Live Twitter Tempas

Labor Web-Technologien

Jonas Bock
Gottfried Wilhelm Leibniz Universität Hannover
Forschungszentrum L3S Research Center
Hannover, Deutschland

Christian Dirkes
Gottfried Wilhelm Leibniz Universität Hannover
Forschungszentrum L3S Research Center
Hannover, Deutschland

***Abstract*—Dieses Dokument beschäftigt sich mit der Problematik Twitter als Suchmaschine umzufunktionieren um Links zu interessanten Themen zu erhalten. Dazu wird die Twitter eigene API untersucht, sowie eine Alternative vorgestellt.**

***Index Terms*—Twitter, live, Link Suche.**

I. Einleitung

Twitter als „Suchmaschine“ zu gewinnen hat vielerlei Vorteile. Informationen verbreiten sich über Twitter in rasender Geschwindigkeit und je mehr darüber geredet wird umso relevanter *kann* diese Information für die eigene Recherche sein. Wenn zu einem Thema viel *getweetet* wird, *könnte* das bedeuten, dass dieses Thema wichtig ist.
Zu Beginn einer Recherche oder Ausarbeitung kann hierauf zurückgegriffen werden, um sich Denkanstöße zu holen. Twitter bietet von Haus aus eine API an, mit der nach Tweets gesucht werden kann. Diese Ergebnisse lassen sich dann automatisiert auslesen und verarbeiten.

II. Problemstellung

Ein Webarchiv beinhaltet einen Versionsverlauf von einem Großteil aller Internetseiten. So können dort alte Informationen und alte Aufbauten der Seiten angesehen werden. Der Zugriff auf diesen Versionsverlauf erfolgt über eine WayBackMachine [4], eine Suche jedoch findet derzeit aber nur über die Eingabe der URL statt. Da es über Suchmaschinen wie zum Beispiel Google nicht möglich ist auf das Webarchiv zuzugreifen, soll nun eine Suche ermöglicht werden. Hier muss es möglich sein nach Stichwörtern zu suchen und dabei optional auch einen Zeitraum anzugeben, um dadurch relevante Internetquellen zu finden.

III. Grundlagen

In diesem Kapitel wird auf die Grundlagen eingegangen.

A. Twitter

Twitter ist ein Social Network, das bedeutet dass viele Menschen auf dieser Plattform miteinander via so genannter Tweets kommunizieren. Jeder Benutzer muss dort einen eigenen Account haben um Tweets verfassen zu können. Die Tweets lassen sich am einfachsten mit einer SMS vergleichen. Ähnlich der SMS dürfen Tweets maximal 140 Zeichen lang sein, aber im Gegensatz zu der SMS dürfen auch Bilder oder Links zu verschiedenen Seiten *getweetet* werden und mit sogenannten Hashtags können den Tweets Schlagwörter zugeteilt werden. Twitter stellt eine API zur Verfügung, mit dessen Hilfe alte Tweets ausgelesen oder auch beispielsweise neue Tweets erstellt werden können. Mehr zur Twitter API im Kapitel III.A.

B. PHP

PHP oder auch „PHP: Hypertext Preprocessor“ wurde früher auch „Personal Home Page Tools“ genannt. Es handelt sich hierbei um eine an C sowie Pearl angelehnte Scriptsprache, welche größtenteils dafür genutzt wird, dynamische Web- Inhalte oder Anwendungen zu generieren. Seit PHP 5 ist es auch möglich objektorienteiere Inhalte zu entwickeln.

C. PHP - cURL

cURL ist seit PHP 4.0.2 ein Bestandteil von PHP. Hierbei handelt es sich um eine Bibliothek, welche von Daniel Stenberg entwickelt wurde. Mit ihrer Hilfe ist es möglich mittels verschiedener Protokolle, sowie SSL-Zertifikaten, Verbindungen zu Servern aufzubauen.

Mit der Hilfe durch cURL wird die Möglichkeit geboten an den Quellcode verschiedener Seiten zu kommen und via PHP zu parsen.

