

SentiDash: The Social Media Sentiment Dashboard

Arthur Dodson*, Mark Fitzsimmons†

Department of Engineering Mathematics

University of Bristol

Bristol, UK

{*ad15665, †mf15726}@bristol.ac.uk

James Daly‡, Ren Jiang§, Weijia Lu¶

Department of Computer Science

University of Bristol

Bristol, UK

{‡j.daly.2015, §mu18336, ¶tt18284}@bristol.ac.uk

Abstract—SentiDash is a React-based online dashboard for companies to gain insight into their public sentiment, built on a data pipeline featuring FastText Maximum Entropy-based sentiment analysis and exposed as a web interface at *sn.mrvs.city* and public API at *sn.mrvs.city/api*. The source code for SentiDash is open-source [1].

Index Terms—computer science, data science, machine learning, sentiment analysis, Twitter

I. INTRODUCTION

With the increasing prevalence of social media within modern society, it is important for large corporations to have both a positive and impactful presence there. Social media allows companies to gauge the current feeling of the public by applying sentiment analysis models to the statements being made by the public. Twitter is a good service for companies to track their public opinion. A “tweet” - a single post on Twitter - may contain whatever the publisher wishes to express, up to a 280 character limit. With there being 500 million tweets published a day [2], Twitter should present us with more than enough analysable data in order to deduce public opinion of various organisations.

The Twitter API allows for easy collection of data from each of the organisations which we wish to perform sentiment analysis upon. There is a free option associated with streaming live tweets from their platform.

Turney [3] is credited with much of the beginning of sentiment analysis with an unsupervised learning model which analysed whether reviews of things such as cars and movies were either recommended or not recommended. Sentiment Analysis tools have become more sophisticated and accurate in recent years. This is both due to an increase in interest within this field due to both the dominance of social

media—leading to the availability of data sets of much greater magnitudes—and the increased available computing power, particularly for deep learning models. These methods have been introduced into industry in order to aid evaluate customer service responses, as well as product development.

The specific goal of this project is to create a visualisation dashboard to allow different organisations to analyse their sentiment over a specified period of time, which in this case is simply the time from when the tweets were first collected. Alongside their overall sentiment scores, the number of tweets made within each discrete time interval and the tweets with the largest amount of public interaction within that time period will also be on display. Latest news will also be shown to help companies observe how occurrences of certain events have impact on people’s opinion on them. This can be compared to a stocks dashboard offered by the likes of Bloomberg L.P. [4] and Thomson Reuters [5]—and indeed both in placing a quantifiable value on a company and tracking it over time, and in the relationship shown in recent research [6] between a company’s social sentiment and stock price, this comparison is highly appropriate.

A. Use case

This section demonstrates the use case for SentiDash. Large co-operations are constantly managing public relations (PR) and spend a fortune in doing so. SentiDash gives the opportunity for firms to have an intuitive insight into their brand within the public sphere. This could be especially useful at the time of a product release, or an important announcement for instance. The dashboard will show the change in the total number of tweets about a company, and also,

B. Data Ingress

To collect tweets, we query the Twitter API using the screen names of the companies being investigated in this project. The companies that we are researching are Nike, Apple, Facebook, Nintendo, Tesco, Starbucks, Liverpool Football Club, Golden State Warriors, Huawei, Bape, Netflix and Tesla; this will give an insight into a range of different industries. The chosen companies all dominate their respective industries: for example, Facebook, Apple and Huawei are the major players in the tech industry, Starbucks is one of the world's largest coffee brands and Bape is a popular clothing brand. In particular, these companies are likely to have a high volume of mentions in tweets every hour, giving sufficient data for temporal analysis. This is a non-exhaustive beginning to produce a sentiment dashboard for a variety of different firms. We will be using the Twitter streaming API, which allows the retrieval of live tweets in real time. There is no hard-coded rate limit for the Twitter streaming API except when the stream returns more than 1% of all tweets at that moment. On average there are 6000 tweets per second, so 1% would equate to 60 tweet per second limit. Its unlikely that our stream request would breach this threshold. To achieve a realistic analysis of company sentiment over time, our python script streams tweets 24/7.

