Heinrich C. Mayr, Martin Pinzger (Hrsg.): INFORMATIK 2016,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016 2133

Twistor – Simulation des Twitterstroms für Evaluationszwecke

Harry Schilling[1]

Abstract: Twitter ist ein Mikroblogging-Dienst, in dem aktuelle Ereignisse und Geschehnisse diskutiert werden. Es existiert eine Reihe von Algorithmen, um solche Ereignisse aus Twitterdaten extrahieren zu können. Die Evaluation dieser Verfahren gestaltet sich schwierig. Es stellt sich die Frage, ob ein gefundenes Ereignis auch ein Ereignis aus der Realität repräsentiert. Des Weiteren ist auch nicht bekannt, ob das Verfahren alle Ereignisse findet, da die Informationen fehlen, wie viele Ereignisse in den Twitterdaten wirklich vorhanden sind. Auch ist die Weitergabe von Twitterdaten verboten, sodass jeder Forscher gezwungen ist, seine eigenen Daten zu sammeln, die sich hinsichtlich der Qualität unterscheiden. Aufgrund dessen sind die Evaluationen von Ereigniserkennungsalgorithmen oft sehr heterogen und schlecht miteinander vergleichbar. Eine Möglichkeit diese Probleme anzugehen, bietet Twistor (**Twi**tter **St**ream Simulat**or**). Twistor ist ein Verfahren, um den Twitterstrom (mit dazugehörigen Ereignissen) zu simulieren. Somit ist immer eine konsistente Datenbasis und die Information über die zu suchenden Ereignisse vorhanden, um die Ergebnisse des Ereigniserkennungsalgorithmus bewerten und so einer einheitlichen Evaluation unterziehen zu können.

Keywords: Twitter Social Media Stream, Event Detection, Evaluation, Simulation

1 Einführung

Ein Mikroblogging-Dienst wie Twitter ist ein Medium, das Benutzern erlaubt, sehr kleine, digitale Inhalte wie z.B. kleine Texte mit anderen Benutzern zu tauschen. Dabei zeichnet sich Twitter vor allem dadurch aus, dass kurze Nachrichten schnell verarbeitet und geteilt werden können. So ist es möglich, Informationen in Echtzeit zu verbreiten und so aktuelle Geschehnisse und Ereignisse zu diskutieren oder zu kommentieren. Ereignisse wie z.B. Naturkatastrophen verbreiten sich mittels Twitter innerhalb kürzester Zeit [SOM10]. Es gibt eine Reihe von Algorithmen, die versuchen diese Ereignisse zu erfassen. Ein Problem hierbei ist, dass Twitter zwar unendlich viele und sehr schnell Informationen liefert, aber dass Twitternachrichten oft wenig gehaltvoll sind [KH11].
Neben der Ereigniserkennung an sich bereitet auch die Evaluation der Ergebnisse eines Ereigniserkennungsalgorithmus Schwierigkeiten. So muss z.B. sichergestellt werden, ob ein gefundenes Ereignis auch ein tatsächliches Ereignis repräsentiert. Um die Qualität der Ergebnisse richtig einordnen zu können, müssen neben der Verifikation eines Ereignisses auch alle in den Daten vorhandene Ereignisse bekannt sein. Hierzu können z.B. Nachrichtenseiten wie Reuters[2] hinzugezogen werden. Das ist deshalb problematisch, da die

[1] Universität Konstanz, Fachbereich Informatik und Informationswissenschaft, P.O. Box 188, 78457 Konstanz, Germany, harry.schilling@uni-konstanz.de
[2] `http://www.reuters.com/`