D. JavaScript

JavaScript wurde 1995 von Brendan Eich entwickelt und ist ebenfalls eine Methode um dynamische Homepage Inhalte zu generieren, ändern und sogar nachzuladen.

Angelehnt ist JavaScript, ähnlich wie PHP, auch an Pearl und C, jedoch haben noch weitere Sprachen wie Scheme, Python und Java Einfluss auf die Entwicklung gehabt.

Ein weiterer Unterschied ist, dass JavaScript nur Zugriff auf Browserinhalte hat und somit nicht auf das Dateisystem zugreifen kann.

E. jQuery

jQuery [3] ist die meist verwendete JavaScript Bibliothek, welche frei zur Verfügung steht. Nahezu 50% aller Webseiten nutzt diese Bibliothek bereits [1]. Diese existiert in zwei verschiedenen Versionen. Version 1.x unterstützt noch ältere Browser, wie beispielsweise den Internet Explorer bis Version 8, wogegen Version 2.x für die neuere Generation Browser, aufgrund verbesserter JavaScript-Unterstützung der aktuelleren Webbrowsern, gedacht ist.

IV. Lösungsansatz

Für diese Suche bietet sich Twitter als Zwischenschritt an. Tweets bestehen aus einem kurzen Text, in dem Stichwörter nach denen wir suchen enthalten sein können und eine Zeitangabe der Veröffentlichung. Des weiteren können Tweets einen Link enthalten, welcher für uns interessant ist. Diese Informationen können genutzt werden um einen zu den Stichwörtern zugehörigen Link, zum Zeitpunkt des Tweets, passenden Eintrag aus dem Webarchiv anzuzeigen.

V. Such Methoden

An dieser Stelle wird auf verschiedene Methoden eingegangen, welche benutzt werden können um an die gewünschten Twitter Informationen zu gelangen.

A. Twitter API

Twitter bietet eine eigene Schnittstelle, auch API (Application-Programming-Interface) genannt [2], über die man auf verschiedensten Arten Daten abfragen, verarbeiten oder übertragen kann. Um auf die API zugreifen zu können muss zunächst ein Benutzer (User) auf Twitter registriert werden. Anschließend kann mittels einer OAuth Autorisierung eine Verbindung zu Twitter hergestellt werden.

OAuth ist ein offenes Protokoll, welches eine sichere, standardisierte Autorisierung für Web-, Desktop oder Mobile-Applikationen erlaubt.

Nachdem eine sichere Verbindung hergestellt wurde, ist es möglich auf die eigenen Tweets zuzugreifen. Dieses Verfahren wird seitens Twitter jedoch stark eingeschränkt. So sind kleinere Zugriffsmengen kostenlos, wenn jedoch viele Daten benötigt werden wird es kostenpflichtig.

1) Nachteile der Twitter API

Zunächst ist man gezwungen einen Twitter Account zu erstellen und sich via OAuth zu authentifizieren. Anschließend hat man nur begrenzt Suchanfragen, als registrierter User 180 innerhalb von 15 Minuten, zur Verfügung.

Diese limitierten Suchanfragen sind noch jeweils auf 100 Ergebnisse begrenzt.

Der größte Nachteil ist jedoch, das Twitter nur die Tweets der letzten 7 Tage zur Verfügung stellt. Diese Limitierungen sind Stand 03.2016.

2) Vorteile der Twitter API

Der einzige Vorteil hierbei liegt darin, dass die von Twitter zurückgelieferten Daten im JSON Format vorliegen. Dadurch lassen sich diese leicht automatisiert verarbeiten.

Aufgrund der vielen Nachteile ist die Twitter API für unsere Zwecke leider völlig unzureichend. Wir wollen nämlich alle Daten zu einem bestimmten Thema innerhalb eines gewissen Zeitraumes. Deshalb suchten wir nach einer Alternative.

B. Twitter Suche

Diese war schnell gefunden. Twitter bietet von Haus aus eine sehr Umfangreiche Suche an. Im Gegensatz zur Twitter API, sind wir hier nicht auf 100 Tweets, oder den letzten 7 Tagen, limitiert sondern es stehen uns alle Tweets seit 2006 zur Verfügung. Des Weiteren bietet die Twitter Suche die Möglichkeit Operatoren, wie beispielsweise :) oder :(Smileys für positive oder negative Tweets, zu verwenden, wodurch noch exaktere Suchergebnisse erzielt werden können.