C. Data Preprocessing

The tweets come in a wide variety of formats. Common features include tweets with misspelt words, capital letters in seemingly random positions, or certain symbols that are not useful for sentiment analysis. To train an initial model we had to make decisions and trade-offs on how to preprocess the tweets. The first task was to convert the tweets into lowercase. Capital letters do contain important sentiment information and are a way for people to mark certain parts of a tweet as important [7]. However, it increases the complexity of the model and tweets with capital letters in seemingly random places are more likely to give adverse results. Then we removed certain symbols. The "@" sign is a common symbol in Twitter because it precedes the identification name of all Twitter accounts, for example, you would mention Apple using "@apple". This symbol is not

useful in sentiment analysis. Emojis are ideograms created by a specific pattern of symbols. They can represent smiley faces, or unhappy faces and represent an important indicator of sentiment within a tweet. Therefore, we are not removing other symbols. All "retweets" in Twitter start with an RT which are also removed because it contains no information about positive or negative sentiment. These processed tweets can now be used to test the model, and consequently, to label the tweets with a sentiment score before they are stored in a database.

D. Data Storage

From consistently streaming tweets for only 5 weeks, we have stored 3 million tweets, equating to 17.7GB of data. Therefore, we believe that a cloud-based environment is needed to store this large volume of information, and to process the tweets continuously. There are a variety of different services that meet these requirements and for this project we have decided to use Amazon DynamoDB. DynamoDB can handle 20 trillion requests per day, and support peaks of 20 million requests per second. The data will be in JSON format, where each tweet represents a dictionary with different keys. Each dictionary contains a multitude of information including the tweet, the date, the country, etc. DynamoDB is also a NoSQL database and it works well with the JSON format.

By creating different indices on DynamoDB tables, flexible querying requirements can be satisfied. In our case, the most recent tweets are retrieved for frontend display which requires a partition key on company name and a sort key on timestamp.

E. Data transformation and integration

For visualisation purposes, we introduce temporal analysis for data transformation and integration. Some statistics such as mean and standard deviance are needed for visualisation, in order to understand the error that exists within the model. It is not feasible, however, to query all the tweets required (e.g. tweets of one week) from DynamoDB and then calculate the statistics in real time as the data volume is too big.

Instead, these statistics are precomputed and stored in another DynamoDB table for real-time

reading by the frontend. Specifically, an AWS Lambda function is created and triggered by a CloudWatch event every day to aggregate and pre-computed statistics for the previous day. According to different granularity requirements, different time intervals can be configured. This method of temporal analysis is known as a tumbling window.

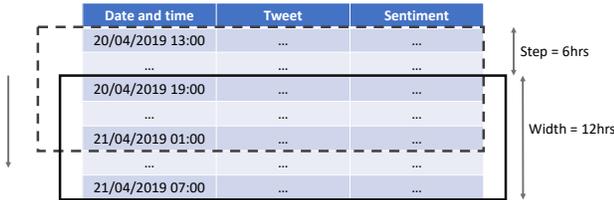


Fig. 3. A schematic to describe time window analysis of consecutive overlapping sections.

On the other hand, in order to gain finer-grained sights of people’s attitude towards companies over time, another temporal analysis method is used. This is performed by taking consecutive, overlapping sections of the time series data, commonly known as a rolling window. The process is defined by the window width and the time step between each window. A schematic of the process can be seen in figure 3. Within each window we can apply different techniques for analysis. A simple, yet interesting metric is the change in number of tweet mentions a company has. This could be useful for company’s by indicating if the news of their announcement or product has sufficiently spread. This can be calculated by aggregating the number of tweets about a company within a time window (n). In addition, the average sentiment regarding their company can be tracked in a similar way by taking the mean sentiment in the time window:

$$\bar{S} = \frac{1}{n} \sum_{i=0}^n S_i, \quad (1)$$

where S_i is the sentiment—in the range $[0, 100]$ —of the tweet on row i . The choice in window width is a balance between results that are too noisy and too smooth. We will be using a 12 hour window width and a step size of 6 hours (can change these parameters).

Another useful metric is standard deviation, which is a useful insight for volatility or, in our case, contentiousness. This is easily calculated:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=0}^n (x_i - \bar{x})^2}, \quad (2)$$

In preparation for visualisation, these aggregates are stored in a DynamoDB table (although a time series database would appear more appropriate, the lower pricing, automated management and interoperability with other AWS services make DynamoDB a convenient choice). An API deployed on Amazon Lambda (a ‘serverless’ function execution service) and managed through Amazon API Gateway makes this data available to end-users through a URL.

