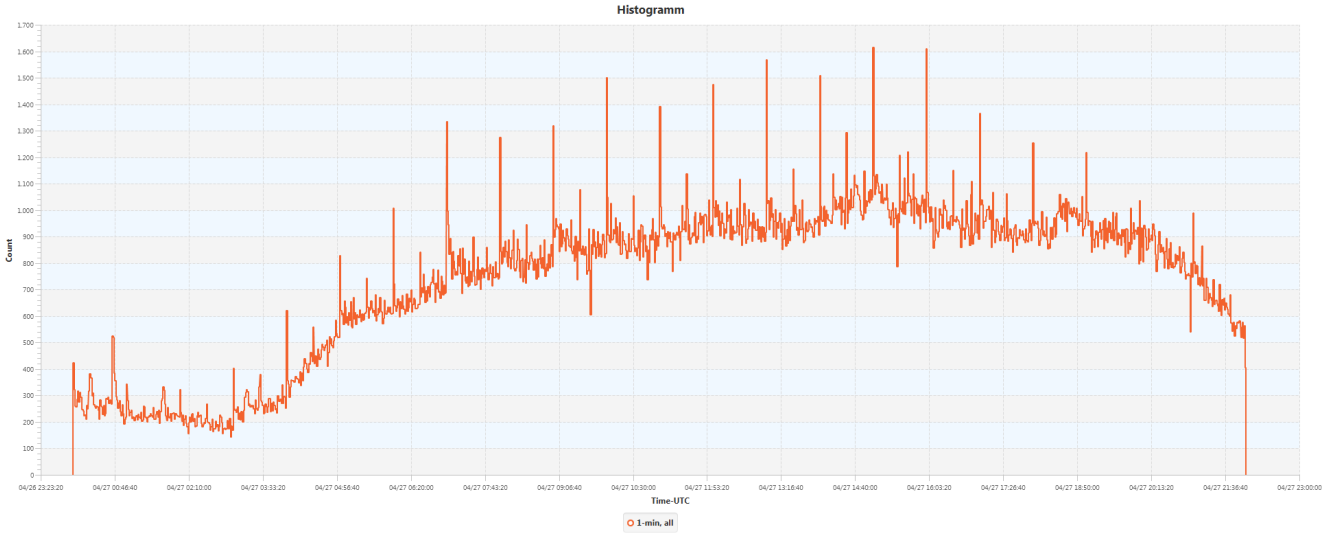
Free trade can be wonderful if you have the power of nuclear weapons.
(@Deepdrumpf)

Highland Analytica oder der Twitter-Bot Vanellus:

Die Berichte über Cambridge Analytica haben mich inspiriert, selbst einen Twitter-Bot namens ***Vanellus*** vulgo ***der Kiebitz*** zu schreiben. Er hält bisher aber entgegen seinen Namen bis auf Testtweets das Maul. Vanellus sammelt alle deutschsprachigen Tweets (rund 1 Million/Tag) und analysiert diese nach einer Reihe von Kriterien.
Um einen Bot zu schreiben, muss man sich zunächst als Benutzer bei Twitter anmelden. In einem zweiten Schritt füllt man ein Formular für Bots aus und erhält von Twitter Zugangs-Schlüssel für das Application Programming Interface (API). Mit diesem API kann man Tweets senden und aus der Twitter-Datenbank Tweets herunter laden. Tatsächlich schlägt man sich nicht direkt mit den Feinheiten dieses APIs herum. Alle wichtigen Programmiersprachen haben eigene Bibliotheken, die den Umgang damit sehr erleichtern. Ich verwende für Vanellus die Programmiersprache Java, weil meine Trading Software *CashBot* in Java läuft und ich einige Teile von CashBot direkt übernehmen konnte. Java gehört zu den wichtigsten Programmiersprachen und hat mit twitter4J eine ausgereifte Twitter-Library. Ich bin nicht der Erste und Einzige, der auf diese Idee gekommen ist.

Vanellus hat sich nach kurzer Zeit bereits geklont. Es gibt inzwischen A- und B-Vanella, sowie C- und D-Vanellus. Die weibliche Form ist auf die Benutzerin Amalia_vEnz angemeldet. Vermutlich hätte Twitter keine Einwände gehabt, wenn ich die Viererbande nur auf meinem Account registriert hätte. Jeder zusätzliche Benutzer, jeder zusätzliche Tweet, sind ein Plus in der Twitter-Bilanz. Ob in einem Rechenzentrum ein paar Rechner mehr oder weniger stehen, ist auch schon wurscht.
Das Klonen war notwendig, weil mit dem freien Zugang die Datenrate auf 12 Abfragen pro Minute (mit jeweils maximal 100 Tweets) beschränkt ist. Wenn man diese Grenze verletzt bekommt man eine Zeitsperre von 15 Minuten. Wenn man etwas Geld hinlegt, kann man größere Mengen herunter saugen. Mit einem freien Zugang kann man nicht – wie geplant - alle deutschsprachigen Tweets erfassen. Es teilen sich nun 3 Bots die Arbeit auf (der 4. dient zu Testzwecken). Sie fragen Tag und Nacht im Abstand von 2,2 Sekunden die Twitter-Datenbank nach dem Muster „lang:de“ ab. Der Twitter-Server schickt alle Tweets zurück die seit der letzten Abfrage vor 2,2 Sekunden angefallen sind. Twitter kann diesen einfachen Trick leicht erkennen. Bisher hatte ich jedoch keine Probleme.

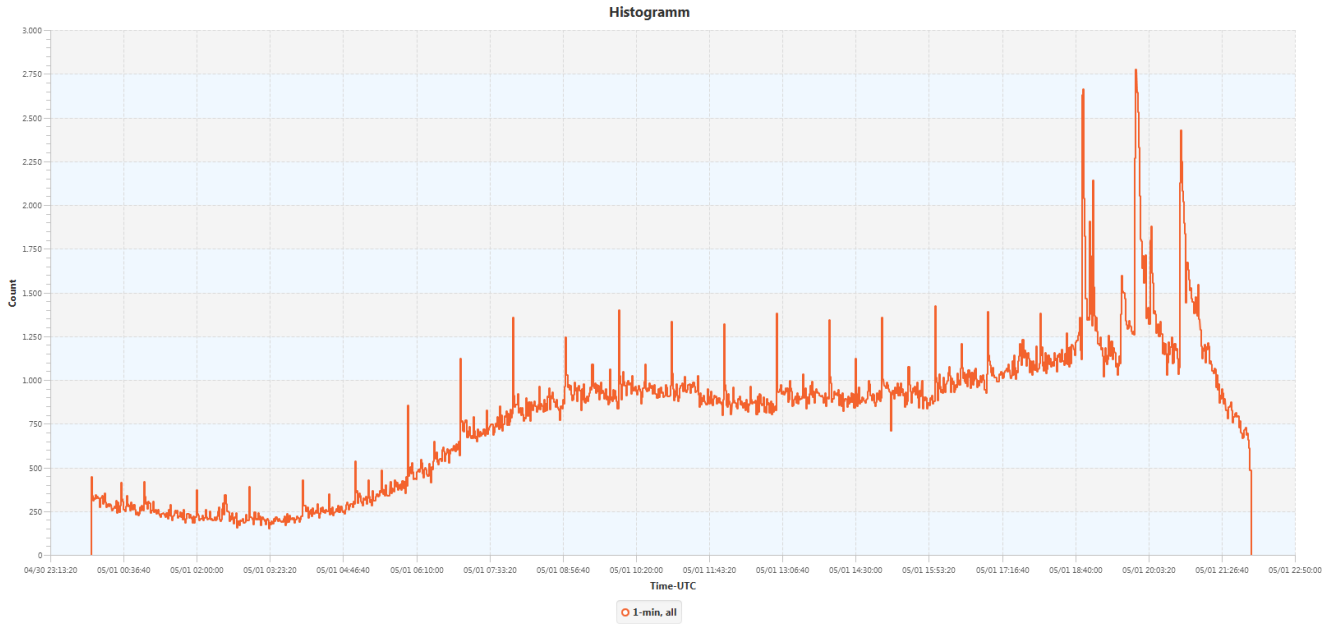


Grafik 1: Histogramm, Tweets/min am Freitag 27. April 2018

Grafik 1 zeigt den typischen täglichen Verlauf der Tweets/min. Die regelmäßigen Spitzen finden zu jeder vollen Stunde statt. Das ist die Aktivität von Nachrichtenportalen, von Bots die Turmuhr spielen und vor allem von Wetterstationen, die zu jeder vollen Stunde ihre Messdaten durchgeben. Teilweise sind es auch die Reaktionen von Usern auf diese Tweets. Der Twitter-Server liefert zwischen 0:00 und 2:00 MEST keine Daten. Twitter nützt wahrscheinlich after midnight zum Server-Service.

