Ein Big-Data-Anwendungsfall im Bereich der Schwarmintelligenz

Wir sind Weltmeister

Großereignisse werden gerne genutzt, um bestimmte Marken oder neue Produkte intensiv zu bewerben. Diese Chance wollten auch die Autoren dieses Beitrags mit der Fußballweltmeisterschaft 2014 in Brasilien wahrnehmen, um die Vorteile von Big Data für geschäftliche Innovationen an einem konkreten Einsatzbeispiel zu demonstrieren. Ein passender Zeitpunkt, denn aufgrund der Negativschlagzeilen zur NSA-Affäre gerieten die positiven Aspekte von Big-Data-Lösungen damals extrem ins Hintertreffen.

AUTOREN: CHRISTOPHER THOMSEN UND JOCHEN WILMS

Vor diesem Hintergrund entstand die Idee, Twitter für die Vorhersage von Spielergebnissen zu nutzen: Die Big-Data-Spezialisten wollten die Fußballtipps von Twitter-Nutzern als „Schwarmintelligenz“ nutzen, um bessere Tipps zu erhalten. Die Ergebnisse konnten sich sehen lassen. Das Gimmick fand aber nicht nur großen Zuspruch in der BI-Community, sondern mit ihm lassen sich auch die unterschiedlichen „V“ im Big-Data-Kontext anschaulich erläutern und technologische Umsetzungsmöglichkeiten beispielhaft darstellen.

DIE DREI V ALS KRITERIUM FÜR BIG DATA

Wenn wir von Big Data sprechen, kommen wir an den drei V: Volume, Variety und Velocity als wesentliche Kriterien nicht vorbei. Beim WM-Tippspiel kam das erste V (Volume) durch die hohe Anzahl an Twitter-Nachrichten zustande, die von den Autoren gesammelt und ausgewertet wurden, um möglichst alle Varianten der Tippabgabe zu berücksichtigen. Bis zum WM-Finale wurden knapp 250 Millionen Tweets analysiert (**Abb. 1**).

Abb. 1: Beispiel eines WM-Tipps via Twitter und des extrahierten Ergebnisses kurz vor Beginn des Finales

Das zweite V (Variety) ergab sich durch die eher unstrukturierten, 140 Zeichen umfassenden Textnachrichten. Diese lud das Projektteam als Textdatenstrom in fünf gängigen Weltsprachen ein und wertete diese teils im Datenstrom, teils im verteilten Batch-Verarbeitungsframework MapReduce unter Nutzung von Text-Mining-Werkzeugen aus. Die Ergebnisse fügten sie anschließend einem relationalem Datenmodell hinzu, das sich aus bestimmten Eigenschaften und dem manuell hinzugefügten tatsächlichen Spielergebnis zusammensetzte.

Velocity als drittes Big-Data-V war insbesondere der Tatsache geschuldet, dass die meisten Twitterer ihre Beiträge wenige Minuten vor dem Anpfiff posteten. Es hieß also, die Tweets noch vor Spielbeginn in Echtzeit zu verarbeiten.

DIE TECHNOLOGISCHE UMSETZUNG

Das breite Technologiespektrum, dem sich Big Data zuordnen lässt, bot verschiedene Möglichkeiten, die Idee

für das Tippspiel in die Praxis umzusetzen.

DIE LAMBDA-ARCHITEKTUR

Die Herausforderung an die Architektur war zum einen, dass die Kurznachrichten der Social-Media-Plattform zu Spitzenlasten, also kurz vor den Spielen und während der Spiele, abgefangen werden mussten, und dass neue Kurznachrichten binnen Sekunden in ein aktualisiertes Tippergebnis einfließen sollten, um auch noch kurz vor Spielbeginn abgegebene Tipps zu berücksichtigen. Zum anderen galt es, zusammenhängende Kurznachrichten, wie Kommentare, in denen ein Tippergebnis zu einer in der dazugehörigen Frage genannten Begegnung genannt wird, zueinander in Verbindung zu setzen und auszuwerten.