F. Sentiment Analysis

1) *FastText*: There are many different approaches to sentiment analysis, with the main split being between the lexicon and machine-learning based approaches. Lexicons are used to describe the vocabulary of a language, an individual, a group of individuals or a subject matter. These lexicon approaches can be further split into Dictionary based approaches, used by tools such as Vader, or they make use of a Corpus, like the Stanford Natural Language Corpus. The Machine-Learning models are of more interest to us as they allow us to build and fit our own models depending on their required purpose. In this case, it was decided that FastText [8] would be used in order to build a maximum entropy model. The probability of a tweet having a particular sentiment equals the probability which maximises the entropy of that system.

$$H(x) = \sum p(x) \log_2(p(x)) \quad (3)$$

The main justification behind this is that we should accept the most uniform model available to satisfy all available constraints. We should not infer information and keep the model general as a result. Nigram et al. [9] found that MaxEnt models were superior to Naive Bayes classifiers in classifying various webpages and articles into their specific categories. FastText in particular is often performs just as well if not better than deep learning models, but with a much shorter training period due to the

lack of backpropagation, and the multiple layers required for the model to be suitable complex.

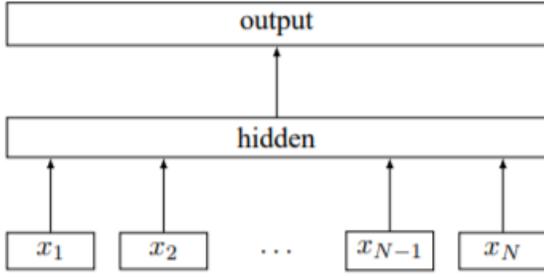


Fig. 4. FastText model architecture, where each x represents a word from a tweet of N length. The hidden layer of the model involve the bag of characters and softmax functions detailed later.

FastText aims to minimise the negative log likelihood over all of the given classes. In this project, there are three different classes representing positive, negative and neutral sentiment. Given x is the normalised bag of features, y_n is the label of the analysed tweet, while A and B equal the weight matrices used, this can be expressed as Equation 4.

$$-\frac{1}{3} \sum_{n=1}^N y_n \log(f(ABx_n)) \quad (4)$$

One of the assumptions of this model is that it assumes that the words are not independent of one another, and so a bag of words approach is undertaken. This means that the model only takes into account the list of known words, and how many of each word exists within the statement. It takes no notice of the structure of the statement. FastText also uses both a bag of characters [10] approach, also known as an n-gram. Each word is split into 3 character long strings. Boundary symbols are added to distinguish the subsection of a word from a full word itself. So for example, the word “beat” would be split into <be, bea, eat, at>. The vector for “beat” would subsequently be the sum of the vectors for each different n-gram. If the word eat was being represented it would be written as <eat>. These n-grams are effective in capturing information relating to the local word order, and can help identify word groupings based upon things such as the prefixes.

Removing frequently used words from the training data is another important tool in sentiment analysis. Many common words such as “and” or “the”

offer little to no information regarding the actual content of a sentence. To avoid using frequently repeated words, FastText discards training words with a distribution of:

$$p(w) = \frac{c_w}{t_w} \quad (5)$$

where $p(w)$ equals the probability of the word being removed, c_w is the total count of the word and t_w is the total count of all words.

2) *Training Process:* In order to train the sentiment analysis models used within this project, labelled data had to first be obtained. A data set [11] of approximately 3 million tweets about professional football teams is chosen, which were labelled using Amazon Comprehend, Amazon’s sentiment analysis module on the AWS. Before the training, the tweets should be cleaned first to remove the part unrelated to sentiment. The elements including <@,#,URL,HTML tags>are deleted. After removing the unrelated part, the emojis are replaced by the corresponding plain expression. Emojis play an important role in expressing sentiment so that it should be replaced by plain word rather than easily deleted. What’s more, as about 70% of the labels are neutral, upsampling is required to avoid the model’s preference of predicating neutral. We simply copy the positive and negative tweets to make the training set more balanced.

During the training process, the model is trained

Number of tweets	Accuracy on Training Set	Accuracy on Validation Set
3M	84%	81%
1M	84%	80%
500K	86%	80%
200K	88%	78%
100K	90%	75%
10K	96%	67%
1K	73%	50%

TABLE I
ACCURACY OVER SIZE OF TRAINING SET

step by step with size of training data gradually increasing. To evaluate the model, a validation set of independent football tweets about world cup event is selected. And the accuracy index is chosen to evaluate the model. As Table I shows, the number of training tweets we use increases from 1K to 3M. As more data is used for training, the accuracy of training set decreases from 96% to 84% as it’s

less overfitting. And the accuracy of validation set always increases with the increase of size of data set. It reaches the bottleneck when training set reaches 500K tweets. Finally this model achieves 81% accuracy on validation set and then used to classify the Twitter data streamed from the API.