in den Daten vorhandenen Ereignisse nicht unbedingt die Ereignisse der hier beispielhaft aufgeführten Nachrichtenseite Reuters widerspiegeln. Infolgedessen sind viele Evaluationen von Ereigniserkennungsalgorithmen oft sehr unterschiedlich gestaltet, was zu einer schlechten Vergleichbarkeit zwischen den verschiedenen Ereigniserkennungsverfahren führt. Hinzu kommt, dass die Weitergabe von Twitterdaten von Twitter verboten wurde. Somit ist auch die Weitergabe der Evaluation zugrunde liegenden Daten untersagt. Es muss folglich jeder Forscher seine eigenen Twitterdaten sammeln, die sich in Qualität und Quantität unterscheiden. Die unterschiedliche Quantität und Qualität ist auch auf die Wahl der verschiedenen Twitter APIs, die benutzt werden, um die Daten zu sammeln, zurückzuführen. Möchte man mehr als 1 % der Twitterdaten erfassen, müssen verschiedene öffentlich zugängliche APIs miteinander kombiniert werden. Eine API, die einen speziellen Zugang erfordert, liefert 10 % des Stroms (Gardenhose). Gegen eine Bezahlung ist es möglich, 100 % des Stroms zu erfassen (Firehose). Auch die auf verschiedene Arten gesammelten Daten führen dazu, dass sich die Evaluationsergebnisse zwischen verschiedenen Ereigniserkennungsverfahren schlecht vergleichen lassen.
Es bietet sich somit an, ein standardisiertes Evaluationsverfahren zu verwenden, das sich in der Quantität und Qualität der zugrundeliegenden Daten nicht unterscheidet und vorgibt, wie viele Ereignisse vorhanden sind und welche Ereignisse gefunden werden können. Die Idee ist hierbei, den Twitterstrom zu simulieren. Als Grundlage dient hier der Gardenhose Zugang. In diesem erzeugten Twitterstrom werden dann künstlich Ereignisse integriert. Somit sind die Ereignisse, die der Ereigniserkennungsalgorithmus erkennen muss, bekannt und die Datenbasis weist im Hinblick auf die Größe und Qualität eine konsistente Struktur auf. Dieses Verfahren eignet sich somit als einheitliche Evaluationsmethode, um verschiedene Ereigniserkennungsalgorithmen miteinander vergleichen zu können. Das hier vorgestellte Verfahren, um den Twitterstrom zu simulieren, wird Twistor (**Twi**tter **St**ream Simulat**or**) genannt.

2 Verwandte Arbeiten

Da es eine Fülle von verschiedenen Evaluationsmethoden für die Ergebnisse von Ereigniserkennungsalgorithmen gibt, wird hier nur eine kleine Übersicht gegeben. Die Evaluationsmethoden werden in vier Gruppen aufgeteilt.
Eine von diesen Gruppen sind Fallstudien. So führte Corney [CMG14] eine Studie über die Ereignisse (z.B. Treffer erzielt) während eines Fußballspiels durch. Eine andere Gruppe sind einzelne Evaluierungen, die durchgeführt wurden, damit verschiedene Parameter angepasst werden können, um so bessere Ergebnisse zu erzielen. Ein Beispiel hierfür stammt von Ifrim [GI14]. Hier war ein Bestandteil der Evaluation zu untersuchen, wie sich die Veränderung der Parameter auf die Ereigniserkennung auswirkt. Die einzelnen Verfahren in dieser Gruppe lassen sich nur schlecht miteinander vergleichen. Die dritte Gruppe bilden komparative Evaluationen. So verglichen z.B. Weng und Lee [WL11] ihr Verfahren mit einer LDA [BNJ03]. Eine andere Gruppe von Evaluationen basiert auf Benutzerstudien. Hier werden die Resultate von Menschen bewertet. Eine Benutzerstudie zur Evaluation von Ergebnissen führte z.B. Thapen [TSH15] durch.
Die Heterogenität der verschiedenen Evaluationsmethoden macht es schwierig, die ver-

schiedenen Ereigniserkennungsalgorithmen miteinander zu vergleichen. Um dieses Problem zu lösen, implementierte Weiler [WGS15] verschiedene Algorithmen zur Ereigniserkennung in das gleiche Framework (Niagarino[3]) und unterzog die einzelnen Verfahren einer einheitlichen Evaluation.

3 Twistor – Simulation des Twitterstroms

3.1 Analyse des Twitterstroms

Da ein Großteil der Twitternachrichten oft wenig gehaltvoll ist [KH11], wird der Twitterstrom vor allem durch ein Grundrauschen (zur Ereigniserkennung nicht verwendbare Twitternachrichten) charakterisiert. Nach der Analyse der Twitterdaten, die mittels des Gardenhose-Zugang gesammelt wurden, kann festgestellt werden, dass die Verteilung der Wörter in den Daten innerhalb eines Tages einem ähnlichem Muster folgt.

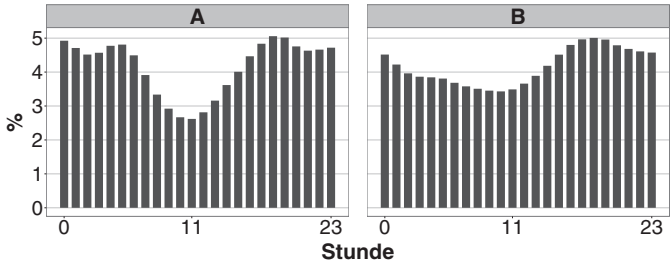

Abb. 1: Ein Beispiel für eine typische Verteilung der Wörteranzahl über einen Tag.