Mit diesen Daten kommen wir dem Ziel, Twitter als eine Art Suchmaschine zu benutzen, einen großen Schritt näher. Der einzige Nachteil hierbei ist, dass die Daten nicht mehr im JSON Format vorliegen.

C. Zwischenfazit zu den Methoden

Da die Twitter API uns starke Limitierungen auflegt, die Twitter Suche jedoch nicht haben wir uns an dieser Stelle dazu entschlossen mit der Suche zu Arbeiten. Der einzige Nachteil an der Suche wurde nur der Vollständigkeitshalber erwähnt, wird jedoch keinen Einfluss auf die weitere Entwicklung des Live Twitter Tempas haben.

VI. Entwicklung & Probleme

Zunächst wurde mittels Reverse Engineering die Strukturen der Twitter Suchseite analysiert, wodurch wir herausfanden, dass die Suchergebnisse in einer Tabellenstruktur ausgeliefert werden. Dies wurde zum ersten Ansatz um die Daten auszulesen und zu verarbeiten.

Mit diesem Wissen bauten wir eine Suchmaske nach, um unsere eigenen Suchwörter mittels PHPCurl an die Twitter Suche zu senden und die Ergebnisse ebenfalls per PHP String-Funktionen auszuwerten.

Zur Auswertung benutzen wir Reguläre Ausrücke um die für uns relevanten Code Blöcke zu erhalten. Dieser wäre der „tweet-container“, welcher alle wichtigen Informationen zum Tweet erhält. Aus diesem Container extrahieren wird anschließend alle Hashtags und Links, welche wir paarweise speichern und deren Häufigkeit für ein internes Ranking mitzählen. Das Ranking ist einfach gehalten und Listet lediglich die Ergebnisse in absteigender Reihenfolge.

Im Folgenden verschiedene Probleme und unsere Lösungsansätze.

A. Problem: Suchergebnisse über mehrere Seiten

Die Twitter Suche liefert zwar sämtliche Ergebnisse zu einer Suche innerhalb eines Zeitraumes, jedoch werden nur einige wenige Tweets angezeigt und weitere können mittels einem „Weiter“ Buttons nachgeladen werden. Das Problem hierbei ist jedoch, dass Twitter diese mittels AJAX nachlädt und wir mit Curl diesen Asynchronen Aufruf nicht nachbauen konnten.

Die Lösung hierbei ist: Wir benutzen die Mobile Twitter Suche. Diese benutz, im Gegensatz zur „richtigen“ kein JavaScript, wodurch keine Asynchronen Aufrufe möglich sind und der „Weiter“ Button lediglich einen Link enthält, den wir mit Curl verfolgen können. Somit ist es möglich sämtliche Suchergebnisse zu durchlaufen und zu Ranken.

B. Problem: Twitter macht aus URLs ShortURLs

Wie in der Überschrift schon geschrieben – Twitter macht aus nahezu allen Links in dem Tweet Twitter eigene Shortlinks der Form:

https:/t.co/xyz

Dies war ein großes Problem, da wir kein Interesse an den kurzen Links haben, sondern den kompletten Link auslesen wollen. Die Lösung hierbei brachte uns erneut das Reverse Engineering, wodurch wir festgestellt haben, dass Twitter uns auch schon im tweet-container den ursprünglichen Link mitteilt. Dieser liegt in einem Untercontainer namens „data-expended-url“, welchen wir nun ebenfalls mit einem Regulären Ausdruck auslesen und für unser Ranking speichern.

C. Gültige URL oder Spam?

Ein großes Problem ist, dass Twitter oft auch als Spam Plattform herhält. Viele Spammer posten Links mit Hashtags, welche zwar aufgrund der Hashtags zu unser Suchergebnis passen könnten, leider aber inhaltlich absolut nicht dem entspricht, was wir wollen. Es musste ein Weg gefunden werden diese Links heraus zu filtern.