Am 1. Mai gab es am Abend heftige Ausschläge. Das sind die Tore im Champions-League-Schlager Real-Madrid gegen die Bayern. Man kann durch die Anzahl der eintrudelnden Tweets bereits feststellen, dass irgendwas Wichtiges passiert ist. Man weiß allerdings nicht, wer die Tore geschossen hat. Ich verwende diese Methode auch bei meiner Trading-Software. Es läuft rund um die Uhr der S&P-500-Aktienindex mit. Wenn er plötzlich nach unten sackt, ist auf der Welt was Gröberes passiert. Ich schaue anschließend am Netz nach, was es ist. Meist tauchen die Meldungen aber erst nach einiger Zeit auf. Die großen Trader sitzen näher an der Quelle und reagieren vor der Herde.

Ähnliche Effekte gab es beim Pokal-Schlager zwischen Eintracht-Frankfurt und den Bayern oder dem CL-Finale zwischen Real und Liverpool.
Anmerkung: Twitter vergibt jeden Tweet einen Zeitstempel. Dieser ist in UCT (Greenwich). Vanellus verwendet ebenfalls UCT und rechnet im Moment nicht um. Die Mitteleuropäische Sommerzeit ist UCT+2h.



Grafik 2: Histogramm, Tweets/min am Dienstag 1. Mai 2018

Als Fan von FC Erzgebirge Aue hat mich das Abstiegsderby gegen Darmstadt98 mehr interessiert. Man kann durch „hineinzoomen“ auch den Verlauf dieses Spieles und die Folgen sehr schön verfolgen. Vanellus hat zu diesem Zweck eine flexible Abfragesprache implementiert.

```
histogramm( query: "<i>#D98Aue <or> <i><Erzgebirge Aue> <or> <i>#AUE <or> <w>VAR", stepSz: 1, since: "2018-05-13", until: "2018-05-14 14:00");
```

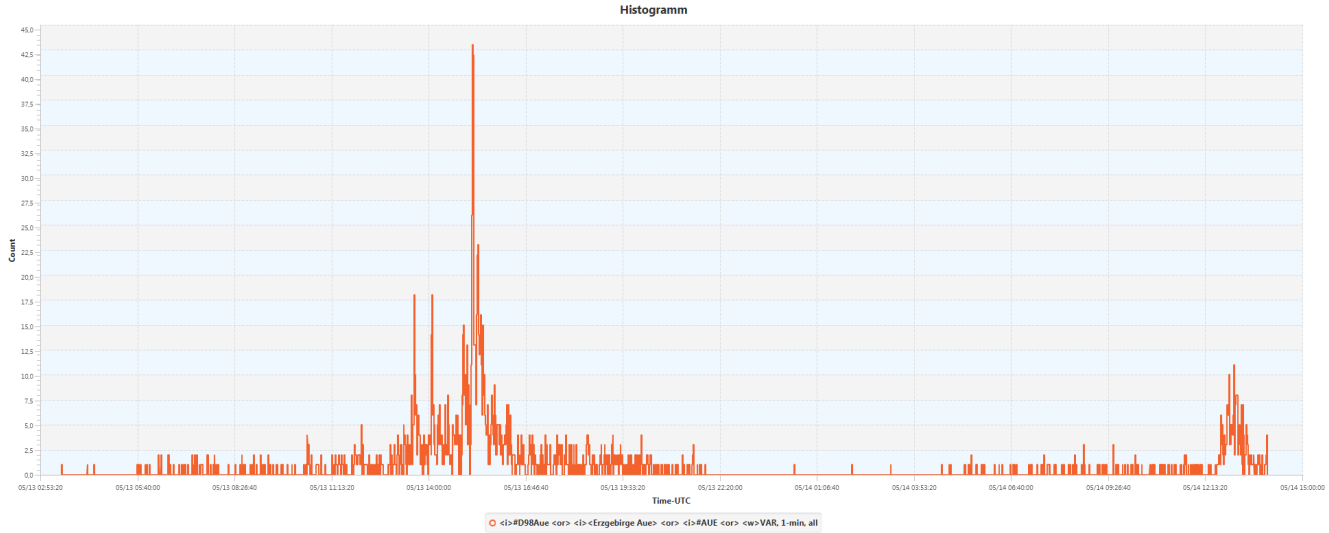
Go < >

Man kann bei Twitter nicht davon ausgehen, dass sich die Benutzer an die Regeln der Groß- und Kleinschreibung halten. Mit dem vorgestellten <i> wird auch „#aue“ oder „#aueforever“ gefunden. Der Begriff „VAR“ muss wegen des vorgestellten „<w>“ jedoch exakt so als Wort geschrieben werden. Ansonsten hat man zu viele Treffer die mit dem Video-Assistant-Referee nix zu tun haben. Der VAR spielte in diesem

Spiel eine wichtige Rolle, da Aue bereits in der 4. Minute ein klares Tor aberkannt wurde (siehe Bild). Aue gehörte zu jenen Teams, die sich zu Beginn der Meisterschaft gegen die Einführung dieses Systems ausgesprochen haben und heimste deswegen einige Häme ein.

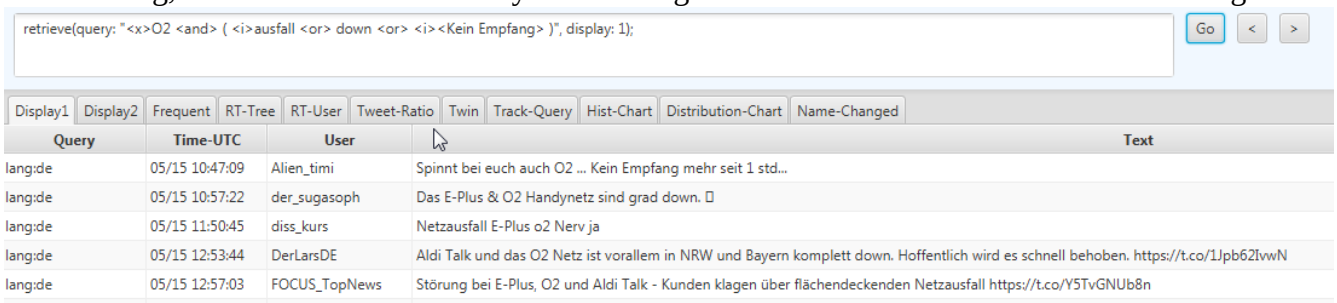
lang:de	05/13 15:17:43	9zehn100_8und70	Alles was mir zu Aue einfällt: Das kann ja wohl nicht VAR sein! #D98AUE #SkyBuli
lang:de	05/13 15:25:15	iMia_San_FCB	Eine Schande, was da in Darmstadt passiert ist. Das ist doch der Beweis, dass man VAR oder wenigstens die Torlinientechnik brauch!! #D98AUE
lang:de	05/13 15:36:35	RC_KH	Liebe Gegner von VAR oder Torlinientechnologie. Schöne Grüße aus Aue. #Liga2
lang:de	05/13 15:37:27	SveMaHe	RT @RC_KH: Liebe Gegner von VAR oder Torlinientechnologie. Schöne Grüße aus Aue. #Liga2

In Grafik 3 sieht man links das übliche „*Vorgeplänkel*“ zum Spiel. Die erste Spitze ist die Aberkennung des Tores. Die zweite Spitze ein nicht gegebener – klarer – Elfer für Aue, kurz vor Schluss hat der Schiri Aue einen weiteren Elfer vorenthalten. Die hohe Spitze markiert das Spielende und die aufgeregte Diskussion über das Spiel. Zu diesem Zeitpunkt steigen – bei jedem Spiel – die Nachrichten-Medien ein, die Twitter-User reagieren darauf. Die Aufregung legt sich aber schnell. Aue legte am Montag Protest gegen das Spiel ein und beschwerte sich über den Schiri. Das ist die Spitze rechts. Es endete wie bei Aschenputtel. Der von Karlsruhe wegen einer Verletzung nach Aue abgegebene Sören Bertram schießt mit 3 Toren seinen Ex-Klub in der Relegation vom Platz. Die „*Macht aus dem Schacht*“ bleibt in der 2. Liga.