In Abbildung 1 ist auf der linken Seite (A) die relative (zu allen Wörtern pro Tag) Anzahl aller Wörter pro Stunde abgetragen. Hier ist zu erkennen, dass die Anzahl der Wörter zur Mitte des Tages abnimmt. Im weiteren Verlauf erhöht sich dann die Anzahl der Wörter wieder. Auf der rechten Seite (B) ist die relative (zu allen einmaligen Wörtern pro Tag) Anzahl an einmaligen Wörtern pro Stunde gegeben. Bei den einmaligen Wörtern (B) ist ein ähnlicher Effekt wie bei (A) auszumachen, der aber nicht ganz so stark ausgeprägt ist.

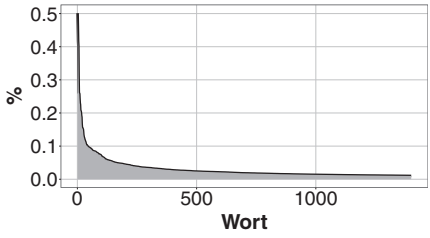

Abb. 2: Eine typische Verteilung der Worthäufigkeit über einen Tag betrachtet.

In Abbildung 2 ist die relative (zu allen Wörtern pro Tag) Häufigkeit aller Wörter über den ganzen Tag zu sehen. Da es für einen Tag über 3,5 Millionen Wörter gibt, sind hier

[3] http://www.informatik.uni-konstanz.de/grossniklaus/software/niagarino/

nur die Werte, die über 0,01 % liegen, abgetragen. Dies entspricht 1400 Wörtern. Es ist zu erkennen, dass einige wenige Wörter im Vergleich zu allen anderen Wörtern relativ häufig vorkommen (linke Seite der Verteilung). Der Großteil der Wörter tritt aber im Vergleich zu allen Wörtern eher selten auf.

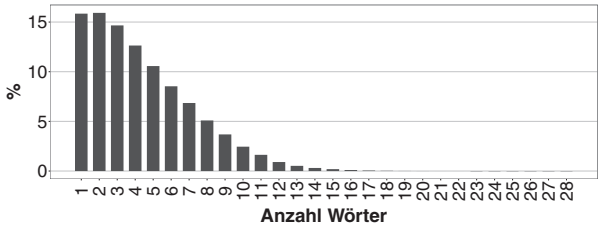

Abb. 3: Eine typische Verteilung der Wortanzahl in Twitternachrichten über einen ganzen Tag.

In Abbildung 3 ist die relative (zu allen Twitternachrichten pro Tag) Anzahl der Wörter in Twitternachrichten über einen ganzen Tag zu erkennen. Erwähnt werden muss, dass sog. Stoppwörter wie z.B. „the“, „also“ usw. oder auch Wörter, die nur aus Zahlen bestehen, URLs etc. gefiltert werden, da diese Wörter von den Ereigniserkennungsverfahren nicht verwendet werden. Es ist zu sehen, dass Twitternachrichten mit einem und zwei Wörter am häufigsten vorkommen (ca. 15 % aller Twitternachrichten). Je mehr Wörter in den Twitternachrichten vorkommen, desto geringer wird auch die Anzahl dieser Twitternachrichten.

Das letztendliche Ziel ist es, das Grundrauschen zu simulieren, indem eine ähnliche Verteilung der Wörter erzeugt wird.

3.2 Erzeugung des Twitterstroms

3.2.1 Überblick

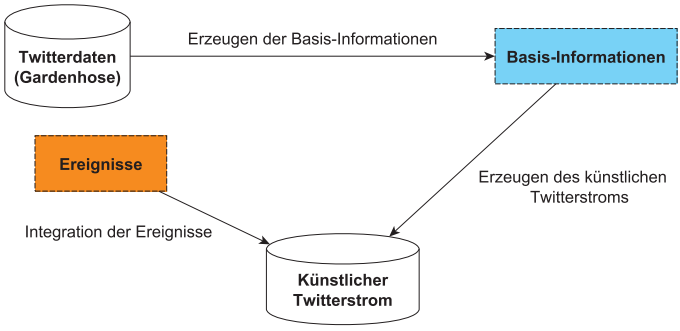

Abb. 4: Eine Übersicht über den Ablauf von Twistor.