Eine Lambda-Architektur adressiert eben diese Herausforderungen, indem sie skalierbare Technologien für Echtzeitdatenstromverarbeitung, Batch-Prozesse und Ad-hoc-Abfragen kombiniert. **Abbildung 2** zeigt wie dies im Allgemeinen aussieht. **Abbildung 3** stellt die Umsetzung im geschilderten Einsatz dar, also bei der Auswertung von Twitter-Kurznachrichten.

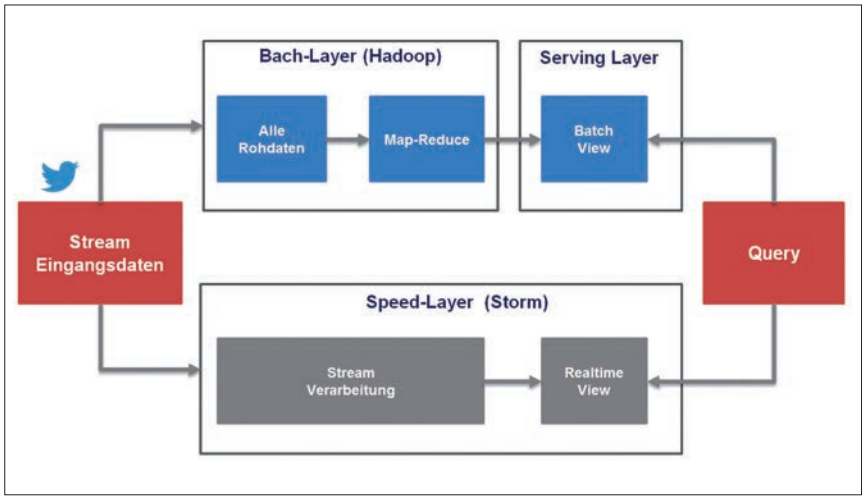

Abb. 2: Lambda-Referenzarchitektur

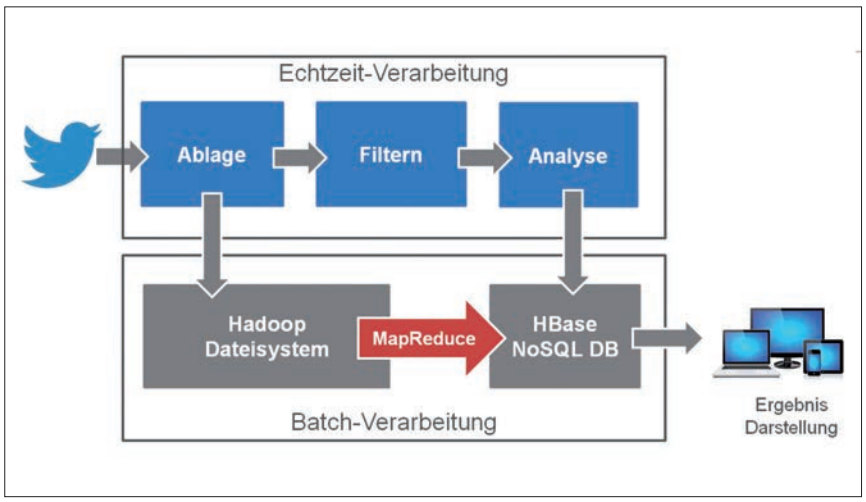

Abb. 3: Umsetzung der Lambda-Architektur für die Auswertung von Tippergebnissen auf Twitter

Die Lambda-Referenzarchitektur ermöglicht, wie gesagt, einen Live Stream aus dem sozialen Netz direkt in einem „Echtzeitstrom“ auszuwerten, parallel dazu komplexe Auswertungen auf den zuvor gesammelten Daten Batch-orientiert auszuführen und diese in einem Resultat wieder zusammenzuführen.