Grafik 3: Histogramm, Erzgebirge Aue gegen Darmstadt, So./Mo. 13./14. Mai

Am Dienstag, 15. Mai fiel das O2-Handynetz für längere Zeit aus. Mit der obenstehenden Abfrage



Display1 | Display2 | Frequent | RT-Tree | RT-User | Tweet-Ratio | Twin | Track-Query | Hist-Chart | Distribution-Chart | Name-Changed

Query	Time-UTC	User	Text
lang:de	05/15 10:47:09	Alien_timi	Spinnt bei euch auch O2 ... Kein Empfang mehr seit 1 std...
lang:de	05/15 10:57:22	der_sugasoph	Das E-Plus & O2 Handynetz sind grad down. □
lang:de	05/15 11:50:45	diss_kurs	Netzausfall E-Plus o2 Nerv ja
lang:de	05/15 12:53:44	DerLarsDE	Aldi Talk und das O2 Netz ist vorallem in NRW und Bayern komplett down. Hoffentlich wird es schnell behoben. https://t.co/1Jpb62IvwN
lang:de	05/15 12:57:03	FOCUS_TopNews	Störung bei E-Plus, O2 und Aldi Talk - Kunden klagen über flächendeckenden Netzausfall https://t.co/Y5TvGNUb8n

erhält man die darunter stehenden Tweets. „<x>O2“ bedeutet, die Groß- und Kleinschreibung ist egal, aber es muss ein eigenes Wort sein. Ansonsten erhält man auch Echo2018 als Treffer. Zusätzlich muss es noch um den Zusammenbruch des Netzes gehen und z.B. nicht um günstige O2-Tarife. Bei „ausfall“ soll auch „Ausfall“, „Netzausfall“ oder „Totalausfall“ erkannt werden. Eine Feinheit dieses Ereignisses ist: Die Betroffenen können ihre Unpässlichkeit nicht direkt tweeten. Trotzdem ist der User *Alien_timi*

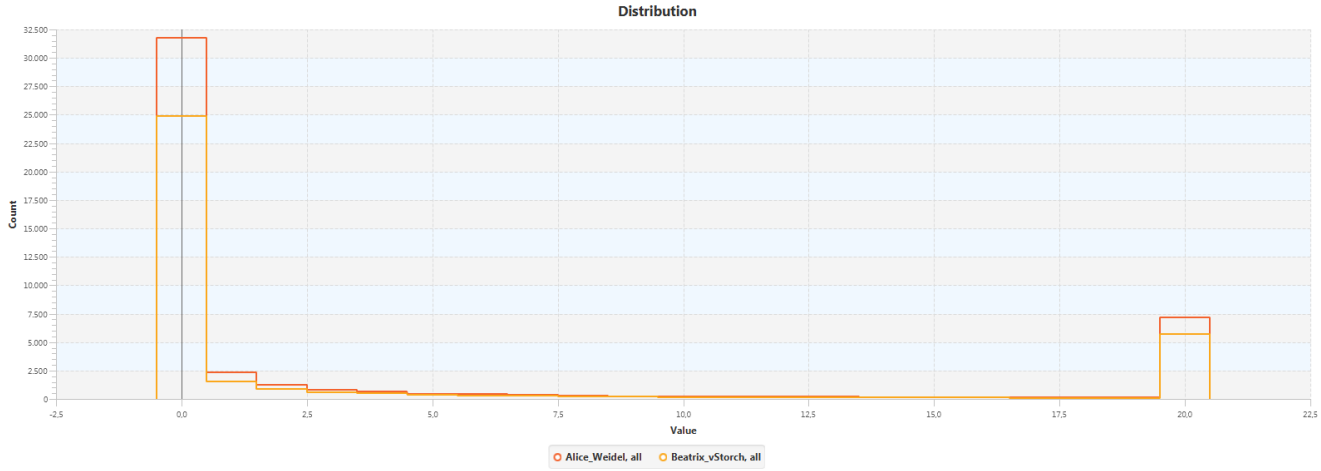
FOCUS um mehr als 2 Stunden voraus. Man kann sich wie bei Aue das Histogramm anschauen und eine Reihe von anderen Auswertungen machen. Eine gute handverlesene Datenbank und eine mächtige Abfragesprache ist jedoch die notwendige Basis um aus der Masse der Tweets Information heraus filtern zu können. Umgekehrt muss man die vollständigen Daten haben um auch speziellen Ereignisse erfassen zu können. Bei 1 Million Tweets pro Tag wird das auf Dauer auf einem normalen PC eine Herausforderung.

Spieglein, Spieglein an der Wand, wer hat die meisten Follower im Land?

Am 29. April hat die AfD Politikern Alice Weidel einen Tweet der Pollytix-Strategic-Research weiter geleitet. Sie ist vor ihrer Parteifreundin Beatrix v.Storch die Follower-Königin im AfD-Land.

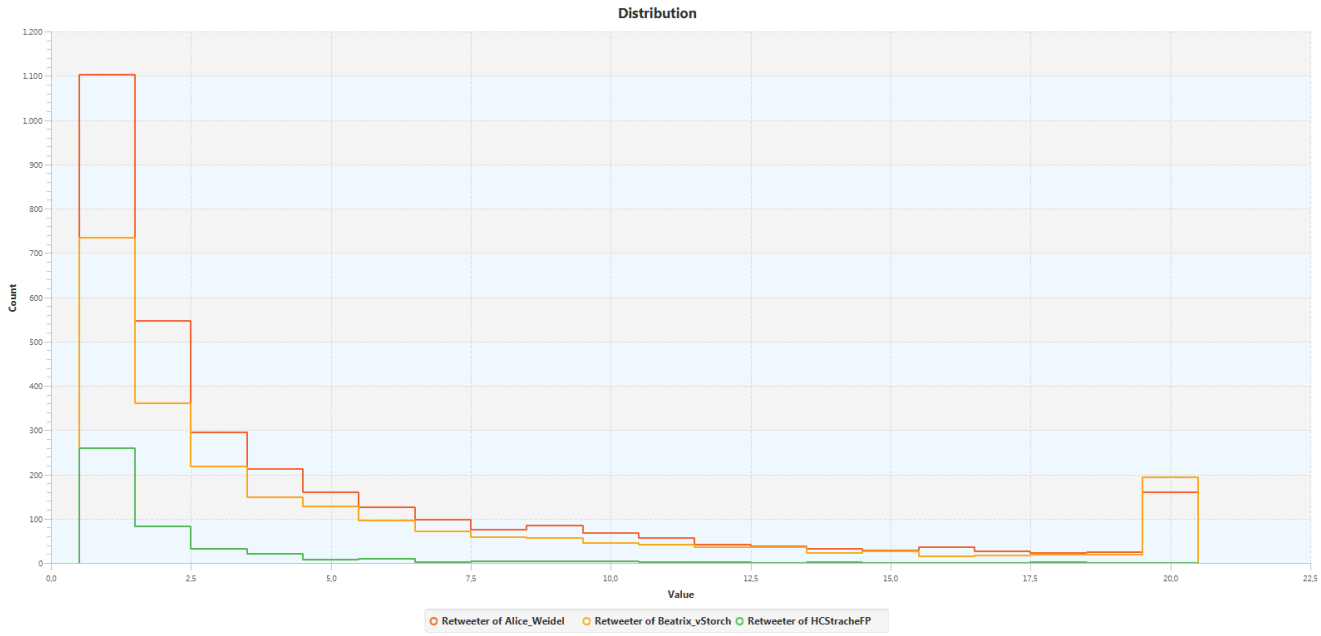
lang:de	04/29 14:28:16	Alice_Weidel	RT @pollytix_gmbh: Die #MdB's der @AfDimBundestag mit den meisten #Follower|n bei #Twitter sind @Alice_Weidel @Beatrix_vStorch @PetrBystro...