Zuerst werden die Basis-Informationen aus den Twitterdaten ermittelt. Die Basis-Informationen geben Auskunft über die Verteilung der Twitterdaten und müssen nur einmal erzeugt

werden. Dann muss ausgewählt werden, welche Ereignisse in den künstlichen Twitterstrom integriert werden sollen. Die Ereignisse basieren auf realen Gegebenheiten und werden anhand von Wörtern, die subjektiv passend zum Ereignis sind, identifiziert. Von diesen Wörtern ist die Information vorhanden, wie häufig diese in den künstlichen Twitterdaten auftreten müssen. Die Ereignisse werden in den künstlichen Twitterstrom integriert, indem die Häufigkeiten der jeweiligen Wörter in den Strom abgebildet werden.

3.2.2 Ermitteln der Basis-Informationen

Um den Twitterstrom zu verarbeiten, wird dieser vom Ereigniserkennungsverfahren in Zeitfenster aufgeteilt. Alle Twitternachrichten, die innerhalb des Zeitfensters enthalten sind, werden gesammelt und verarbeitet. Dies wird für alle aufeinanderfolgenden Zeitfenster so durchgeführt. Die Größe des Zeitfensters kann beliebig gewählt werden. Liegen z.B. insgesamt Twitterdaten von einer Stunde vor und als Zeitfenster wird 15 Minuten gewählt, ergeben sich vier Zeitfenster. Bei der Erzeugung des Twitterstroms ist wichtig zu beachten, dass ein Wort z.B. im ersten und vierten Zeitfenster vorkommen kann, dazwischen aber nicht unbedingt.
Als Basis für die Erzeugung des künstlichen Twitterstroms dienen Daten, die sich über einen zufällig ausgewählten Tag (24 Stunden) erstrecken. Diese Daten wurden anhand des Gardenhose-Zugang gesammelt und repräsentieren 10 % des Twitterstroms. Die 24 Stunden dieser Daten werden in 1-Minuten-Zeitfenster aufgeteilt. Für jedes 1-Minuten-Zeitfenster werden zwei charakteristische Merkmale erfasst. Erstens wird ermittelt, wie häufig jedes Wortes pro 1-Minuten-Zeitfenster vorkommt. Zweitens wird die Verteilung der Anzahl der Wörter in den Twitternachrichten pro 1-Minuten-Zeitfenster festgestellt (z.B. Twitternachrichten mit einem Wort kommen 20 Mal vor, mit vier Wörtern 60 Mal und so weiter). Nach Abarbeitung aller 1-Minuten-Zeitfenster wird jedes vorhandene Wort durch ein künstlich erzeugtes Wort ersetzt. Die so erzeugten Informationen werden abgespeichert und dienen als Basis-Informationen. Da die Basis-Informationen abgespeichert werden, müssen diese auch nur einmal erzeugt werden. Möchte man einen anderen Tag als Grundlage für die Basis-Informationen hinzuziehen, so müssen diese neu erzeugt werden.

3.2.3 Erzeugen des künstlichen Twitterstroms aus den Basis-Informationen

Aufbauend auf den Basis-Informationen kann nun der künstliche Twitterstrom erzeugt werden. Hierbei muss zunächst angegeben werden, welche Größe das Zeitfenster haben soll. Die Größe des Zeitfensters sollte der Einstellung des Ereigniserkennungsverfahren entsprechen. Die Mindestgröße für das Zeitfenster beträgt eine Minute. Außerdem muss angegeben werden, wie viel Stunden an Daten erzeugt werden sollen. Der maximale Wert hierfür ist 24 Stunden.
Bevor eine genaue Beschreibung des Algorithmus gegeben werden kann, müssen einige Definitionen getroffen werden.

Sei $W = \{w_1, \ldots, w_n\}$ alle Wörter aus den Basis-Informationen. $I = \{i_1, \ldots, i_m\}$ entspricht allen vorhandenen 1-Minuten-Zeitfenstern. Die Häufigkeit eines Wortes in einem gegebenen Zeitfenster wird mit $\theta_{k,l}$ angegeben, wobei k der Index für das Wort und l der Index für das Zeitfenster ist. Es gilt $1 \leq k \leq |W|$ und $1 \leq l \leq |I|$.
Es sei mit $C = \{c_1, \ldots, c_p\}$ die Menge gegeben, bei der jedes Element eine mögliche Wortanzahl einer Twitternachricht repräsentiert. Mit $\nu_{q,t}$ wird angegeben, wie oft eine Twitternachricht mit q Wörtern im Zeitfenster t vorkommt, wobei $1 \leq q \leq \max(C)$ und $1 \leq t \leq |I|$ gilt.
Die Beschreibung des Algorithmus zur Erzeugung des künstlichen Twitterstroms soll nachfolgend in Pseudocode gegeben werden.