Im Beispielprojekt registrierten die Entwickler über ein von Twitter angebotenes Streaming-API eine Suchanfrage mit über 100 Suchtermen sowie einige Sammelaccounts zur Fußball-Weltmeisterschaft. Die Anbindung des Twitter-API übernahm eine dedizierte Komponente, die auf einem eigenen System läuft. Diese Komponente übernahm in diesem Szenario die Rolle einer Datenquelle, da sie die Tweets, die über das Twitter-Streaming-API geliefert wurden, direkt über eine TCP-Socket-Verbindung an das Streaming-System weiterleitet. Über diese Registrierung sendete Twitter in Echtzeit neu erstellte Tweets an einen Streamingserver. Die Größenordnung betrug während der WM zwischen 500 und 4 000 Tweets pro Sekunde.

Die Streamingapplikation selbst übernahm die Analysen, die in Echtzeit auf jedem einzelnen Tweet durchgeführt wurden. Diese isolierte Verarbeitung ermöglichte eine optimale Skalierbarkeit. Zugleich wurden eingehende Tweets vom Streamingsystem zur Archivierung in das verteilte Dateisystem eines Hadoop-Clusters (HDFS) geschrieben.

Im weiteren Verlauf der Echtzeitverarbeitung ließ das Team mehrfach Daten auf dieses Dateisystem schreiben. Dieser Vorgang lief asynchron, um die eigentliche Verarbeitung nicht auszubremsen. Diese Daten werden für Batch-Läufe benötigt, die nicht in Echtzeit stattfinden. Als Konsument trat der Application Server auf, der der Webseite mit den Tippergebnissen eine REST-Schnittstelle anbot. Der Application Server sendete bei

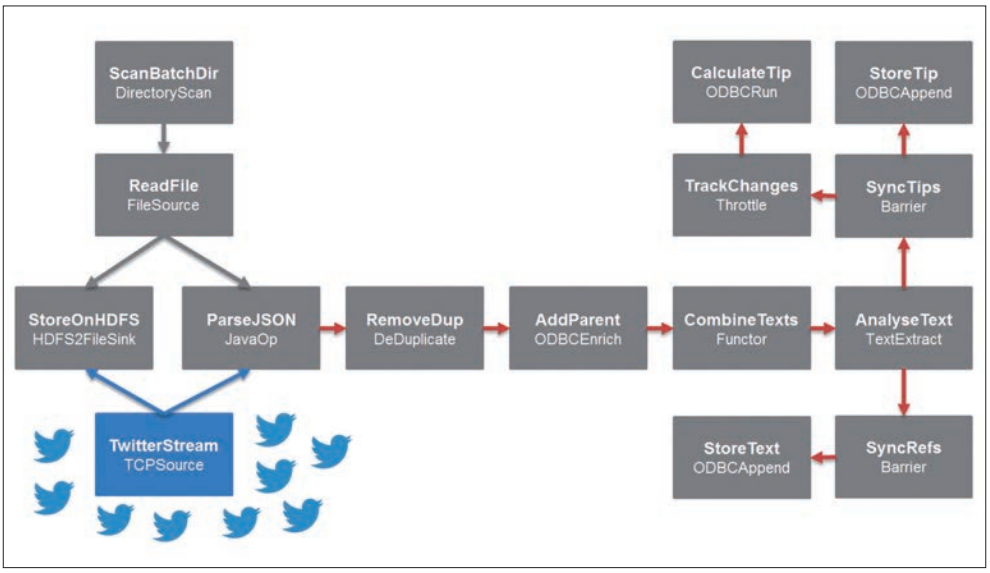

Abb. 4: Graph der Echtzeitverarbeitung zur Analyse, Interpretation und Persistierung der Tweets