G. Data Egress and Visualisation

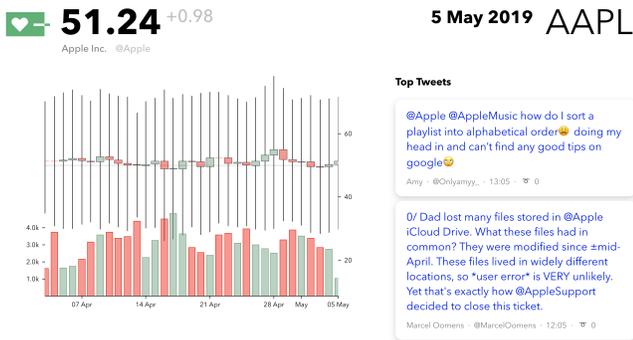


Fig. 5. Example Dashboard recording example sentiment scores and tweet numbers from Apple.

1) *Dashboard*: The dashboard itself is a React [12] application stored on Amazon S3 and served to visitors through CloudFlare¹. The application displays, for a given company, its social media handle and stock ticker (for easy referencing against traditional stock monitoring services) and an interactive candlestick graph displaying sentiment score and tweet volume over time. The graph’s ‘open’ and ‘close’ markers (candles) map to the average sentiment and ‘high’ and ‘low’ (wicks) map to 1 standard deviation, as we would expect consistently extreme values—as such, standard deviation more aptly communicates the volatility or divisiveness of a company’s public sentiment compared to the maxima and minima.

In addition, for a time selected on the graph (or by default the last few hours) most ‘significant’ tweets, hashtags and news stories (from Microsoft Cognitive Services Bing News Search) are displayed. This can be used to track trends in sentiment, and derive insight on events which cause major contention or shifts in public perception of a company, or major

¹In development, the frontend is deployed from GitHub using Netflix instead, for accelerated deployment and simpler domain management.

shifts in company value (as, before mentioned, there is growing research which shows a correlation between social sentiment and stock price).

The graph format and application layout have been chosen to maximise familiarity, and therefore utility, to our intended audiences of business people and investors.

2) *Server*: For the server, we employed a serverless architecture on AWS consisting of API Gateway, Lambda functions and DynamoDB. This architecture is powerful, cost-effective and is relatively simple to develop. In our case, the functionality of the server can easily fit within a single Lambda function, and we don’t need to deal with manual resource provisioning. Furthermore, serverless architecture is the easiest way to achieve auto-scalability which is useful for the potential commercial development of this project.

Since we also are also interested in the impacts of certain events on people’s attitudes towards companies, we incorporate the latest news for data visualisation, which makes it easy for human analysts to observe the correlation in a straightforward way.

As a result, when a Lambda function is invoked by an API request, it will query the precomputed daily statistics from DynamoDB for the given period of time, invoke the Bing News API to search for related news, fetch top popular tweets from the tweets table in DynamoDB and return the collected information as a JSON object.

III. RESULTS

A. Temporal Analysis

From the results in Figure 6, we can see the change in average sentiment over time for Nike, and also, the standard deviation in this sentiment. The standard deviation appears to be consistently high. It also does not appear to show any significant results. If it were to be showing significant results, then it is not clear from this figure. It is also diverting attention from the more interesting results of the average sentiment. This is likely to be due to the nature of the rolling window analysis. However, it is likely to be more suited to the tumbling window analysis that will be displayed in the dashboard.