Algorithmus 1: Erzeugung des künstlichen Twitterstroms

Eingabe: W, I, $\theta_{k,l}$, $\nu_{q,t}$

```
foreach i ∈ I do
    tweets ← List()
    foreach ν ∈ ν_{q,i} do
        wordAmount ← q
        tweetAmount ← ν
        for j ← 1 to tweetAmount do
            words ← List()
            for l ← 1 to wordAmount do
                word ← ExtractWord(W, θ_{k,i}, words)
                words ← Insert(word)
            tweets ← Insert(words) // mehrere Wörter repräsentieren einen Tweet
    Output(tweets) // schreibe oder streame Tweets
```

Um das passende Wort aus der Menge von Wörtern (W) für eine Twitternachricht auszuwählen, wird folgender Algorithmus verwendet:

Algorithmus 2: ExtractWord

Eingabe: W, $\theta_{k,l}$, words

```
word ← null
for i ← 1 to |W| do
    if θ_{i,l} > 0 and W[i] IsNotInList(words) then
        word ← W[i]
        θ_{i,l} ← θ_{i,l} − 1
        break
/* word kann nicht null werden, da die Verteilung der Wörter so angelegt ist,
   dass immer ein Wort gefunden wird */
return word
```

Zu beachten ist, dass das in Algorithmus 1 angegebene Verfahren in der dargestellten Form nur für 1-Minuten-Zeitfenster definiert ist. Möchte man z.B. 5-Minuten-Zeitfenster verwenden, so müssen immer zuerst fünf 1-Minuten-Zeitfenster zusammengefasst werden, bevor der Algorithmus angewendet werden kann.

3.3 Definition und Identifikation von Ereignissen

Neben dem Erzeugen des Grundrauschens des Twitterstroms müssen Ereignisse künstlich in diesen integriert werden. Um dieses umsetzen zu können, muss ein Ereignis erst einmal charakterisiert werden. Vielen Ereigniserkennungsverfahren wie [WL11, Co12, Kl02, Fu05] liegt zugrunde, dass diese an- und absteigende Häufigkeiten von Wörtern in den Twitternachrichten verwenden bzw. weiterverarbeiten, um Ereignisse ausfindig zu machen. Um diese Häufigkeiten zu repräsentieren, wird das IDF-Signal [SB88] der Wörter (engl. *Inverse Document Frequency*) verwendet. Dies hat den Vorteil, dass Wörter, die in fast jeder Twitternachricht oder nur selten vorkommen, eine geringere Bedeutung beigemessen wird als Wörtern, die nur in wenigen Twitternachrichten oft auftreten. Wörter, die in wenigen Twitternachrichten häufig vorkommen, können ein gerade stattfindendes Ereignis beschreiben. Noch zu erwähnen ist, dass je öfter ein Wort auftritt, desto kleiner sein IDF-Wert wird (und nicht größer).
Um eine möglichst genaue Definition von Ereignissen wie z.B. Naturkatastrophen oder Sportveranstaltungen anhand des IDF-Signals treffen zu können, wurden reale Ereignisse herangezogen und näher analysiert. Es folgen nun zwei ausgewählte Beispiele. Die Wörter, von denen das IDF-Signal erfasst wurde, wurden subjektiv passend zum Ereignis ausgewählt.

3.3.1 Beispiele

Das erste Ereignis ist das Tor von Mario Götze im Finale der Weltmeisterschaft von 2014. Dieses ist am 13.07.14 um 21:24 Uhr (GMT) gefallen. Als passende Wörter für dieses Ereignis wurden „goal“, „goetze“ und „scored“ ausgewählt. Abbildung 5 zeigt den Verlauf der IDF-Werte für die drei Wörter. Es ist gut zu erkennen, dass ab ca. 21:25 Uhr die IDF-Werte für alle drei Wörter sehr viel kleiner werden. Dies bedeutet, dass die Wörter häufiger genannt werden.

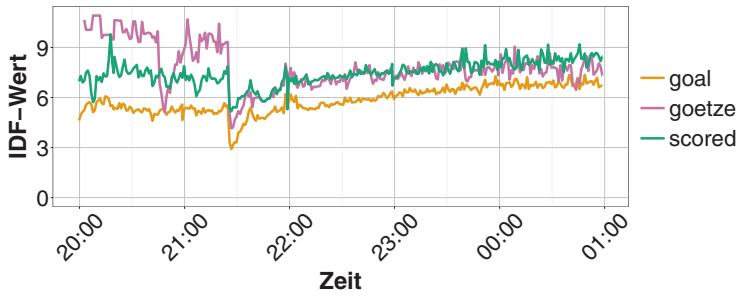

Abb. 5: Die IDF-Werte für das Tor im Finale der Weltmeisterschaft von 2014.