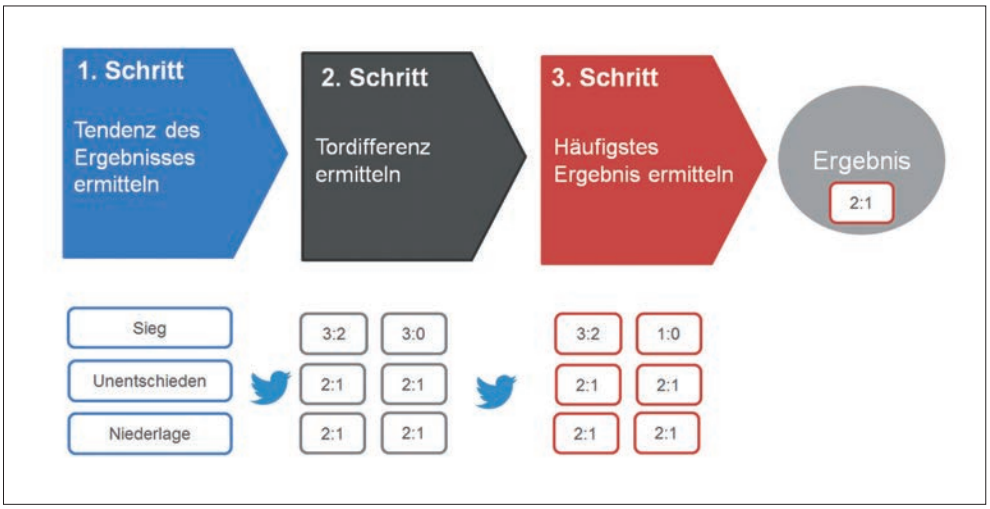

Abb. 5: Ermittlung des optimalen Tippergebnisses

jeder Aktualisierung der Seite – via AJAX alle fünf Sekunden – Querys an die Datenbank, die ihm vorbereitete Result Views zurücklieferten.

ANALYSEVERFAHREN

Die Analyse des eigentlichen Tweet-Texts separierte die Tweets in irrelevante, direkte Tipps (Mannschaftsnennungen plus Ergebnistipp) und in Referenzen (nur Mannschaftsnennung), die für die Anreicherung von Antworten genutzt wurden. Der Streamingprozess sendete in regelmäßigen Abständen Informationen an das Batch-System, die meldeten, zu welchen Spielen neue valide und interpretierbare Tipps eingingen. Zu diesen Spielen startete das Programm daraufhin die Neuberechnung des Tippwerts über eine einfache Auszählung aller Tipps, die per Tweet zu diesem Spiel abgegeben wurden, skaliert über einen MapReduce-Job. Den neu berechneten Tipp speicherte das System in der Ergebnistabelle der View-Datenbank.

Ebenfalls transformierte das Tool die Tweets, die über das Twitter-Streaming-API eingingen als einzelne Data Tuple in einem Streamingsystem transformiert, filterte sie, analysierte und persistierte sie mit einer Engine für Textextraktion, Regular Expression und Spracherkennung. Das Streamingsystem arbeitet in Echtzeit, was in diesem Zusammenhang bedeutete, dass die Daten während aller Operationen im Arbeitsspeicher gehalten und niemals auf die Festplatte geschrieben wurden. Dies erhöhte die Verarbeitungsgeschwindigkeit um mehr als das Hundertfache (**Abb. 4**).

Die relevanten Informationen der im JSON-Format serialisierten Tweets konvertierte die Software zur Weiterverarbeitung in Data Tuple und schrieb sie in ihrer Rohform auf das HDFS. So bereinigte sie die Daten um Duplikate oder ergänzte gegebenenfalls Informationen eines bereits persistierten Tweets, wenn es Antworten auf ihn gab. Nach Ermittlung der Einzeltipps durch die im nächsten Abschnitt beschriebene Textextraktion leitete der Bot in regelmäßigen Zyklen aus den Einzeltipps die optimalen Tipps ab. **Abbildung 5** veranschaulicht dieses Verfahren.

Im ersten Schritt ermittelte das Programm die mehrheitliche Tendenz, im zweiten Schritt bestimmte es aus der daraus hervorgehenden Mehrheitsgruppe die vorwiegend getippte Tordifferenz, und zuletzt machte es aus allen Tipps für das favorisierte Team mit der entsprechenden Tordifferenz das mehrheitliche Tippergeb-

nis ausfindig. Dieses Vorgehen führte zu einem besseren Resultat als eine einfache Bestimmung des Mehrheitstipps. Warum dies so war, lässt sich an einem einfachen Beispiel darstellen: Wir haben im Spiel A gegen B zehn Tipps auf ein 1:0, vier Tipps für ein 0:1, acht Tipps für ein 0:2 und sechs Tipps für ein 1:2. Demnach ist die Mehrheit für einen Sieg von Mannschaft B, eine einfache Mehrheitstippbestimmung käme jedoch auf ein 1:0 für Mannschaft A. Nach dem Algorithmus unseres Beispiels wäre der optimale Tipp hingegen das 1:2 für B, obwohl der Einzeltipp 1:2 nur sechs der 28 Stimmen erhielt und nur auf Platz 3 von 4 in absoluter Tippreihenfolge lag, da die Mehrheit der Tipps einen Sieg für B und mehrheitlich eine Tordifferenz von einem Tor voraussagte.