Um das fest zu stellen braucht man keinen Strategic-Research, man braucht nur auf den Twitter-Accounts der Politiker nachschauen. Das sollte ein Ferialpraktikant oder AzuBi in einer Stunde problemlos schaffen. Die ungekrönte AfD-Twitterkönigin ist die ehemalige CDU Politikerin Erika Steinbach. Aber Steinbach ist keine Abgeordnete und so hat Pollytix schon richtig gezählt.
Interessanter ist schon die Frage: Gibt es diese Follower überhaupt und leisten sie Gefolgschaft? Man kann von Twitter die Benutzer-Kennung der Follower herunter laden. Diese Kennung wird bei jedem Tweet mit geschickt. Vanellus kann so dem Treiben der Follower folgen. Für die folgenden Ergebnisse habe ich die Daten von 4 Wochen, vom Fr. 27. April bis Do. 24. Mai, verwendet. In Summe sind das gut 27 Millionen Tweets von 3,4 Millionen verschiedenen Usern.



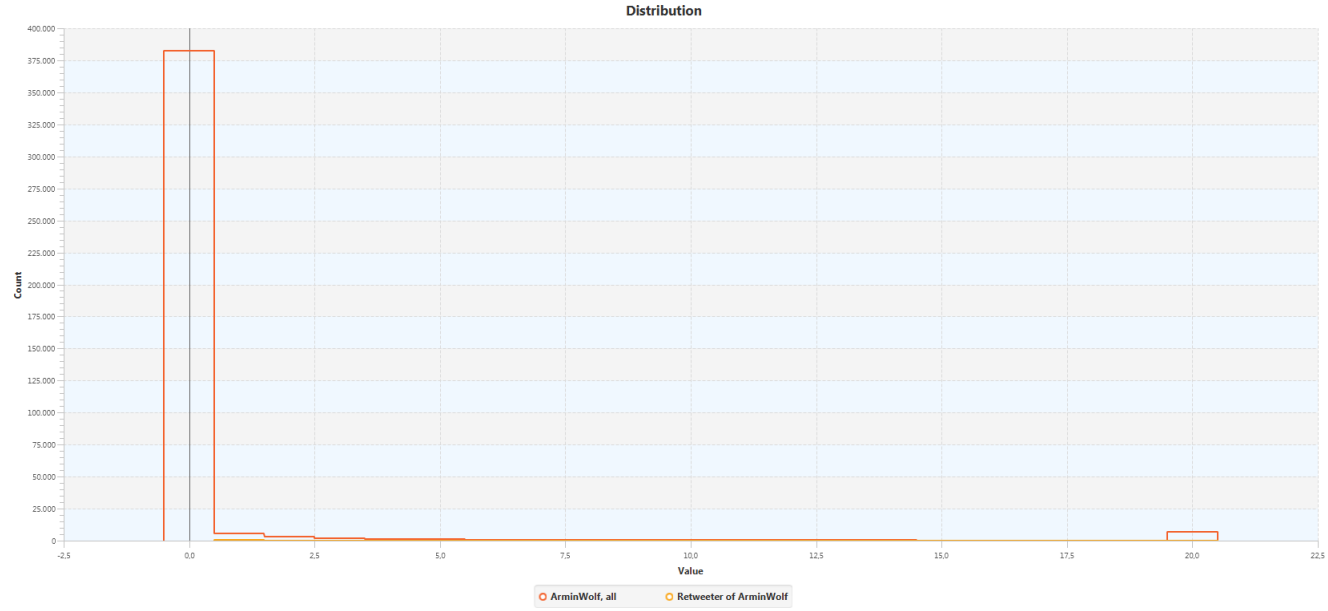
Grafik 4: Verteilung der (in)-aktiven Follower von Alice Weidel (rot) und Beatrix v. Storch (gelb)

Alice Weidel hat gut 47.000, ihre Parteifreundin Beatrix v. Storch 36.000 Follower. Grafik 4 zeigt, wie viele Tweets diese Follower in den letzten 4 Wochen versandt haben. Bei 32.000 dürfte das Interesse für Twitter nicht sehr ausgeprägt sein. Sie haben innerhalb dieses Zeitraum nichts gezwitschert. Das heißt nicht unbedingt, dass sie die Tweets von Alice Weidel gänzlich ignorieren. Aber es dürfte dieser mit Abstand größten Gruppe Twitter nicht so wichtig sein. 2.500 haben ein Mal getweetet. Ganz rechts sieht man die Gruppe der „Power-User“ die mehr als 20 Tweets versandt haben. Diese Statistik erfasst jeden Tweet. Wenn jemand bei Real gegen Bayern „Tooor“ getweetet hat, dann ist er bereits ein aktiver User. Das Verhältnis von aktiven zu inaktiven Usern ist bei Beatrix v. Storch ähnlich.
Aussagekräftiger ist wohl die Anzahl der Follower die zumindest eine Meldung retweeten. Wie man in Grafik 5 sieht, sind es bei Alice Weidel 1.100 User die das einmal machen. Die Verteilung sinkt mit höheren Anzahl von Retweets steil ab. Es gibt aber auch hier wieder eine Gruppe von 200 Power-Usern. Einige davon sind jedoch Bots die so ziemlich alles weiterleiten was ihnen in die Fänge kommt. Die Verteilung von Beatrix v. Storch ist wieder aliquot zur Follower-Zahl.



Grafik 5: Retweeter-Verteilung, Alice Weidel (rot), Beatrix v. Storch (gelb), H.C. Strache (grün)

Die grüne Linie zeigt die Verteilung der Retweets für H.C. Strache. Strache hat mit 49.000 Followern etwas mehr als Alice Weidel. Er ist jedoch im Twitter-Universum eine vernachlässigbare Größe geworden. Man muss sich um seinen Gesundheitszustand ernsthaft Sorgen machen: Soviel Kreide wie H.C. seit dem Eintritt der FPÖ in die Regierung frisst, kann nicht gesund sein. Um ja nichts Falsches zu sagen, retweetet er primär den Regierungssprecher. Darüber hinaus meldet er, dass er mit Norbert Hofer Kränze für die Opfer des Faschismus niedergelegt hat, er ist von Arik Brauers Rede zur Befreiung vom Faschismus tief berührt und sendet in verschiedensten Varianten Israel Glückwünsche zum 70. Jahrestag der Staatsgründung. Das reißt seine Fans nicht vom Stockerl. Es zeigt sehr schön das generelle Dilemma eines Populisten. Einmal in der Regierung schaut die Welt ganz anders aus.



Grafik 6: Follower (rot) und Retweeter-Verteilung (gelb) von ArminWolf (ORF).

Der offizielle Österreichische Twitter-König ist der ORF-Journalist Armin Wolf. Er hat 400.000 Follower. Twitter erlaubt – bei einem freien Zugang – das Herunterladen von 75.000 Follower-

Kennungen innerhalb von 15 Minuten. Vanellus lädt den ersten Schwung herunter, schläft 15 Minuten, lädt den zweiten Schwung, schläft 15 Minuten ... Diese Aufgabe wird nebenläufig erledigt. Der allgemeine Betrieb wird davon nicht beeinträchtigt.
Wie man in Grafik 6 sieht, sind von den 400.000 Wolf-Followern 380.000 Twitter-Zombies. Die Anzahl der Retweeter (gelb) ist in der Grafik kaum auszumachen. Die Twitter Gemeinde interessiert inzwischen Armin Wolf genauso wenig wie H.C. Strache. Möglicher Weise gibt es einen direkten Zusammenhang. Armin Wolf hat sich als wackerer Kämpfer gegen den pöhszen H.C. positioniert. Seit dieser auf dem Kreide-Trip ist, fehlt ihm die Angriffsfläche. Möglicher Weise läuft sich auf einem so flüchtigen und schnelllebigen Medium wie Twitter auch alles schnell tot.