In Figure 7, we can see the average sentiment of Nike along with a background bar plot of the number of tweets in each time window. There is a

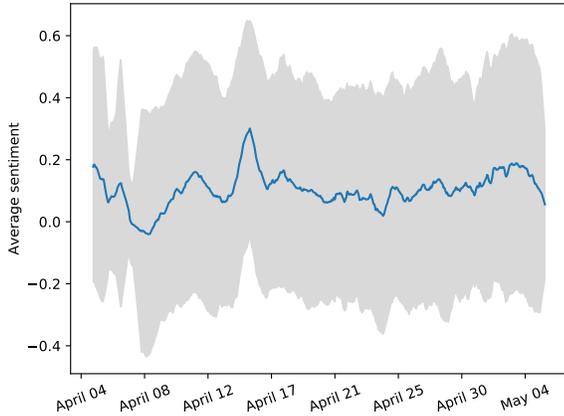


Fig. 6. The plot shows the rolling window analysis of the average sentiment of Nike for each time window is depicted in the blue line. The grey shaded area is bounded by an upper and lower limit of 1 standard deviation from the average sentiment during the time window.

clear trough and peak in this graph highlighted by the overlaid green circles. On further investigation, they also respond to two key news events about Nike. On 7th and 8th April, there were prominent news articles from sources such as NBC and CNBC about bribery charges related to college sport. This is a particularly sensitive topic in America at the moment and likely to attract significant media attention. The graph shows a steep drop in average sentiment for these dates. Moreover, the famous Masters golf championship was between 11th to the 14th April where a significant amount of Nike advertising would be on view to millions of people. Then Tiger Woods, who has a long history with the organisation, won the championship. This led to a number of news articles on 15th April praising Nike for their continued support of Woods who has been dogged by scandals in recent years. With this knowledge, it is not surprising that we see a steep increase in the average tweet mentioning Nike near the 15th April. As a result, Nike's company value rose by \$2 billion [13] after the victory. In addition, the bar chart indicating the number of tweets about Nike shows a big spike around the 14th April. This provides evidence that the news regarding Nike is spreading to a high volume of people. The high volume of tweets and the positive sentiment in this

instance, indicates their PR strategy has worked effectively.

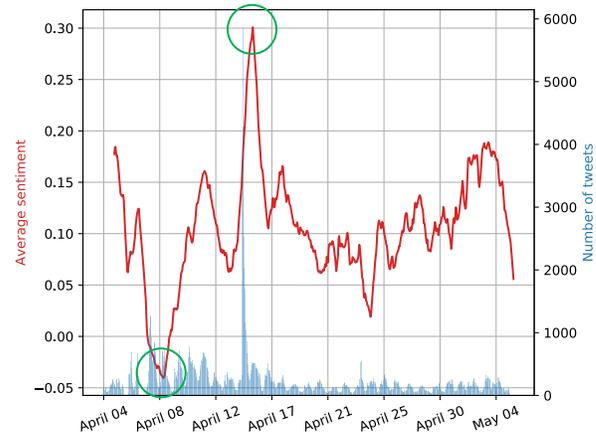


Fig. 7. The red line shows the average sentiment of Nike plotted over the whole time period of our data. The bar plot shows the total number of tweets for each time period. A moving average of 20 data points has been applied to ease visualisation. The green circles outline the minimum and maximum sentiment in the graph and also refer to key events discussed in the main text.

From the results in figure 8, we can see the change in average sentiment for Apple and the background bar plot of the number of tweets in each time window. This is another example of two clear deviations in average sentiment. Apple and Qualcomm were involved in a significant legal battle [14] that was ended in a damaging settlement by Apple. This was covered heavily in the mainstream media on 17th April and one would expect negative sentiment regarding the company in social media. This again can be clearly seen by a steep decrease in average sentiment in figure 8. This was also further confirmed on viewing a sample of tweets regarding Apple on 17th April. In a similar manner, the spike in average sentiment about Apple around the 28th April is also backed up by significant news coverage of a security feature [15] that also makes it easy for people to view how much time they are spending on specific applications. This was met with positive sentiment when analysing the tweets in social media. This would be useful information for Apple regarding an issue that could have arguably led to negative sentiment on social media.

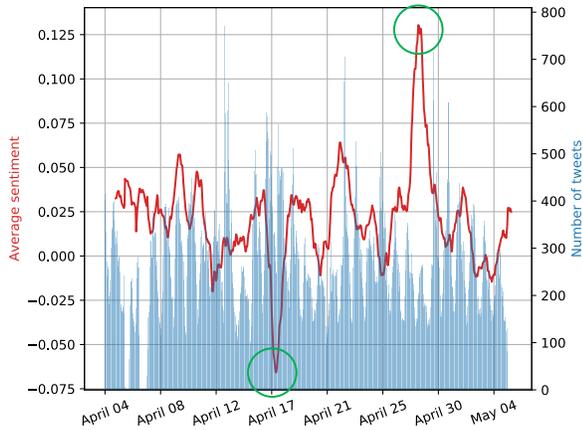


Fig. 8. The graph is constructed in the same manner as figure 7. The figure shows the average sentiment for Apple and the number of tweet mentions for the company over time.