In den fortlaufenden Stunden steigen die IDF-Werte dann langsam an. Die drei Wörter werden somit immer weniger genannt, sodass dieses Ereignis an Bedeutung verliert.

Das zweite Ereignis ist die Papst-Wahl am 13.03.2013 um 18:06 Uhr (GMT). Als Wörter, die das Ereignis beschreiben, wurden „habemus“ und „papam“ ausgewählt.

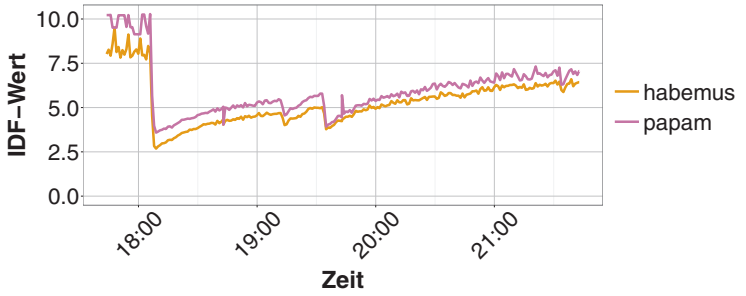

Abb. 6: Die IDF-Werte für die Papst-Wahl von 2013.

In Abbildung 6 ist das IDF-Signal für die beiden Wörter abgebildet. Es zeigt sich, dass es ab ca. 18:07 Uhr zu einem Abfall der IDF-Werte kommt. Die beiden Wörter werden zu diesem Zeitpunkt deutlich häufiger genannt als vorher. Im weiteren Verlauf steigen die IDF-Werte langsam an. Die zwei Wörter treten immer seltener auf und die Relevanz des Ereignisses sinkt.

Es zeigt sich, dass Ereignisse in den Twitterdaten sich dadurch auszeichnen, dass der Verlauf der IDF-Werte der zum Ereignis passenden Wörter einen starken „Knick“ (s. Abbildung 5 ca. 21:25 Uhr und Abbildung 6 ca. 18:07 Uhr) aufweist.

3.4 Integration der Ereignisse

Um Ereignisse in den künstlichen Twitterstrom zu integrieren, werden die Daten von realen Ereignissen herangezogen. Das bedeutet, dass der Verlauf der IDF-Werte von Wörtern, die für das Ereignis relevant sind, in den künstlichen Twitterstrom abgebildet wird. Auf diese Weise entsteht in dem künstlichen Twitterstrom ein reales Abbild eines Ereignisses. Um den IDF-Verlauf von diesen Wörtern zu imitieren, muss die Anzahl an Twitternachrichten und die Auftretenshäufigkeit der Wörter in ein bestimmtes Verhältnis gesetzt werden.
Zunächst die Definition des IDF-Wertes:

$$\mathrm{idf}(w) = \log\left(\frac{N}{n_w}\right) \tag{1}$$

N steht für die Anzahl aller Twitternachrichten, w entspricht dem Wort, von dem der IDF-Wert berechnet wird und n_w ist die Anzahl aller Twitternachrichten, die w enthalten. Da der IDF-Wert bekannt ist (dieser soll imitiert werden) und auch die Anzahl der Twitternachrichten durch den künstlichen Twitterstrom vorgegeben wird, muss die Anzahl an

Twitternachrichten, die w enthalten, berechnet werden. Umstellen der Gleichung 1 nach n_w:

$$n_w = \frac{N}{e^{\mathrm{idf}(w)}} \tag{2}$$

Durch Berechnen von n_w kann nun angegeben werden, wie oft das Wort w in den Twitterdaten auftreten muss, damit der entsprechende IDF-Wert erreicht wird.

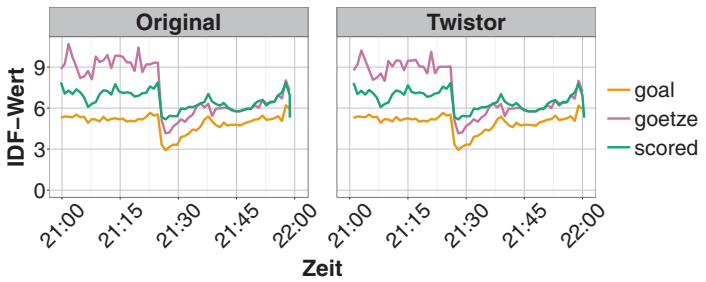

Abb. 7: Die IDF-Werte für das WM-Tor im Finale von 2014.

