 9: sentiment analysis in Twitter. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), S 73–80. Konferenzort: Dublin, Irland, Konferenzjahr: 2014
Thelwall M (2015) SentiStrength. http://sentistrength.wlv.ac.uk. Zugegriffen: 11. März 2016
Theodoridis S, Koutroumbas K (2008) Pattern recognition, 4. Aufl. Academic Press, Burlington, MA, USA; San Diego, California, USA; London, United Kingdom
Twitter (2013) New tweets per second record, and how! https://blog.twitter.com/2013/new-tweets-per-second-record-and-how. Zugegriffen: 11. März 2016
Twitter (2015) About Twitter. https://about.twitter.com/company. Zugegriffen: 20-Feb-2016
Witten IH, Frank E, Hall MA (2005) Data mining practical machine learning tools and techniques. Morgan Kaufmann, San Francisco
Witten IH, Frank E, Hall MA (2011) Data mining practical machine learning tools and techniques, 3. Aufl. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA