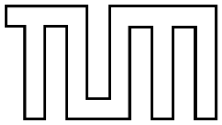

FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Application Project

Sentiment Analysis on developers' reaction to Change in API

Author:	Mainak Ghosh, Muhammad Zeeshan
Examiner:	
Supervisor:	David Soto Setzke

Situation

Monolithic infrastructure and applications that have powered the businesses over the past few decades are modularizing the business features based on independent and reusable microservices, giving birth to API. Consequently, developers can focus on own utility functions, which might need to access other processes through APIs. Besides this, there are several reasons for APIs being popular in software development industry. A new breed of third party APIs frees developers from getting stuck to any specific platform, rather they can bring their features into the market efficiently. APIs are smarter, more generic than building own SaaS solution and reinventing the wheel again. Other than Salesforce, Amazon, Facebook, Twitter, there are promising third party APIs such as Strip, Plaid for payment connectivity, Twilio for telephony, Factual for location-based data. Menlo's portfolio company Signifyd offers fraud analysis as an API. So, this horizon is interesting and revenue generator for entrepreneurs and investors. Salesforce reportedly generates 50 percent of its revenue through API, eBay nearly 60 percent [1]. These motivate developers to get into enterprise-oriented technologies such as SaaS, big data, microservices.

Besides this fruitful result, there are certain risk and challenges which developers must consider while accessing APIs. Once developer uses third party APIs for a business solution, then developer must consider security aspect of that solution, since security breach is no way acceptable. Client code developers usually evolve client application to keep the application up to date with API providers. If API is migrated, then the developers might face problem integrating the system, because API providers have little knowledge of how client code developers collaborate with the API and react to the API changes. There are other complications as well such as if the platforms (e.g. Facebook, Twitter) on which developers build APIs, are out of service, then the business features will not work. Sometimes improper documentation misguides the client code developers and cause confusion in developers' community which leads to negative reactions to those APIs. It has been observed that adding new method causes more questions to developers [2]. Another aspect is that if any API provided by Facebook, Twitter gets closed, then all the developments using that API go in vain and developers must put extra effort to get the system working. So, there are certain complications which developers have to face while accessing API services of development platform of Facebook, Twitter etc. We can see one-real example in Twitter API platform. Twitter launched their APIs in 2006 and initially every API used to have just basic authentication, so all the third-party development certainly would use that basic authentication. But after four or five years, Twitter brought the inevitable changes in their authentication which being OAuth protocol. This decision made the developer's life hard and hampered the developers-Twitter API smooth relation.

Purpose

In the era of digitalization and IoT, there are different APIs coming up so that developers can implement features more efficiently. Client developers integrate web API to their applications or services to accelerate their development and stay away from low level programming tasks. On the other hand, client code developers have no control over API evolutions and API providers are not able to find out how APIs are referred in the growing

market of third-party APIs. So, it is worthwhile to analysis the sentiment of developers over the changes of APIs. It will help API providers to figure out the despite complications, maintenance overhead, risks, which APIs developers think are useful and why. This observation will mitigate the communication gap between API providers and client code developers and be helpful for API providers to learn how developers react to API evolutions. This motivates us to work on sentiment analysis of developers' reaction on changes of APIs.

Across the different versions of APIs, response format can be changed, resource URL can be changed (e.g. the domain name of Twitter changed from api.twitter.com/1 to api.twitter.com/1.1.), response format is deleted (e.g. XML is not supported in Twitter API v1.1.), method name can be changed (e.g. this is a practice from Twitter, Blogger, MusicBranz, FriendFeed, Yelp and NYT Article Search) [2]. In this project, we will address below research questions. By answering those questions, we will figure out that for what type of changes developers give positive sentiment or negative sentiment. To do analysis, we will collect data from StackOverflow[1], Twitter[2], Reddit[3]. Nowadays, lot of people including developers discuss in these platforms, so these platforms are good for capturing sentiment and opinion. More over StackOverflow is very popular crowd-sourced media for developers [2].

Research Question 1:
What kind of API do the developers give positive or negative sentiment for?

In this part, we will find out for which type of APIs developers give positive or negative response. It has been observed that TextBlob and W-WSD are better than the SentiWordNet approach for more accurate prediction [3]. So, we will use TextBlob library for this. The textblob is a python-based library for text processing and it uses NLTK for natural language processing [4]. The approach behind this task is followed as:

a. We will clean the sentences first. This step includes removal of re-tweets (in case of tweet data), converting upper case letters to lower case, removal of stop words. Then we will use stemmer to get the cleaned words out of the sentence.
b. Next, we will use textblob to identify which sentence is positive and which one is negative. This helps to label the data which can be used for training data and validation set.
c. Besides these, we intend to implement clustering of APIs so that we get to find similarity within the APIs for which sentiments are positive and similarly for negative ones.

Research Question 2:
What leads the developers to give positive reactions for some APIs and negative reactions for others?

It's an important factor of this project where we will figure out why developer gives positive feedbacks for some APIs and negative feedbacks for others. We will search the difference in sentences between these two levels of APIs. This finding is useful for training the model which will be responsible to predict the developer's sentiment for the recent APIs.

Term frequency-Inverse Document frequency is an efficient approach which uses numerical statistics [5]. We will implement document term matrix using TfidfVectorizer() from Scikit-learn to understand significance of different words across multiple sentences in the dataset. This will guide us to identify set of words for which an API is considered as positive and another set of words which causes an API to be treated as negative.

Research Question 3:
How does our model perform in real-time survey?

After building the classifier, we will do the testing of the model. We are planning to do it in a different way than keeping a set of data out of the dataset for testing. It will indeed measure optimal accuracy of the model.

1. We will find out recent launched APIs of Facebook, Twitter, Netflix, Amazon etc and recent changed APIs.
2. We will find out developers' community where we can send survey form to gather their sentiments on those collected APIs.
3. We will execute our model on those collected API.
4. Final step is that we will compare our model's result with the survey.

This process will help us measure accuracy of our model.

Building Model

There are certain machine learning techniques such as Naïve Bayes Classifiers and Support Vector Machines(SVM) can be used for sentiment classifications. Bhumika *et al.* (2017) [6], report that SVM has better average accuracy than Naïve Bayes Classifiers. So, we will use SVM model from python-based library, scikit. To train this model, we will provide document term matrix as training set, which we will generate to answer the research question 3. We will use N gram features in document term matrix where N=1 or 2. [7] use 10-Fold Cross Validation approach to validate the model, so we will go by that.

Timeline

Project Goal	Expected completion Date
Data Collection	31.05.2018
Pre-processing of the data	07.06.2018
Sentiment Analysis using TextBlob	15.06.2018
Building the model	30.06.2018
Testing of the model	15.07.2018
Improvement of the model	31.07.2018

References:

[1]. *Matt Murphy, Steve Sloane, The Rise of APIs, May 22, 2016. Retrieved from https://techcrunch.com/2016/05/21/the-rise-of-apis/*

[2]. *Wang S., Keivanloo I., Zou Y. (2014) How Do Developers React to RESTful API Evolution?. In: Franch X., Ghose A.K., Lewis G.A., Bhiri S. (eds) Service-Oriented Computing. ICSOC 2014. Lecture Notes in Computer Science, vol 8831. Springer, Berlin, Heidelberg*

[3]. *Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine Learning-Based Sentiment Analysis for Twitter Accounts.* Math. Comput. Appl. *2018,* 23, *11.*

[4]. *TextBlob, 2017, https://textblob.readthedocs.io/en/dev/*

[5]. *TFIDF, WikiPedia, https://en.wikipedia.org/wiki/Tf%E2%80%93idf*

[6]. *Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat and Priyanka Badhani. Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python. International Journal of Computer Applications 165(9):29-34, May 2017.*

[7]. *Peiman Barnaghi, John G. Breslin and Parsa Ghaffari, Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment, 2016 IEEE Second International Conference on Big Data Computing Service and Applications.*