In Abbildung 7 ist beides mal das Ereignis des WM-Tors im Finale von 2014 mit den entsprechenden IDF-Werten zu sehen. Links (Original) ist der Verlauf der IDF-Werte in den originalen Twitterdaten dargestellt. Rechts (Twistor) ist das IDF-Signal abgebildet, nachdem es in den künstlichen Twitterstrom integriert wurde. Vergleicht man alle IDF-Werte des Originals mit den Twistor Werten so beträgt die durchschnittliche Abweichung gerundet 0.04. Die Unterschiede sind also minimal.

Auch lassen sich mehrere Ereignisse mit den entsprechenden IDF-Signalen innerhalb des künstlichen Twitterstroms an verschiedenen Zeitpunkten abbilden.

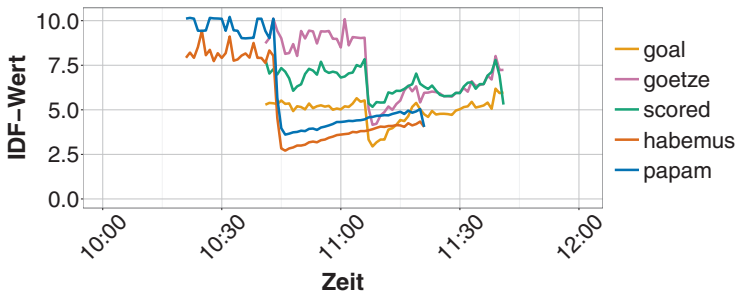

Abb. 8: Das Ereignis des WM-Tors im Finale von 2014 und der Papst-Wahl von 2013.

In Abbildung 8 wurde das Ereignis der Papst-Wahl von 2013 („habemus“ und „papam“) um 20 Minuten und das Ereignis des WM-Tors im Finale von 2014 („goal“, „goetze“ und „scored“) um 40 Minuten verschoben.

Die IDF-Signale der jeweils zum Ereignis passenden Wörter sind gespeichert und müssen nur einmal gebildet werden.

4 Abschluss und Ausblick

Mit Twistor lässt sich die Evaluation von Algorithmen zur Ereigniserkennung besser bewältigen, da die Ereignisse, die zu finden sind, vorgegeben werden können. Es bietet somit eine zuverlässigere Variante als z.B. das Abgleichen der Ergebnisse des Ereigniserkennungsverfahren mit einer Nachrichtenseite wie Reuters. Es lässt sich damit eine standardisierte Evaluation durchführen.
Mittlerweile untersagt Twitter die Weitergabe von Twitterdaten, was dazu führt, dass jeder Forscher seine eigenen Twitterdaten sammeln muss. So benutzen alle ihre eigenen Daten, die sich von der Qualität und Größe unterscheiden. Dieser Umstand führt dazu, dass die Vergleichbarkeit zwischen den Ergebnissen der Ereigniserkennungsalgorithmen nicht gewährleistet ist. Auch hier kann Twistor helfen, da keine Daten weitergegeben werden müssten.
Außerdem kann Twistor z.B. auch direkt als Input-Quelle in einem Datenstrommanagementsystem genutzt werden. Üblicherweise werden in einem Datenstrommanagementsystem zur Analyse des Twitterstroms nicht der Twitterstrom an sich, sondern Textdateien benutzt. Um den Twitterstrom zu simulieren, kann Twistor benutzt werden.
Das Projekt ist in dieser Form noch nicht abgeschlossen. Als Grundlage für die Erzeugung der Basis-Informationen dient der Gardenhose-Zugang. Dieser liefert 10 % des gesamten Twitterstroms. Somit produziert auch Twistor 10 % des realen Twitterstroms. Zukünftig ist geplant, dass die Anzahl der Twitternachrichten z.B. auf 100 % hochskaliert werden kann. Bei einer Hochskalierung der Anzahl der Twitternachrichten bleibt die Verteilung der Wörter auf dem Stand von 10 %, da keine weiteren Informationen über die Verteilung der Wörter vorliegen.
Weiterhin ist auch geplant, dass Ereignisse, die in den künstlichen Twitterstrom integriert werden, „abgeschwächt“ werden können. Dies bedeutet, dass der charakteristische „Knick“ im Verlauf der IDF-Werte eines Ereignisses in die Länge gezogen wird, sodass der drastische Abfall der IDF-Werte reduziert wird.
Außerdem soll auch ein synthetisches Ereignis, das keinem realen Ereignis nachempfunden ist, definiert und in den künstlichen Twitterstrom integriert werden können.
Trotz aller Vorteile, die Twistor gegenüber den schon vorhandenen Evaluationsmethoden liefern würde, muss natürlich die Qualität und Brauchbarkeit des simulierten Stroms überprüft werden. Dies soll geschehen, indem die Ergebnisse der Ereigniserkennungsalgorithmen, die einmal durch den künstlichen Twitterstrom und ein anderes Mal durch die Originaldaten (Gardenhose) entstanden sind, miteinander verglichen werden. Zur Evaluation sollen verschiedene Ereigniserkennungsverfahren hinzugezogen werden. Wie die Evaluation im Detail aussehen soll, steht zu diesem Zeitpunkt aber noch nicht fest.