B. Sentiment Analysis

In order to find limitations within the model it is important to look at some case of success and failure. Within tables II and III, tweets collected from Apple and Nike are presented with associated sentiment scores. The accurately labelled tweets are shown above the black bar, with those worse below. The model appears to take issue in associating Tiger Woods’ success with Nike, and thus reviewing it positively. However, tweets such as “This new @Nike ad congratulating Tiger on winning #TheMasters is everything” received positive sentiment scores (in this case, 0.537). As the case for Apple, the first occurrence of failure is a case of misinterpreted sarcasm, as well as a lack of information or context in what is being said: the statement of “Thanks” at the end, with the tweet also including phrases of positive sentiment such as “so much better” is likely causing the issue. As for the other case of failure, it is hard to identify the tweet to be of negative sentiment, as there are not any clearly negative words used. However, in this case the user is complaining about a minor issue which exists within the maps service Apple provides, as so some negative score should be assigned. In the cases where the company has been tagged before the tweet, the tag has been removed.

Tweet	Sentiment Score
Yep...been using reader ever since I got my iPhone! Helps greatly when using VoiceOver. :)	0.622
You can see they are not responding. I invested my money in Apple and now it's violating customer laws	-0.288
When my payments are up on this iPhone, I'm going back to android. It was a mistake to even try an iPhone again.	-0.704
Predictive text was so much better when Steve Jobs was still alive. Thanks @Apple.	0.700
Apple why do you think I am in Portland? I've never been to Portland (unless you mean Portland Maine)	-0.003

TABLE II
EXAMPLE TWEETS INCLUDING THE @APPLE SCREEN NAME, WITH ASSOCIATED SENTIMENT SCORES.

Tweet	Sentiment Score
S/O to @flightclub for my @Nike Huaraches ! Yeyuuuhh never received a package so fast	0.452
Dan Kane, !"Carolina did what?!	-0.704
very disappointed in the quality of your products. Had this sweatshirt for only a few weeks before...	-0.997
There are a million reasons to support Nike. Tiger is only one. This is why Nike is #1 in the world.	-0.007
Props to Nike for always standing behind Tiger and not dropping him or turning their back	-0.008

TABLE III
EXAMPLE TWEETS INCLUDING THE @NIKE SCREEN NAME, WITH ASSOCIATED SENTIMENT SCORES.

Another limitation of the model is that it is less sensitive to topics other than football which leads to a worse accuracy when tested. Besides the validation set, two manually labelled test sets, which respectively contain 1000 tweets on various topics, are used to test the model. As table IV shows, the model achieves 55% accuracy when tested with the validation set covering multiple topics, compared to 72% accuracy achieved when tested by the set covering football only. Despite being difficult to label tweets manually and somewhat impossible to evaluate the accuracy of manually labelled data, as the two datasets are with same size and labelled by same person we can assume the two datasets have a similar accuracy. Therefore, we can infer that the model is more sensitive to the football topic and has a higher accuracy when predicating football tweets.

IV. FUTURE WORK

The ultimate goal for this project is to be a fully commercial platform for businesses to track global public opinions of their brand and the economic

	multiple fields test set	football field test set
Accuracy	55%	72%

TABLE IV
ACCURACY BY DIFFERENT TEST SET

effect on their business. There are many things we can do to achieve this.

Firstly, we were manually selecting brands or companies we are interested in. We can improve this by crawling data from company information databases such as [16] and store it in our database. In the meantime, the tweet streaming program should be able to watch the change of this database and track tweets mentioning new companies.

Secondly, since we want to provide a internationally relevant service, different regions and different languages need to be supported. Specifically, the tweet tracking program needs to be modified to track not only English tweets but also other languages; in addition, separate sentiment analysis models would need to be created for each language. Besides, different countries may have their own trending social media platforms—for example, the Chinese counterpart of Twitter is Sina Weibo [17]—so we could also track these platforms.

Thirdly, human observation is required to correlate events and their impact on change of public sentiment over time. In the future, we could add a feature that describes the most popular tweet topics about a company at a given time. This would help the company by giving more information about the tweets and the changes in sentiment.