TEXT-MINING

Big-Data-Fachleute nutzen Werkzeuge zur Tokenization, Annotation, Regex-Interpretation und Syntaxinterpretation für die Zerlegung eines Texts in Einzelbausteine und deren Anreicherung mit Metainformationen. Für diese annotierten Bausteine können sie dann Abfragen und Bedingungen für die Interpretation von Aussagen und Kennzahlen formulieren.

Listing 1: Laden der Wörterbücher und Mappings für die Annotation der WM-Tipps

```
create external table Teams (id Text, notation Text, lang Text) allow_empty false;
create dictionary TeamsDict from table Teams with entries from notation;
create view TeamAnnotationsRaw as extract dictionary 'TeamsDict' on D.text as
                                                                match from Document D;
create view TeamAnnotations as
 select T.id as id, T.notation as notation , T.lang as lang , R.match as match
 from TeamAnnotationsRaw R , Teams T
 where Equals (ToLowerCase(GetText(R.match)), GetText (T.notation));
export view TeamAnnotations;
```

Um beim ersten Schritt anzufangen und aus dem unstrukturierten Rohtext einer Textnachricht Informationen herauszuziehen, braucht es ein Verfahren, das eine Struktur in den Text bringt. Hierzu identifiziert das System möglichst viele Wörter des Texts und ordnet sie einer Domäne zu. Mit der Zuordnung zu einer Domäne wird ein Wort inhaltlich klassifiziert. Beispiele für Domänen wären Städtenamen oder Produktmarken. Die Identifikation kann auf verschiedene Weise erfolgen:

- Durch die Zusammenstellung von Wörterbüchern, die alle Wörter einer Domäne enthalten und diese eventuell mit zusätzlichen Informationen verknüpfen. Dies ist der typische Ansatz für Eigennamen wie Länder- oder Unternehmensbezeichnungen sowie für die Zuordnung von Eigenschaften zu ganzen Wortgruppen einer Sprache.
- Anhand von Regular Expressions (Regex), die Wörter anhand von bestimmten Pattern erkennen und einer Domäne zuordnen. Klassische Beispiele hierfür sind Telefonnummern, ISIN-Nummern für Wertpapiere und IBAN-Nummern für Bankkonten.
- Mit der Aufstellung von Grammatikregeln, die Wörter einer Sprache ihrer Wortart sowie ihrer Deklination bzw. Konjugation zuordnen.

Die durch diese Regeln markierten Textpassagen, inklusive ihrer durch die Domäne oder den Wörterbucheintrag beigefügten Eigenschaften und ihrer Position im Text, nennen sich Annotationen. Bei diesen Regularien ist es durchaus möglich, dass Wörterbucheinträge und Regex-Regeln nicht nur Einzelwörter, sondern auch Wortkombinationen suchen, genauso wie es üblich ist, dass einzelne Wörter zu mehreren Annotationen gehören.

Als Ergebnis dieses Vorgangs erhalten wir eine Tabelle pro Domäne mit allen dazu passenden Annotati-