Danksagung

Einen großen Dank gebührt Michael Grossniklaus und Andreas Weiler für die Betreuung und Unterstützung bei der Anfertigung dieser Arbeit. Die Forschung zu Erlangung der in diesem Beitrag präsentierten Forschungsergebnisse wird teilweise gefördert von der Deutschen Forschungsgemeinschaft (DFG), Grant No. GR 4497/4: Adaptive and Scalable Event Detection Techniques for Twitter Data Streams.

Literaturverzeichnis

[BNJ03] Blei, David M.; Ng, Andrew Y.; Jordan, Michael I.: Latent Dirichlet Allocation. J. Mach. Learn. Res., 3:993–1022, 2003.

[CMG14] Corney, David; Martin, Carlos; Göker, Ayse: Spot the Ball: Detecting Sports Events on Twitter. In (de Rijke, Maarten; Kenter, Tom; de Vries, Arjen P.; Zhai, ChengXiang; de Jong, Franciska; Radinsky, Kira; Hofmann, Katja, Hrsg.): Advances in Information Retrieval, Jgg. 8416 in Lecture Notes in Computer Science, S. 449–454. Springer International Publishing, 2014.

[Co12] Cordeiro, Mário: Twitter Event Detection: Combining Wavelet Analysis and Topic Inference Summarization. In: Proc. Doctoral Symposium on Informatics Engineering (DSIE). S. 123–138, 2012.

[Fu05] Fung, Gabriel Pui Cheong; Yu, Jeffrey Xu; Yu, Philip S.; Lu, Hongjun: Parameter Free Bursty Events Detection in Text Streams. In: Proc. Intl. Conf. on Very Large Data Bases (VLDB). S. 181–192, 2005.

[GI14] Georgiana Ifrim, Bichen Shi, Igor Brigadir: Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering. In: Proc. Workshop on Social News on the Web (SNOW) in conjunction with Intl. Conf. Companion on World Wide Web (WWW). S. 33–40, 2014.

[KH11] Kaplan, Andreas M.; Haenlein, Michael: The Early Bird Catches the News: Nine Things You Should Know about Micro-Blogging. Business Horizons, 54(2):105–113, 2011.

[Kl02] Kleinberg, Jon: Bursty and Hierarchical Structure in Streams. In: Proc. Intl. Conf. on Knowledge Discovery and Data Mining (KDD). S. 91–101, 2002.

[SB88] Salton, Gerard; Buckley, Christopher: Term-Weighting Approaches in Automatic Text Retrieval. Inf. Process. Manage., 24(5):513–523, 1988.

[SOM10] Sakaki, Takeshi; Okazaki, Makoto; Matsuo, Yutaka: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In: Proc. Intl. Conf. on World Wide Web (WWW). S. 851–860, 2010.

[TSH15] Thapen, Nicholas A.; Simmie, Donal Stephen; Hankin, Chris: The Early Bird Catches the Term: Combining Twitter and News Data for Event Detection and Situational Awareness. CoRR, abs/1504.02335, 2015.

[WGS15] Weiler, Andreas; Grossniklaus, Michael; Scholl, Marc H.: Evaluation Measures for Event Detection Techniques on Twitter Data Streams. In: Proc. British Intl. Conf. on Databases (BICOD). S. 108–119, 2015.

[WL11] Weng, Jianshu; Lee, Bu-Sung: Event Detection in Twitter. In: Proc. Intl. Conf on Weblogs and Social Media (ICWSM). S. 401–408, 2011.