User	Level	Retweets
Beatrix_vStorch	0	1
AfDimBundestag	1	2
JoachimPaul_AfD	1	1
BlauesWunder18	1	1
RolandTichy	1	2
BlondJedi	1	1
krk979	1	2
Dieter_Stein	1	1
MarcFelixSerrao	1	1
UbbelohdeBerlin	1	1
Hartes_Geld	1	1
AfD	1	6
AfD_SE	1	1
FamUnt	1	1
LSteinwandter	1	2
WMuhsal	1	1
ronaldglaeser	1	2
Alice_Weidel	1	2
Joerg_Meuthen	1	1
StBrandner	1	1
PetrBystronAfD	1	1
DrDavidBerger	1	1
SteinbachErika	1	1
Martin_Hess_AfD	1	1
JoanaCotar	1	1
AfDBerlin	1	2
maxotte_says	1	1
AfDKompakt	1	4
UdoHemmelgarn	1	1
GtzFrmming	1	2

User	Level	Retweets
Alice_Weidel	0	0
AfDimBundestag	1	7
AfD	1	4
GtzFrmming	1	1
pollytix_gmbh	1	1
Beatrix_vStorch	1	1
Mewinator89	1	1

Man kann sich mit Vanellus mehrere Ebenen von Retweet-Beziehungen ansehen. Level-0 ist der ausgewählte User. Level-1 sind die User die dieser retweeted, Level-2 die Benutzer, die von Level-1 Usern retweeted werden .... Man erhält so einen Retweeter-Baum. Dieser wird jedoch sehr schnell unübersichtlich und groß.
Die Screenshots zeigen links den Retweet-Baum (bis Ebene-1) von Beatrix v. Storch und rechts von Alice Weidel. Weidel geht mit Retweets – von Parteifreunden - sehr sparsam um. In größerer Anzahl retweeted sie nur die Personen neutralen Partei- und Fraktions-Tweets.
Beatrix v. Storch ist kollegialer. Sie schreckt aber auch nicht davor zurück sich selbst zu retweeten.
Bei H.C. Strache hat der Baum nur einen Stamm. Er retweeted wie bereits erwähnt ausschließlich und ausführlich „*RegSprecher_AT*“.

Twitter-Zwillinge und Bot-Erkennung:

Speziell seit dem US-Präsidentenwahlkampf 2016 gibt es eine intensive Diskussion um den Einfluss von Social-Media-Bots. Es hat auch das amerikanische Verteidigungsministerium einen Wettbewerb zur Bot-Erkennung ausgeschrieben. Es gibt verschiedene Ansätze zur Bot-Erkennung. Der klassische Versuch ist inhaltlicher Natur. Man versucht durch Spracherkennung und Sprachanalyse den Bots auf die Spur zu kommen. Der andere ist rein statistisch. Man untersucht das Zeitverhalten und auch die Tatsache, dass Bots in der Regel im Rudel auftreten. Man kann beide Ansätze verbinden. Bei Vanellus dominiert der statistische Ansatz.

Text-Zwillinge:

Unter einem Text-Zwilling versteht Vanellus zwei User die ungefähr die selbe Anzahl von Tweets senden und die eine hohe Anzahl von Tweets gemeinsam haben. Die Feststellung von Zwillingen ist für einen alten Hacker eine nette Herausforderung. Bei 3,4 Millionen Usern gibt es 5,8 Milliarden User-Paare. Wenn man jeden User mit jeden anderen direkt vergleicht, wird man nie fertig. Durch eine Reihe von Tricks geht die Berechnung bei Vanellus ziemlich flott von statten.

Auto-Zwillinge:

Zu meiner Überraschung erzeugte die Methode zur Erkennung von Zwillingen auch sogenannte Auto-Zwillinge. Ein Bot ist der Zwilling von sich selbst. Das passiert, wenn ein Bot dieselbe Nachricht

mehrmals schickt. Ein Beispiel dafür ist „*phil_de_bot*“. Dieser Bot verschickt alle 30 Minuten einen Tweet mit einem zufällig ausgewählten philosophischen Zitat. Der Zitatenschatz ist nicht sehr groß und so wiederholt „*phil_de_bot*“ regelmäßig Zitate. Es gibt Bots, die Kirchturm-Glocke spielen und jede volle Stunde die Zeit durchgeben. Auch diese Meldungen wiederholen sich periodisch. Ein anderes Beispiel sind Radiosender, die den jeweiligen Liedtitel tweeten. Aktuelle Hits werden häufig gespielt und es wiederholt sich dementsprechend der Tweet. Die Auto-Zwillinge sind mir passiert. Ich habe den Bug (Fehler) zum Feature gemacht. Man kann die Erkennung auch ausschalten bzw. nur nach Auto-Zwillingen suchen. Ich fand die Erkennung der Auto-Zwillinge am Ende durchaus interessant und habe dieses ungeplante Feature im Kode gelassen. Es gibt Bots, die sowohl Auto- als auch echte Zwillinge sind. Sie wiederholen eine Botschaft mehrmals und tun dies im Rudel.

Echte Zwillinge:

Bei echten Zwillingen handelt es sich um zwei verschiedene User. Wobei diese in der Regel ganze Rudel bilden.

148	Hashtags	#Homöopathie
12	Hashtags	#MachAuchDuMit
10	Hashtags	#machauchdumit
8	Hashtags	#homeopathy
7	Hashtags	#Homeopathy
6	Hashtags	#Globuli
4	Hashtags	#Placebo

Ein Beispiel dafür ist der Homöopathie-Rudel. Der User „*sand_iris*“ bildet einen Zwilling mit „*Ann_Walters_*“, „*Barbara_Hofer_*“, „*H_Schneider_*“ …. Diese bilden mit weiteren Usern wieder Zwillinge. Ziel dieser Gang ist die Propagierung der Homöopathie. Der Screenshot zeigt die häufigsten Hashtags des Bots „*sand_iris*“ Es hat eine gewisse Ironie, wenn Anhänger der sanften Medizin ein umfangreiches Bot-Netzwerk zur Propagierung ihrer Ansichten aufbauen.

Die mit Abstand größte Bot-Gang ist mit dem User „*Arikatytm31*“ verbunden (man könnte ein beliebig anderes Rudelmitglied als Repräsentanten auswählen). Diese Gang hat einige Hundert hoch aktiver Mitglieder. Wobei sich – und auch das ist für Bot-Gangs typisch – die Bots häufig umbenennen.

lang:de	04/27 00:13:32	Arikatytm31	RT RTLisaFitzpatrick #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 00:28:22	Arikatytm31	RT Scottdrives #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 00:46:04	Arikatytm31	RT KristinePThomas #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 00:46:09	Arikatytm31	RT glmh101 #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 00:46:12	Arikatytm31	RT mindwanderer #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 13:38:22	Arikatytm31	RT RTgbrodermann #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 13:5[illegible]23	Arikatytm31	RT sofiaberen #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 14:08:24	Arikatytm31	RT RT DizzyDortch #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 14:08:30	Arikatytm31	RT RTDizzyDortch #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC
lang:de	04/27 14:23:11	Arikatytm31	RT RT JasonMehrtens #MTVBRPETNUGGET #MTVBRSHADETAYLORKATY #PremiosMTVMiaw #MTVLAINSTAGLCAMILAC

Man erkennt sie trotzdem leicht, weil die Twitter-Kennung gleich bleibt. Der Screenshot zeigt die ersten 10 Tweets von *Arikatytm31*. Alle Tweets dieser Gang enthalten nur Hashtags. Offensichtliches Ziel ist das Hashtag-“Pumpen“. Der von dieser Gang „gepumpte“ Hashtag „*#PremiosMTVMiaw*“ hat innerhalb des Deutschen Netzes die Poleposition. Ich kenne mich in der Welt der Social Media zu wenig aus, um beurteilen zu können, welchen Zweck es hat, mit „*#PremiosMTVMiaw*“ die Poleposition zu erringen. Angeblich nimmt Twitter sozial auffällige Bots bzw. Bot-Gangs aus dem Rennen. Bei dieser sehr leicht erkennbaren Gang ist dies nicht der Fall. Bei „philo_de_bot“ oder den Bots der Radiosender die den aktuellen Titel ansagen, gibt es für Twitter keinen Grund, dagegen vorzugehen. Die Existenz eines API und ausgereifter Programmbibliotheken beweist hinlänglich, dass Bots ein akzeptierter und wichtiger Teil des Twitter Kommunikationsnetzes sind. Der Eingangs zitierte Bot DeepDrumpf hat es sogar zu einiger Medienberühmtheit gebracht. Dieser am M.I.T. entwickelte Bot baut aus den Tweets des Users „*RealDonaldTrump*“ nach einer relativ einfachen Methode neue

Texte zusammen. RealDonaldTrump eignet sich sehr gut als Vorlage. Seine Sprache ist einfach gestrickt, der Inhalt ist sprunghaft. Man bekommt das auch mit minimaler Bot-Intelligenz hin.

Fan oder Gegner, die Sentiment-Analyse:
Ich habe mich primär auf die AfD und ihre markantesten Repräsentanten konzentriert, weil diese auf Twitter sehr präsent sind und es innerhalb der User zwei klar erkennbare Lager gibt. Es ist – für einen Menschen – sehr leicht erkennbar, ob jemand die AfD mag bzw. sie verabscheut. Teilweise deklarieren die User dies bereits in ihrem User-Namen. Z.B. heißen die beiden aktivsten Gegner „KeineAFD2017“ und „ContraAfD1“. AfD Politiker hängen – wenn sie nicht so bekannt wie die beiden Damen sind – häufig ein „AfD“ an ihren Namen an. Ortsgruppen nennen sich z.B. „AfD_Frankenthal“.
Für einen Bot ist die Unterscheidung hingegen gar nicht so trivial. Die automatische Klassifikation von Benutzern ist unter dem Titel „*Sentiment Analyse*“ ein sehr heißes Forschungsgebiet. Man kann diese Klassifizierung auch für zielgerichtete Werbung einsetzen. Ein verwandtes Gebiet sind „*Recommender Systems*“. Es schlägt einem z.B. Amazon auf Grund des bisherigen Verhaltens neue Produkte vor. Der Netflix-Recommender empfiehlt Filme. Netflix hat einen mit 1 Million $ dotierten Wettbewerb zur Verbesserung dieses Systems ausgeschrieben. Diese Systeme werden von den Benutzern in der Regel als nützliches Feature und nicht als pöhsze Verführer wahr genommen. Die Techniken von Cambridge Analytica bewegten sich nach meinen Informationen innerhalb dieses allgemein bekannten Spektrums. Die Firma hat sich primär durch Angeberei ihres Chefs einen schlechten Ruf eingehandelt. Das war ihr Glück und Ende. Es war dieser Ruf und nicht die besonders fiesen Methoden die große Kunden zur Beendigung der Verträge bewogen hat.

User	Matches	Total-Tweets	Ratio%
mrstone0856	1735	1843	94
kardasiapat	1094	1491	73
DerDude72445273	1	1382	0
KeineAFD2017	2	1365	0
artepmobil	900	1139	79
ContraAfD1	3	1138	0
SchafDas	764	1096	69
torsten_luer	963	1046	92
Tinimaus1110	866	1018	85
GertWalterWolf1	780	991	78
mginpl	1	965	0
APVogt	394	951	41
m_aus_s	797	936	85
LC180666	668	928	71
e_pitzky	791	911	86
ruthlissy	809	902	89
wmoebius	7	901	0
Chat_Atkins	0	881	0
Renft1964	688	854	80
genauhinsehen	719	819	87
schnaggii	1	775	0

Die Retweet-Ratio:
Eine naheliegende Klassifizierung ist, sich die Retweets eines Benutzers anzuschauen. Wenn jemand einen AfD-Politiker Retweeted, dann kann man dies als Zustimmung werten. Der Screenshot zeigt die Ratio der AfD-Retweets innerhalb der Poweruser. Z.B. sind 69% der Tweets des Users „SchafDas“ AfD Retweets. Es ist auch eine klare Unterscheidung in Freund und Feind erkennbar. Es kommt zwar vor, dass deklarierte Gegner wie „KeineAFD2017“ AfD-Quellen retweeten. In diesem Fall will der Gegner die Ansichten eines AfD-Politikers an den Pranger stellen.
Man muss für eine derartige Untersuchung die Tweets inhaltlich filtern. Es können politisch Zerstrittene gemeinsam den Bayern die Daumen halten. Es mussten im Tweet der Begriff „AfD“ bzw. die bekanntesten Personen oder eindeutig zuordenbare Schlagwörter wie „#MerkelMussWeg“ vorkommen.
Anmerkungen: Man könnte eine ähnliche Untersuchung über die Pro- und Contra Bayern Fans machen. Es böte sich auch Rasenball Leipzig an.
Zunächst habe ich per Hand eine Liste der bekanntesten AfD-Politiker erstellt. Das war erstens mühsam und zweitens sind mir u.A. AfD-Ortsgruppen durch die Lappen gerutscht. Diese retweeten primär ihre lokalen Repräsentanten. Ich bin dann auf die glorreiche Idee gekommen, die Retweets von eindeutig erkennbaren AfD-Powerusern – konkret von „*mrstone0856*“ und „*torsten_luer*“ als Vorlage zu nehmen. Vanellus erstellt aus den Retweets dieser beiden unermüdlichen hardcore-Fans die Liste der AfD-Retweet Kandidaten. Damit hatte ich das Problem der Lokalhäuptlinge und der AfD-Leithammel

im Netz gelöst. Allerdings retweeten diese Poweruser auch Meldungen von neutralen Quellen wie SPIEGELONLINE oder NZZ. Ich habe zunächst wieder per Hand eine Liste dieser neutralen Benutzer erstellt, die nicht zur AfD Retweet-Ratio gezählt werden sollen. Das hatte einen ähnlichen Effekt wie der ursprüngliche Ansatz. Man findet immer wieder Medien oder User die man nicht zählen sollte. Bis ich auf die ebenfalls naheliegende Idee gekommen bin: Wenn jemand von den aktiven Gegnern häufig retweetet wird, dann kann man diesen im AfD Sinn ignorieren. Konkret verwende ich zu diesem Zweck die User „*Chat_Atkins*“ und „*DerDude72445273*“.
Diese Klassifizierung funktioniert auch bei Usern, die weniger aktiv sind, recht gut. Allerdings hatte die AfD_Frankenthal eine AfD-Ratio von 0%. Es handelt sich eindeutig um eine sehr linientreue Lokalorganisation. Nur hält sich – der vermutlich Ältere – Twitter Verantwortliche nicht an die informellen Twitter-Standards. Entsprechend diesem Standard schreibt man „RT @AfD Kompakt“ und nicht wie er „AfD Kompakt:“. Vanellus erkennt dieses Format nicht als Retweet. Der Versuch auch derartige Tweets als Retweet zu klassifizieren, erzeugte bei den Gegnern wieder zu viele Fehlalarme. Ich bin schlussendlich zum Entschluss gekommen: Auf diese Art und Weise ist die AfD_Frankenthal nicht zu erfassen.

lang:de	04/27 03:31:27	AfD_Frankenthal	AfD Kompakt: Bundesregierung völlig kenntnislos bei Ein- und Ausreisen https://t.co/wjcnZh0esj https://t.co/42tEL8Tsub
lang:de	04/27 03:46:27	AfD_Frankenthal	AfD Kompakt: Eklat im Potsdamer Stadtschloss https://t.co/p3vEj1Zq2R https://t.co/lBb4rjHbow
lang:de	04/27 04:01:26	AfD_Frankenthal	AfD Kompakt: Neuerliche Überprüfung von NO2-Grenzwerte herbeiführen https://t.co/sTmKkKJ9i9 https://t.co/LZYGtoNae5
lang:de	04/27 04:16:26	AfD_Frankenthal	AfD Kompakt: Stoppt Genozid an Christen im Nahen Osten https://t.co/RkoWmFSZvo https://t.co/X4Fn1Mwqqi
lang:de	04/27 04:31:26	AfD_Frankenthal	AfD Kompakt: Deutschland und Europa können Weltklima nicht retten https://t.co/gN9XBdZDAK https://t.co/5nX9cQcF9A
lang:de	04/27 10:58:05	AfD_Frankenthal	AfD Kompakt: Muslime werden antisemitische Passagen im Koran nicht freiwillig schwärzen https://t.co/1oCDtcYHUK https://t.co/VWXRdRNyOw
lang:de	04/27 16:33:03	AfD_Frankenthal	Presseschau: Wohlfahrtsverbände tabuisieren Behinderungen durch Inzest https://t.co/RShgnzsmzB

Man kann die Methode umdrehen und mit vertauschen Leithammeln auch eine Anti-AfD Ratio berechnen. Das funktioniert nicht so gut. Die AfD-Fans bilden – schon alleine durch die Existenz von AfD-Parteistrukturen – einen relativ geschlossenen Block. Die Gegner sind - was das Retweeten angeht - diffuser.