Fourthly, accurate predictions act as the cornerstone of our product. As mentioned before, there is a issue with the sentiment analysis model that it was trained using a limited data set of all football related tweets. To deal with this disadvantage, a data set of different fields needs to be used to train a robust model. A solution is using the Stanford Natural Language Corpus [18] to label a subset of tweets taken from the database. To assemble the corpus the labelling of compatible sentences was crowdsourced using the Amazon Mechanical Turk, and it contains 570,000 of these sentence pairs. This will then be used as the training set to train a new model. This model should be more capable at correctly assessing the information within each tweet as it has been

trained on the more generalised data set as opposed to the football specific one.

We too could further develop our model by combining natural language processing and regression analysis on historical news and sentiment changes; for instance, we can develop an agent that continuously watches news events from multiple sources on the internet. When a piece of new of a company is captured, a feature vector can be retrieved from the news and then it can be used as input to predict sentiment changes in the coming hour using the regression model we learned from historical data. This is especially useful for companies to take actions on public relation to handle emergencies.

Finally, competitor analysis can be provided as an added-value service to companies using SentiDash. It is an important aspect for most businesses because events don't normally have influence on a single company but the whole industry. For example, it would be valuable for Microsoft to know the change of people's opinions on Xbox One when new games are released on the competing PS4.

With the elastic computing powers provided by cloud services on AWS, this project can be extended without worrying about most of operation issues. Also, existing tools for data analysis and machine learning have made this goal achievable.

V. CONCLUSION

This project has given rise to the creation of SentiDash which can be seen to be an effective and useful tool. The methods described have been created building on current data science practices and utilising an efficient cloud-based environment: this has led to a scalable and computationally powerful structure. The project has discussed various approaches to creating a sentiment model of which there has been a thorough testing phase with hand-labelled data, and the results show a clear indication of the effectiveness of rolling window analysis of the sentiment. The two examples shown (Nike and Apple) indicate clear instances of high and low sentiment regarding the companies, clearly linked to their respective media attention. This would then give the said company a gauge for social media sentiment with regards to their media attention, and the effects on their business. This is made available

through an API and dashboard interface which is familiar to investors and business people.

REFERENCES

- [1] J. Daly and R. Jiang, “Antisocial Network on GitHub,” May 2019. <https://github.com/antisocial-network>.
- [2] “Twitter usage statistics.” <https://www.internetlivestats.com/twitter-statistics/>.
- [3] P. D. Turney, “Thumbs up or thumbs down,” in *ACL 2002*, 2002.
- [4] Bloomberg, LLC, “MSFT:NASDAQ GS stock quote – Microsoft Corp. – Bloomberg Markets,” May 2019. <https://www.bloomberg.com/quote/MSFT:US>.
- [5] Thomson Reuters, “Thomson Reuters Eikon,” May 2019. <https://eikon.thomsonreuters.com/>.
- [6] E. Birbeck and D. Cliff, “Using stock prices as ground truth in sentiment analysis to generate profitable trading signals,” *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2018.
- [7] S. Chan and A. Fyshe, “Social and emotional correlates of capitalization on Twitter,” in *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, (New Orleans, Louisiana, USA), pp. 10–15, Association for Computational Linguistics, June 2018.
- [8] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” in *EACL*, 2017.
- [9] K. Nigam, J. D. Lafferty, and A. McCallum, “Using maximum entropy for text classification,” 1999.
- [10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [11] “BetSentiment dataset.” <https://betsentiment.com/resources/dataset/english-tweets>.
- [12] Facebook, Inc, “React, a JavaScript library for building user interfaces,” May 2019. <https://reactjs.org/>.
- [13] CBS News, “Tiger Woods’ Masters win a \$2 billion hole-in-one for Nike stock.” <https://www.cbsnews.com/news/tiger-woods-2019-comeback-nike-shares-rise-on-woods-masters-win/>.
- [14] The Washington Post, “Apple said Qualcomm’s tech was no good. but in private communications, it was ‘the best.’” <https://www.washingtonpost.com/technology/2019/04/19/apple-said-qualcomms-tech-was-no-good-private-communications-it-was-best>.
- [15] BBC News, “Apple defends removing parental control apps.” <https://www.bbc.co.uk/news/technology48092151>.
- [16] “Corporateinformation.com.” <https://www.corporateinformation.com/>.
- [17] Sina Corp, “Weibo.” <https://www.weibo.com>.
- [18] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” in *EMNLP*, 2015.