RESEARCH MANUAL

TweetMine – Twitter Sentiment Analysis Tool

KRZYSZTOF OBLAK
C00161361

Table of Contents

1. Introduction

The purpose of this document is to research and gather all information relevant to the development of a web based sentiment analysis tool for Twitter. This document will cover topics such as front end & back end web technologies, storage and similar applications/services available.

2. Web Technologies

2.1. Front End Technology

Front end technology is the technology that is running on the client side of the web technology being used. The front is the user interface to the application being used.

2.1.1. Solutions to Front End Technologies

2.1.1.1. HTML

HTML which stands for HyperText Markup Language is the core technology used to structure and represent the content on World Wide Web it is the backbone of any website development process, it is the HTML code that provides an overall framework of how the site will look. The latest version of HTML is called HTML5 which has new and efficient way of handling elements such as video and audio files.

2.1.1.2. JavaScript

JavaScript or JS for short is a lightweight scripting language that is embedded into the computer's web browser. This allows it to be platform independent. It is commonly used as a tool to make web pages more interactive with the use of its programs known as "Scripts". It is an event-based imperative programming language (as opposed to HTML's declarative language model) that is used to transform a static HTML page into a dynamic interface. JavaScript code can use the Document Object Model (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input.

2.1.1.3. CSS

CSS which stands for Cascading Style Sheets controls the presentation aspect of the site and allows your site to have its own unique look. It does this by maintaining style sheets which sit on top of other style rules and are triggered based on other inputs, such as device screen size and resolution.

2.2. Back End Technology

A back end technology is a technology that runs on the server-side of an application. The back end usually supports the front end technology.

2.2.1. Solutions to Back End Technology

2.2.1.1. ASP.NET

Open source server-side Web application framework designed for Web development to produce dynamic Web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. The development under the ASP.NET requires that the programmer use one of the languages supported by the .NET framework that are in turn compatible with the common language runtime (CLR) [1].

2.2.1.2. PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP code can be simply mixed with HTML code, or it can be used in combination with various template engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) [2] executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data.

2.2.1.3. Python

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing for the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

2.2.2. Conclusions on Back End Technology

As a result of the performed research I have decided to develop the web application using Python 3 in its latest available release (3.4.2). My decision was strongly influenced by the introduction of this language in this year’s programme as well as my personal drive to learn new languages and the expansion of my knowledgebase.

3. Twitter API

3.1. REST APIs

The REST APIs provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are available in JSON.

3.1.1. Search API

The Twitter Search API is part of Twitter's v1.1 REST API [3]. It allows queries against the indices of recent or popular Tweets and behaves similarly to, but not exactly like the Search feature available in Twitter mobile or web clients, such as Twitter.com search. It's important to know that the Search API is focused on relevance and not completeness. This means that some Tweets and users may be missing from search results.