Die Hashtag-Ratio:

User	Matches	Total-Tweets	Ratio%
mrstone0856	729	1843	39
kardasiapat	430	1491	28
DerDude72445273	49	1382	3
KeineAFD2017	39	1365	2
artepmobil	318	1139	27
ContraAfD1	32	1138	2
SchafDas	266	1096	24
torsten_luer	399	1046	38
Tinimaus1110	242	1018	23
GertWalterWolf1	261	991	26
mginpl	40	965	4

Man kann dieselbe Idee auch für die Twitter-Hashtags verwenden. Von den Retweets war ich schon gewieft genug um die Arbeit den Powerusern zu überlassen. Bei den Hashtags wäre das eine Sisyphus Arbeit, da immer wieder neue hinzukommen und alte nicht mehr verwendet werden. Diese Arbeit erledigen die Poweruser. Bei Hashtags ist die Angabe von „don‘t care“ besonders wichtig. Es dominiert z.B. auf beiden Seiten der Hashtag „#afd“. Mit diesem und anderen Hashtags kann man die Gruppen nicht unterscheiden. Es wird jedoch kein Gegner „#merkelmussweg“ verwenden und umgekehrt kein AfD-Fan „#noafd“. Man sucht nach der Menge aller Hashtags der AfD-Fans die von der Gegenseite nicht verwendet werden. Diese Methode ist nicht so trennscharf wie die Retweet-Ratio. Allerdings bekommt man auch sinnvolle Werte für Benutzer, die von der ersten Methode nicht erfasst werden. Sie funktioniert auch mit umgekehrten Rollen recht gut. Die AfD-Gegner haben keine klar definierten Leithammeln, sie haben jedoch ebenfalls eine relativ geschlossene Agenda. Beide Methoden ergeben zusammen schon eine relativ präzise Gruppeneinteilung. Nur ein kleines Dorf, die AfD_Frankenthal, leistete noch Widerstand. Es greift auch die Hashtag-Methode voll daneben, weil der dafür Verantwortliche sich auch nicht an Hashtag-Regeln hält. Sie kommen in seinen Tweets nicht vor.

Die Schindel-Ratio:

Der Shingling (Dt. Schindel) Algorithmus ist eine 1993 erfundene Methode zur allgemeinen

Klassifizierung von Dokumenten. Man möchte z.B. wissen, ob ein Artikel in die Kategorie Sport, Wetter, Politik … fällt. Man gibt Muster-Berichte vor. Ein neuer Artikel wird entsprechend der größten Übereinstimmung klassifiziert.
Die erste Schindel besteht aus den Zeichen 1..11 des Textes. Die zweite Schindel von 2..12, die dritte von 3..13. Die Textfragmente überlappen sich wie die Schindeln eines Daches. Ein üblicher Wert ist eine Schindellänge von 9. Für diese Analyse hat der Wert 11 etwas besser funktioniert. Man kann z.B. auch vier aufeinanderfolgende Worte als Schindeln definieren. Man geht genauso wie bei den Hashtags vor. Man bildet die Schindeln der AfD-Power-User und zieht von diesen die gleichlautenden Schindeln der AfD-Gegner ab. Danach durchsucht man die Tweets aller User nach übereinstimmenden Schindeln. In einem Tweet können mehrere passende Schindeln vorkommen. Diese werden extra gezählt. Es kann die Ratio wesentlich größer als 100% werden bzw. es haben auch Gegner relativ hohe Schindel-Ratios. Die Frage ist nur, ob sich die Werte signifikant unterscheiden.

AfD_Frankenthal	893	72	1240

Diese Methode pickt auch die Frankenthaler in das richtige Töpfchen. Mit einem Schindel Wert von 1240 gehören sie eindeutig ins AfD Lager. Es gibt jedoch einen schwierigen User namens AfD_Hesse. Nach den Tweets zu schließen, bezeichnet „Hesse“ keine Ortsgruppe. Es ist möglicher Weise eine Referenz an den Dichter Hermann Hesse. AfD_Hesse hat bei allen Methoden nur eine mittelmäßig ausgeprägte AfD-Ratio. Er gehört aber – auf Grund der Ratio – auch nicht ins Lager der Gegner. Er ist bekennender AfDler, engagiert sich – auf Twitter – allerdings auch stark für den Tierschutz und retweeted regelmäßig PETA. Auf einen Tweet, wie das denn zusammen passt, antwortet er mit:

lang:de	04/28 13:49:45	AfD_Hesse	@Andrino11elf @LarsSteinke @neomagazin Duerfen AfD-Waehler keine Tierfreunde sein?

Insofern war die nicht so eindeutige Klassifizierung nicht ganz daneben.
Es ist kaum möglich, die Klassifizierung von Abertausenden Usern per Hand zu überprüfen. Meine Stichproben ergaben eine recht gute Quote von richtigen Einschätzungen des Bots. Dem Mann (es könnte auch eine Frau sein) aus Frankenthal bin ich dankbar, dass er mich nach dem Motto – es wäre doch gelacht, wenn ich ihn nicht richtig hin bekomme – zu zusätzlichen Verfahren angeregt hat.

Potpourri:
Vanellus unterstützt noch eine Reihe von anderen Analyse-Methode. Man kann z.B. eine Liste jener User erstellen, die am ausdauerndsten twittern. Damit erkennt man Bots. Man kann sich die Peak-Tweet-Rate innerhalb eines Zeitraumes ansehen. Mit den oben bereits beschriebenen Methoden kann man u.A. untersuchen, wie lange der Tweet eines Promi ein Thema ist. In der Regel sind es ein paar Stunden. Man kann sich anschauen, welche Tweets bei den Usern ankommen und welche nicht. Man kann bei Powerusern ein Tagesprofil erstellen. Auf Grund des Histogramms erkennt man auf einen Blick, dass Alice Weidel am 1. Mai entgegen ihren sonstigen Gewohnheiten schon ganz zeitig in der Früh ihr Twitter-Büchserl geschultert hat. Sie dürfte gegen Gewerkschafter eine ähnliche Abneigung haben wie der Da Wildschütz vor dem Jaga. Man kann sich auch anschauen, welche Hashtags gerade in sind. Wobei man nach Themen bzw. Schlagwörtern selektieren kann.

Ich habe für diese Analyse die Tweets auf dem Rechner der Lektorin herunter geladen und diese auf meinem mit mehr RAM ausgestatteten Arbeits-PC analysiert (ansonsten ist der neuere PC der Lektorin aber besser ausgestattet). Die Analyse kann aber auch in Echtzeit durchführt werden. Es ist für die Analyse-Routinen egal, ob die Daten von der Platte geladen werden oder vom Twitter-Server kommen. Sie haben kein Mascherl. Allerdings verändern die aktuellen Daten laufend das Ergebnis. Das ist für eine derartige Analyse lästig. Vanellus ist so gebaut, dass er unmittelbar auf äußeres Geschehen reagieren kann. Er tut es allerdings noch nicht.

Aussichten:
Die aktuelle Version von Vanellus ist V0.31. Der Vogel ist gerade erst dem Netz entflogen. Ich habe vor, die Software weiter zu entwickeln. Es gibt sehr viele nette Probleme, die das Herz eines alten Hackers erfreuen und ihm auch wieder das Gefühl von „forever young“ geben. Es wirkt sich jedoch auf mein Sozialverhalten negativ aus. Ich kehre in solchen Phasen der Welt der Rücken zu und fühle mich im Elfenbeinturm sehr wohl. Anmerkung der Lektorin: Der Chef für Alles ist immer so.
Es gibt bereits Interesse – aber noch keine fixen Vereinbarungen – sowohl von akademischer als auch kommerzieller Seite.

Ursprünglich sollte Vanellus auf meinem PC mit 16GB Speicher rund 50 Millionen Tweets auf einmal analysieren können. Tatsächlich ist bei 30 Millionen Schluss. Die Ursache sind die letztklassigen Java-Standardroutinen zur Behandlung von Text/Dokumenten (Für Details siehe den technischen Anhang). Im Grunde genügen auch 30 Millionen Tweets. Aber ein derartiger Mist tut meiner Hacker Seele weh. Mein Vater war ein leidenschaftlicher Energiesparer. Als er meine Schwester in Amerika besucht hat, hat er ständig unter der in seinen Augen unglaublichen Energieverschwendung der Amis gelitten. Mir geht es mit der Java-Text-Bibliothek so.