Die Lösung hierbei ist, dass wir die WayBack Machine von archive.org mit einbinden. Die Idee ist, dass wenn eine Seite vermeintlich brauchbaren Inhalt hat, dieser irgendwann einmal von der WayBack Machine erfasst wurde und somit auch jederzeit abrufbar ist. Der Vorteil gegenüber Spamlinks ist, dass diese nicht von der Machine erfasst werden und wir somit kein Ergebnis erhalten.

D. Keine einfache Einbettung

Ursprünglich war unser Tool sehr statisch in PHP eingebettet. Dies führte dazu, dass es nicht einfach von anderen Seiten ausführbar war.

Als Abschluss der Bearbeitungen machten wir das Tool modular, indem wir den PHP Code vom HTML/JavaScript Code trennten. Das PHP Script wird jetzt asynchron von der nachgebauten Suchmaske aufgerufen und gibt die ausgewerteten Daten via JSON Objekt an die Suchmaske zurück.

VII. Fazit

Nach den Startschwierigkeiten mit der Twitter API, die leider viel zu wenige Funktionen besitzt und uns bei keiner unserer Suchanfragen die richtigen Ergebnisse liefern konnte, bekommen wir nun mithilfe der Twitter Suche alle unsere Ergebnisse. Da die Twitter Suche sehr optimiert ist bekommen wir diese Ergebnisse nun auch ohne selbst viel Rechenleistung aufbringen zu müssen. Durch die Verwendung von AJAX könnten die Daten auch von einer anderen Plattform genutzt werden um sie auszuwerten. Die Grundbaustein, die Anfrage an Twitter, kann nun also auch für andere Anwendungszwecke verwendet werden.

Ein interessanter Punkt fiel uns leider erst zum Ende der Bearbeitungszeit auf. Twitter unterscheidet die unterschiedlichen Browser und sendet die Container je nach Browser anders. Wenn nun Curl beispielsweise behauptet ein Chrome Browser zu sein, erhalten wir von der Twitter Suche einen anderen Tweet-Container, als wenn Curl behauptet ein Firefox Browser zu sein. Hierdurch ist es möglich auch andere Informationen auszulesen, welche in unserer Version verborgen blieben.

VIII. Verwandte Arbeiten

Shine [5] ist ein Prototyp, der es ermöglicht eine Suche auf einem Teil des Webarchivs auszuführen. Hierfür wurden alle Internetseiten mit der Endung „.uk“ von 2006 bis 2013 gecrawlt. Auf diesen Seiten kann nun nach Stichwörtern gesucht werden, wobei aber nicht ein von Menschen erstellter Stichwortdatensatz verwendet wird, sondern alle Dokumente analysiert wurden. So kann eine direkte Suche auf dem Datensatz möglich gemacht werden und des weiteren Trends über den vorgegebenen Zeitraum angezeigt werden. Diese Art der Auswertung erfordert allerdings viel Rechenleistung, Speicherplatz und Zeit, da diese Daten nur von dem Zeitpunkt an zur Verfügung stehen, ab dem das Speichern der Seiten angefordert wurde.

IX. Ausblick

Durch einen Modularen Aufbau unserer Software können nun einzelne Module ausgetauscht werden. So kann die Suche auf Twitter auch für andere Anwendungsfälle genutzt werden. Mit einer angepassten Auswertung könnte so zum Beispiel überprüft werden, ob jemand etwas auf Twitter geteilt hat. Dies ist für Seiten interessant, die einen Teilen Button anbieten. Hier kann nun ausgewertet werden wie oft die Seite wirklich geteilt wurde.

Außerdem kann das Projekt in Live-Twitter-Tempas eingebunden werden, um so auch ältere Ergebnisse anzeigen zu können.

References

[1] W3 Techs, “jQuery now runs on every second website” (03.2016)

[2] Twitter API Docu, https://dev.twitter.com/rest/public (10.2015)

[3] JQuery, https://jquery.com/ (10.2015)

[4] WayBack Machine, https://archive.org (10.2015)

[5] Shine, https://www.webarchive.org.uk/shine (03.2016)