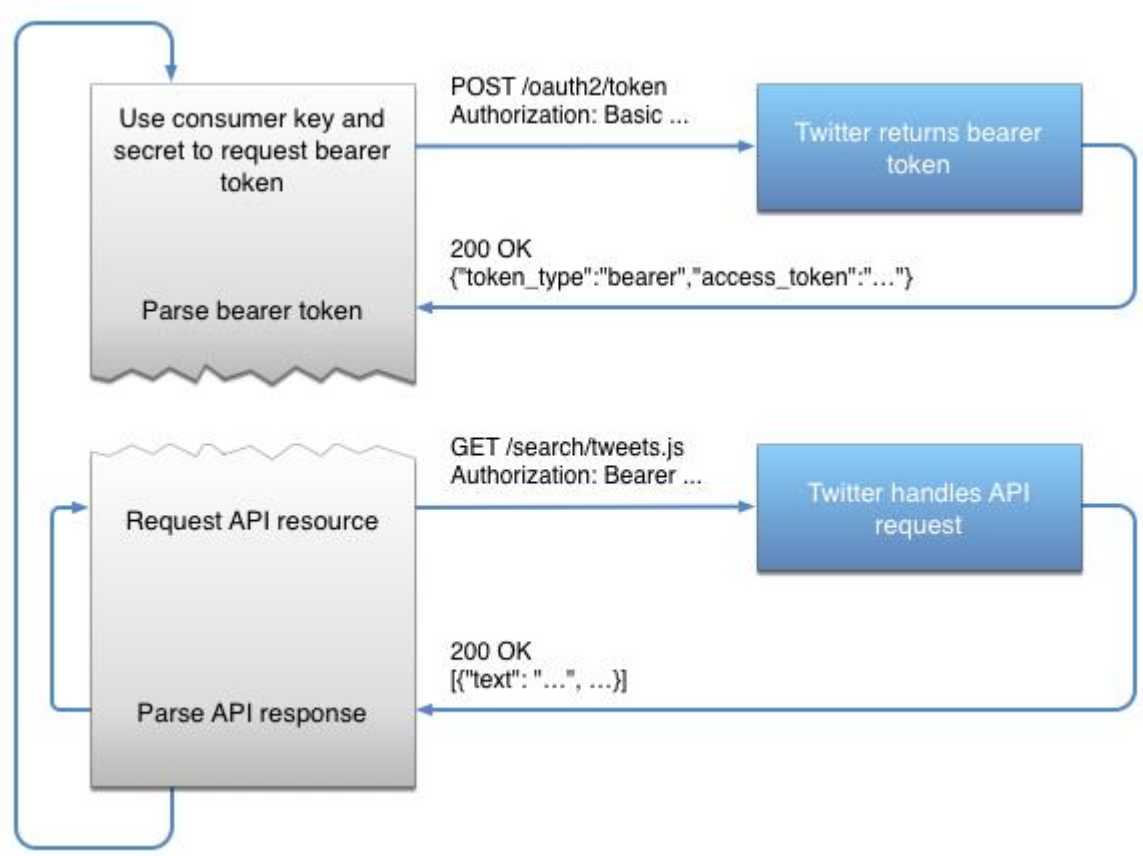
3.2. Application-only Authentication

Twitter offers applications the ability to issue authenticated requests on behalf of the application itself (as opposed to on behalf of a specific user). Twitter's implementation is based on the Client Credentials Grant flow of the OAuth 2 specification [4].

3.2.1. Authentication Flow

The application-only authentication flow follows these steps:

1. An application encodes its consumer key and secret into a specially encoded set of credentials.
2. An application makes a request to the POST oauth2 / token endpoint to exchange these credentials for a bearer token.
3. When accessing the REST API, the application uses the bearer token to authenticate.



4. Storage

4.1. Cloud

Platform-as-a-Service (PaaS) is a way to rent hardware, operating systems, storage and network capacity over the internet. The service delivery model allows the customer to rent virtualised servers and associated services for running existing applications or developing and testing new ones. Platform-as-a-Service is an outgrowth of Software-as-a-Service which provides many advantages for developers shown below:

- Features can be changed and upgraded frequently.
- Use of a single infrastructure service can reduce costs.
- Avoid risk of compatibility problems from maintaining multiple hardware facilities.
- Teams from different locations can work together on projects.

4.1.1. Solutions to Cloud Storage

4.1.1.1. Google App Engine (GAE)

Google App Engine was launched back in 2008. It's a PaaS which allows developers to run web applications on Google's infrastructure. A large number of API's are provided so that these developers can focus solely on the solution rather than having to create their

own APIs for dealing with low level services. GAE provides an application with a fixed amount of resources for free. If the application requires more resources then charges will be applied to this extra usage. An App Engine application can be written in Java, Python and PHP. Each one has its own runtime and SDK with tools for deploying the app as well as developing and testing it locally.

The following features are generally available:

- Data storage, retrieval and search.
- Communications.
- Process management.
- Computation.
- App configuration and management.

4.1.1.2. Heroku

Heroku is a cloud PaaS which now supports many programming languages. It has been in development since 2007. Back then the only language it supported only Ruby, but now it can use Java, Node.js, Scala, Clojure and Python. Heroku is not as well-known as the bigger brands like Microsoft, Google and Amazon. The platform uses add-ons which can be provisioned and scaled in a single command, and consumed by the application as loosely coupled components. They provide services for logging, caching, monitoring, persistence and more. As well as add-ons, Heroku can use buildpacks which are collections of scripts for compiling apps on Heroku specific to the frameworks and languages used in the app. It supports a set of default open source buildpacks but there are also others which can be used from the community. It can scale easily when addressing user growth, traffic spikes and demanding background tasks.

4.2. Database

4.2.1. SQLite3

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format, a file format used to persist application state to disk or to exchange information between programs.
SQLite is an in-process library that implements a self-contained, server less, zero-configuration, transactional SQL database engine, it is a compact library. With all features enabled, the library size can be less than 500KiB, depending on the target platform and compiler optimization settings. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.

4.3. Conclusions on Storage

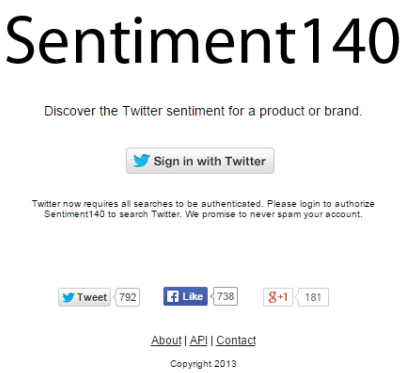
Following the performed research and its results as well as the applicability to the approach I have decided to continue the local development of the web application and refrain from cloud deployment with the lack of the required support for either the language or database system of my choice for a time being. Deployment on GAE would require both language and database change similarly to Heroku where database change would be required and also apply a monthly fee to maintain it.

5. Similar Application

5.1. Sentiment140

Sentiment140 was created by three Computer Science graduate students at Stanford University: Alec Go, Richa Bhayani, and Lei Huang. This sentiment tool requires a Sign in with Twitter as opposed to the Application-only Authentication which means that potential users will be prompted for authorisation of their personal account use with the application.

5.1.1. Home Screen



5.1.2. Search Screen

5.1.3. Result Screen

6. References

[1] Common Language Runtime (CLR)
http://msdn.microsoft.com/en-us/library/8bs2ecf4

[2] Common Gateway Interface (CGI)
http://www.w3.org/CGI/

[3] Twitter - REST APIs
https://dev.twitter.com/rest/public

[4] The OAuth 2.0 Authorization Framework
http://tools.ietf.org/html/rfc6749