Man kann sich noch eine Reihe von anderen Analyse Methoden ausdenken. Z.B. kann man analog zur Page-Rank Methode von Google eine User-Rank einbauen. Wer wird wie oft retweetet und wie wichtig sind diejenigen, die die Botschaft weiter geben. Vanellus könnte auch das Maul aufmachen, aber ich weiß nicht recht, was er sagen soll. Ein wichtiger und großer Schritt wäre die Einbeziehung von anderen sozialen Kanälen und insbesondere von Facebook.
Generell habe ich etwas das Problem von Der Wilde auf seiner Maschin. „*I hab zwar ka Ahnung wo I hin fahr, aber dafür bin I schneller durt*“. Sollte es zur Zusammenarbeit mit einem kommerziellen Interessenten kommen, dann wäre es dessen Aufgabe dieses Problem zu lösen und ich könnte mich auf das konzentrieren was ich kann: Bitschnitzen.

„*Die Software wird schneller langsamer, als die Hardware schneller wird*“.
(N.Wirth).

Technischer Anhang: Das Elend der Java-String-Class:
Anmerkung: Nur für Programmierer geeignet und gedacht.
Vanellus hält alle Daten in einer selbst gebauten In-Memory-Datenbank. Die hereinkommenden Tweets werden in dieser Datenbank abgespeichert. Es wird von einem Tweet die Twitter-UserId, der Zeitstempel, die Tweet-Id, der User-Name, der Tweet-Text und die Abfrage auf Grund dessen er selektiert wurde, gespeichert. Es wird ein Tweet in mehrere Tabellen aufgeteilt. Es wird für den User-Namen und den Tweet-Text eine Hashzahl erzeugt. Sowohl der User-Name als auch der Tweet-Text werden in eine Hashtabelle mit ihrem CRC32-Key eingetragen. Damit werden User-Namen und auch Tweet-Texte die sich wiederholen nur 1x gespeichert. Unmittelbar wird ein Tweet in einer Tabelle mit dem Record
`<Message-Id>,<Time>,<Query-Hash>,<User-Hash>,<Text-Hash>`
gespeichert. Wobei jedes Element eine Long ist. Die Query, der User-Namen und der Tweet-Text wird bei Bedarf aus der jeweiligen Hashtabelle gelesen.
Ich habe mir dafür eigene spezialisierte Collections gebaut, da bei den Java-Collections eine long nicht als 8-Byte Zahl sondern als Long-Objekt mit mindestens 24-Byte overhead gespeichert wird. Sich eigene Collections zu bauen, widerspricht den üblichen Regeln. Meiner Meinung nach stammen diese Regeln von Leuten, die noch probiert haben die vorhandenen Ressourcen effektiv zu nutzen („down to the metal“) bzw. es ist für Leute gedacht, die das nie tun werden. Davon abgesehen findet ein alter Hacker nur das Rad gut, das er selbst erfunden hat.
Nach meinen Berechnungen sollten – wenn man der JVM 12 GByte Speicher zuweist - 50 Millionen

Tweets relativ locker in den Hauptspeicher passen. Beim Plattenimage der Datenbank geht es sich auch sehr schön aus. Zu meiner Überraschung waren es nur 30 Millionen. Das Rätsels Lösung: Java verwendet die UTF-16 Darstellung von Strings. Einst dachte man, dass man damit das Problem zusätzlicher Sprachen und Zeichensätze auf einfache Weise lösen kann. Man denkt sich als Software Entwickler nix und vertraut, dass die Hardware-Ingenieure das Problem schon lösen werden. Tatsächlich hat sich wieder einmal das Wirtsche- gegenüber dem Moorschen Gesetz durchgesetzt. UTF-16 war auch nur eine sehr kurzfristige Lösung. Man ist schnell drauf gekommen, das für asiatische Sprachen auch 2 Bytes nicht ausreichend sind. UTF-16 ist die schlechteste aller Welten. Sie ist für die lateinischen Sprachen ineffizient und löst nicht das Problem von verschieden langen Kodes. Diese einstige Fehlentscheidung treibt noch Jahrzehnte später in vielen Programmiersystemen (neben Java u.A. Qt oder Windows) ihr Unwesen. Eine wesentlich elegantere und effizientere Methode ist das von Ken Thompson und Robert Pike 1992 vorgeschlagene UTF-8 Format. Die Java Textwriter Klasse speichert Strings in diesem Format auf die Festplatte ab. Sie werden beim Einlesen jedoch wieder zu UTF-16 aufgeblasen. Nach meinen Tests benötigt UTF-8 für die deutschsprachigen Tweets nur 51% des Speicherplatzes von UTF-16.

Es gibt zwar mehrere Change Requests in Richtung UTF-8, diese wurden bisher von einer Java-Version auf die nächste verschoben. Eine derartige Umstellung ist kein triviales Problem.

Richtig bitter wird es, wenn man sich den Kode der String-Klasse anschaut. IndexOf() ist so programmiert, wie sie jeder Anfänger ohne viel Nachdenken programmieren würde. Effizientes String-Matching ist eines der besten erforschten Gebiete der Informatik. Diese Forschungen sind bei den Autoren der Klasse offensichtlich spurlos vorüber gegangen. Eine effizientere Suchmethode erzeugt auch keinerlei Kompatibilitätsprobleme. Ein besonders trauriges Kapitel ist der String-Hashkode. Auch das ist ein sehr gut erforschtes Gebiet. Die Umstellung auf eine brauchbare Hashfunktion würde jedoch so manchen Kode brechen. In Vanellus verwende ich bereits bisher die wesentlich bessere CRC32-Methode als Hashwert. Obwohl diese Zahl wie der Name schon sagt 32-Bit lang ist, gibt Java einen long-Wert zurück, weil das im Checksum-Interface so vorgesehen ist. Es rächt sich hier die fehlende Unterstützung von unsigned int durch den Sprachstandard. Ich habe es bei Long belassen. Das ist nicht das eigentliche Problem und möglicher Weise kommt man einmal drauf, dass CRC32 zu viele Kollisionen erzeugt. Dann kann man relativ problemlos auf einen 40- oder 64-Bit Hashmethode umstellen.

Fest steht, dass ich nicht mit so einem Murks weiter arbeiten will. Zu Beginn eines Projektes kann man noch derartige Fehler korrigieren. Man entkommt den Java-Strings nicht gänzlich, da z.B. die Anzeige von Tweets im GUI Strings als Input verlangt. Diese Daten sind jedoch sowieso streng von der Datenbank getrennt und man zeigt auch nicht 50 Millionen Tweets an.

Eine Alternative ist die Verwendung einer Library wie Lucene. Lucene verwendet – wie wohl jede big-data Library die etwas auf sich hält - UTF-8. Die andere Möglichkeit ist eine eigene UTF-8 String-Klasse zu schreiben. Theoretisch sollte auch ein 1-Byte Kode wie ISO-8859-15 für Deutschsprachige Tweets genügen. Man bekommt jedoch bei Emojis Probleme.

Bisher habe ich mich hauptsächlich mit Bit-Schnitzen und Numerischen Operationen beschäftigt. Strings habe ich nur zum Beschriften von Tabellen verwendet. Die Welt der Big-String-Data ist neu – und spannend – für mich. Ich hielt es nicht für möglich, dass eine derartig zentrale Klasse derartig viel Schrott enthält und habe mir über diesen Punkt – abgesehen von der Hashfunktion - anfangs keinerlei Gedanken gemacht.

Nachtrag: Ich habe nun mit dem Bau einer eigenen UTF-8 Klasse begonnen und habe bei der Konvertierung von Groß- in Kleinschreibung Fehler beim türkischen I (\u0130) gefunden. Die Tweets sind auf Deutsch, enthalten jedoch türkische Namen. Bei der Suche am Netz stellte sich heraus, der Bug liegt nicht auf meiner sondern auf Java Seite. Eine vollständige Konvertierung in allen Alphabeten ist eine ziemliche Herausforderung. Allerdings ist Türkisch keine exotische Sprache. Das sollte eine Standard-Library schon können.