Listing 2: Extraktion eines quantifizierbaren Tipps aus einer Kurznachricht

```
create view Results as
 extract regex /(?<![\w-])(\d) ?[:–] ?(\d)(?![\w-])/ on D.text
 return group 1 as result1 and group 2 as result2
 from Document D;
create view ResultCount as select Count(*) as num from Results;
create view Tips as
 (select E.id as encounter, GetText(R.result1) as result1, GetText (R.result2) as
                                                                        result2,
  GetText(D.text) as original_text, E.date as match_date
 from TeamAnnotations TA1, TeamAnnotations TA2, Encounters E, Results R,
                                        TeamCount TC, ResultCount RC , Document D
 where GreaterThan (3, TC.num) and GreaterThan (2 , RC.num)
  and Equals (GetText(E.team1), GetText (TA1.id))
  and Equals (GetText(E.team2), GetText(TA2.id))
  and FollowsTok (TA1.match, TA2.match, 0, 4))
union all
(select E.id as encounter, GetText(R.result2) as result1, GetText (R.result1) as
                                                                        result2,
 GetText(D.text) as original_text , E.date as match_date
from TeamAnnotations TA1, TeamAnnotations TA2, Encounters E, Results R,
                                        TeamCount TC, ResultCount RC, Document D
where GreaterThan(3 ,TC.num) and GreaterThan(2 ,RC.num)
 and Equals(GetText(E.team1), GetText(TA2.id))
 and Equals(GetText(E.team2), GetText(TA1.id))
 and FollowsTok(TA1.match, TA2.match, 0, 4));
output view Tips;
```

onen. Diese Tabelle bildet die Basis für jede weitere Analyse der Textinhalte, und spätestens ab hier gibt es zahlreiche Technologien und Algorithmen, die dabei helfen, aus den gefundenen Annotationen einen semantischen Sinn zu erschließen. Programme und Skripte, die Wörterbücher, Grammatikregeln und Regex nutzen, um Annotationen zu finden und aus diesen quantifizierbare Werte oder Wertetabellen abzuleiten, nennt man Textextraktoren.

Die Implementierung eines Textextraktors für die Tippermittlung aus Twitter-Kurznachrichten hat das Expertenteam beim Fußballtippspiel in AQL vorgenommen, einer Abfragesprache, die von Aufbau und Syntax her an SQL erinnert. Die Funktionsweise kann man sich wie bei einfachem SQL vorstellen, das jedoch einige neue Datentypen enthält, zum Beispiel Matches und Spans, die Pointer zu ganzen Textabschnitten und Informationen über Position und Länge darstellen. Für diese zusätzlichen Datentypen bietet AQL eine Reihe von Funktionen an, um beispielsweise zu testen, wie weit zwei Matches voneinander entfernt liegen, oder ob sie sich überschneiden oder gar identisch sind.

Um die unstrukturierten Daten erstmals in Tabellenform zu bringen, muss der Entwickler diese parsen und in eine quantifizierbare Datenstruktur überführen. Listing 1 zeigt einen Quelltextauszug, in dem neue Tabellen und Wörterbücher aus vorhandenen CSV-Dateien erstellt werden, die zum Beispiel Mappings von Team- sowie Ländernamen enthalten. In Zeile 3 des Codebeispiels werden diese dann verwendet, um eine Extraktion aus dem Zieldokument durchzuführen. Ab diesem Punkt liegen die gefundenen Matches in tabellarischer Form vor und können weiterverarbeitet werden.

Listing 2 zeigt, wie die Matches verwendet werden müssen, um aus ihnen Mannschaftsbeteiligungen über Wörterbücher sowie Tippergebnisse über Regex zu ermitteln und diese dann zu zählen oder um syntaktische Korrektheit zu versichern und die Tipps korrekt zuzuordnen.

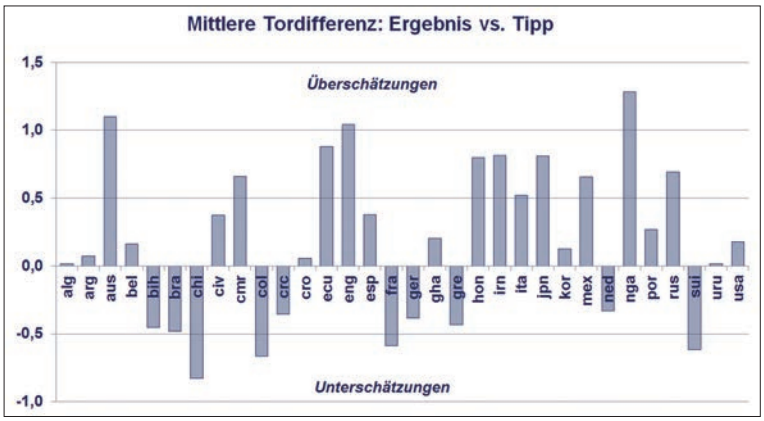

Abb. 6: Abweichung erzielte versus getippte Tore der Mannschaften

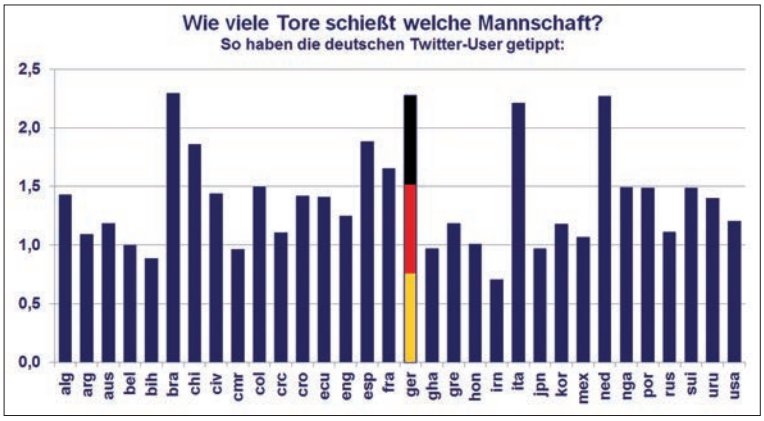

Abb. 7: Durchschnittlich getippte Tore pro Nationalmannschaft von deutschen Twitter-Accounts

FERNAB DES CODES – FUSSBALLERISCHE ERKENNTNISSE

Ein Fußballfan, der sich in der Big-Data-Szene tummelt, ist natürlich nicht nur an Codefragmenten und eingesetzten Werkzeugen interessiert. Er fiebert bei jedem Spiel mit, und diesem Umstand verdanken wir verschiedene Erkenntnisse aus den Tippvorhersagen, die bei Wetteinsätzen für kommende Europa- und Weltmeisterschaften interessant sein könnten.

Rein statistisch zerfällt das wahrscheinliche Ergebnis eines Fußballspiels zu gleich großen Teilen in drei Möglichkeiten: Sieg, Unentschieden, Niederlage. Kommerzielle Wettanbieter haben Prognosemodelle, deren Qualität deutlich über 70 Prozent liegt, was für ein torarmes und damit zufallanfälliges Spiel wie Fußball ein gutes Ergebnis ist. Das Modell aus diesem Beitrag bietet eine Qualität von über 60 Prozent. Hätte dies gereicht, um im Nachbarschafts- oder Kollegentippspiel zu gewinnen? Nein. Denn hier übertrifft der Fußballsachverstand gepaart mit Glück einzelner Kenner meist den eher ausmittelnden und damit durchschnittsorientierten Sachverstand der Schwarmintelligenz.

Nichtsdestotrotz ist die Schwarmintelligenz gut geeignet, um allgemeine Erwartungshaltungen zu überprüfen. So hofften die Twitter-Tipper in Summe auf mehr Tore als wirklich gefallen sind (**Abb. 6**). Und das obwohl die WM in Brasilien mit 2,7 Toren pro Spiel keine besonders schlechte war. Vor allem die Mannschaften aus Australien, England und Nigeria blieben hinter den Erwartungen

der Tipper zurück. Hier tippten die Twitter-Anwender jeweils im Schnitt ein Tor pro Spiel besser als tatsächlich von den Mannschaften erzielt. Positiv überraschten Chile, Kolumbien und die Schweiz, die jeweils ein halbes Tor im Schnitt mehr erzielten als vorausgesagt.

Auch nationale Auswertungen sind anhand der teilweise freigegebenen Ländermerkmale der Twitter-Accounts möglich. So erhofften sich die Deutschen von den Offensivabteilungen der Nationalmannschaften aus Brasilien, Italien, den Niederlanden und des eigenes Teams mehr als zwei Tore pro Spiel. Dem Endspielgegner Argentinien traute man jedoch nur durchschnittlich ein Tor zu, ähnlich wie Griechenland, Belgien oder Russland.

Das außergewöhnliche 7:1 im Halbfinale der Deutschen über Gastgeber Brasilien fand sich in den Kurznachrichten nur einmal. Und dies nicht als wirklicher Tipp. Ein englisches Wettbüro teilte seine Wettquote für viele mögliche Ergebnisse mit, unter anderem auch ein 7:1. An diesem Tweet, der eigentlich kein Tipp ist und deshalb anhand des Geschäftsverständnisses nicht berücksichtigt werden sollte, zeigt sich ein typisches Big-Data-Merkmal: Aufgrund der großen Anzahl erkannter Tipps pro Spiel fallen solche Einzelfehler nicht ins Gewicht. Es braucht also auch keine Fehlerbehandlung. Ein Grundrauschen ist für den Data Scientist bei mehr als 2 000 ausgewerteten, validen Tweets pro Spiel akzeptabel.

BIG DATA ZUM ANFASSEN

Reicht die Qualität der Twitter-Schwarmintelligenz in Sachen Fußball aus, um sich ganz auf das Wettgeschäft zu verlegen? Die Antwort fällt eindeutig negativ aus. Die Qualität der Schwarmintelligenz und die geringe Anzahl tipprelevanter Tweets während des normalen Ligabetriebs reichen nicht aus, um ein solches Geschäftsmodell dauerhaft zu betreiben.

Jedoch lassen sich die eingesetzten Big-Data-Architekturmuster und -Technologien auf viele Anwendungsfälle übertragen. So wäre die vorausschauende Wartung ein typisches Feld, das im Big-Data-Kontext häufig genannt wird. Ebenso die Prognose von Kursentwicklungen anhand von Nachrichten und Posts oder die Vorhersage von Krankheitsverläufen und Erfolgswahrscheinlichkeiten von Behandlungen im Kontext der aufkommenden Selbstvermessung. Auch bei Analysen von Social-Media-Beiträgen zur Marktforschung oder zur Markenbewertung kommt es auf die Masse von vielschichtigen Daten in unterschiedlichen Geschwindigkeiten und auf ihre Interpretation an.

Der eingesetzte und für Big Data typische Architektur- und Technologiemix fordert multidisziplinäre Entwickler, die man wahrscheinlich nicht einzeln sondern in so genannten Data-Science-Teams finden wird. Diese sollten imstande sein, komplexere Big-Data-Anwendungsfälle umzusetzen, die über unser WM-Tipp-Beispiel erheblich hinausgehen, und sie sollten diese über mehrere Iterationen beständig optimieren.

Um zu der eingangs erwähnten Motivation der Autoren dieses Artikels zurückzukommen: Viele Rückmeldungen auf das Anwendungsbeispiel bestätigten am Ende, dass es ihnen mit der Verknüpfung des Themas Big Data und des Populärthemas Fußball durchaus gelungen ist, die Fachöffentlichkeit von der „schönen Seite“ von Big Data zu überzeugen und damit mehr Akzeptanz für dieses spannende Thema zu schaffen.

Christopher Thomsen

ist Consultant bei der OPITZ CONSULTING Deutschland GmbH. Dort arbeitet er im SOA- und JEE-Bereich und leitet das Competence Center Big Data. Sein besonderes Interesse liegt in der Verknüpfung von Technologien aus IT-Trendthemen wie Big Data, Cloud Computing und Mobile Applications.

christopher.thomsen@opitz-consulting.com

Jochen Wilms

arbeitet im Bereich Business Development & Innovation für das Projekthaus OPITZ CONSULTING. Er blickt auf eine langjährige Projekterfahrung im Bereich Business Intelligence und Data Warehouse zurück. Mit großer Begeisterung widmet er sich allen Trends, die anforderungsgenaue Business-Analytics-Lösungen ermöglichen. Aktuell gilt hierbei seine Aufmerksamkeit den Themen Mobile BI, Big Data und Agile BI.

jochen.wilms@opitz-consulting.com