X (formerly Twitter) as a Free and Accessible Source of Public Sentiment

Using Computational Techniques to Analyse Brexit Sentiment on X

Brodie Knight

University of Cambridge

Laidlaw Leadership and Research Scholarship

nk624@cam.ac.uk

Supervised by Prof. Andreas Vlachos

1 Abstract

The aim of this research project is to (1) investigate and develop a method of extracting large volumes of data from X at a low cost, and (2) examine the usefulness of this data as an indicator of public opinion on the topic of Brexit. This paper approaches the topic both from the practical side of obtaining the data and the predictive power of the data. Unlike the main corpus of research in this area which attempts to evaluate how social media data *complements* traditional survey data [1], this paper evaluates the power of social media data *on its own* as a low cost solution to acquire data on public opinion. Given the expense of conducting a traditional survey and the relative ease of acquiring social media data, it could potentially be a cheap alternative to gain insight into public opinion. After the changes to the X Application Programming Interface (API) pricing structure, access to X data is more prohibitive, rendering the techniques described in this report more relevant than ever. Various methods of data access will be outlined in this paper, before focusing on the scraping method custom developed and used as part of the project.

The three core outcomes of this project are as follows:

1. A large volume of Tweets can be fetched free of charge by using a 'distributed tor circuit method' with up to 250 tor circuits operating asynchronously. This method could save thousands of pounds and yields an output of over 17 million tweets per day (a rate of over 12000 tweets per minute was observed[1]). This gives 17 times more Tweets in one day than the $5000 per month X API subscription[2] would give in a month. It is important to note that so as not to overload the Nitter servers the scraper was run at less than 50% for the majority of the Tweets downloaded (hydrated) in this project. The software programmed as part of this project demonstrates this, acting as a proof of concept, and the Tweet data used for the other parts of the project was acquired with this software.
2. Key word extraction of the X data gives quantitative insight into the topics of public discussion at the time. Sentiment analysis can also be done on these Tweets to give some idea of whether the discourse around this keyword was positive or negative.
3. There is no statistically significant correlation between the aggregated sentiment scores of Tweets about Brexit and national polling figures on opinions towards Brexit.

A complete suite of software was programmed as part of this project. Throughout this paper, technical aspects of the code will be described. Any details may require deeper technical knowledge are inlined with a vertical bar on the left as seen below.

> Rust was chosen as the programming language to deliver soundness and speed while processing large volumes of data. The techniques outlined in this paper are implemented in Rust code which is split into 2 separate repositories:
>
> - `nitscrape` (`https://github.com/Merlotec/nitscrape`) - a Rust library that provides the Nitter scraping and tweet hydration functionality via distributed tor circuits. This is where the majority of the complexity is in the project.
> - `tsa` (`https://github.com/Merlotec/tsa`) - the Rust command line executable that depends on the `nitscrape` library and provides a user facing interface to facilitate all the processes described in this paper.
>
> In order to build from source the `libtorch` C++ libraries must be installed. It is also recommended not to use WSL to run the Natural Language Processing functionality since given the volume of data involved, full GPU acceleration will significantly lower the processing time.

It is very important to note that the scraping techniques described in this paper were effective at the time of research but may no longer work due to subsequent updates to the Nitter website.

[1] This figure includes Tweets which were discovered to have been deleted and therefore were not ultimately hydrated.

2 Acquiring Social Media Data

There are three main ways to get data on X and these methods roughly apply to most other social media platforms.

1. Using the social media platform's own API to fetch data from the platform. This can be free for small amounts of data, but given the volume required to investigate public opinion a paid plan is necessary.
2. Using a third party company to fetch, package and deliver data from the platform using a search criteria. This can cost a significant amount of money, which depends on the amount of data required and the way it is packaged.
3. Using custom scraper that 'scrapes' data from the social media platform's publicly facing website. The precise way in which this is done can vary substantially between different platforms, since websites often establish measures to reduce scraping activity.

2.1 The API Method

Accessing social media data through an Application Programming Interface (API) is the 'idiomatic' way to do things. It involves writing computer code that communicates with the social media platform directly via an 'official' communication channel that allows us to send and receive information to/from the platform.

With regards to X, there is an API available. However, during the research period this was prohibitively expensive for this project, and by extension would not be suitable for many occasions for this reason. X charges $5000 USD[2] per month for the pro developer subscription which gives access to 1 million tweets each month. It is also important to note that the X API gives access to many more complex querying features that are not available using the other methods.

2.2 Third Party Method

Hiring a third party company to collect the data requires less technological knowledge than the other two methods. This makes it an appealing option for small companies/projects where tech experience is limited. Additionally, large data collection companies often have greater access to data due to economies of scale.

Once again, given the volume of data required, this approach was too expensive. The price that was quoted from several companies was around $6000 USD for 1 million tweets.

2.3 Scraping Method

It is possible to access social media data for 'free' by *scraping* the data from the website of the platform. This works by programming software to download the web pages that would normally be viewed by human users. The software looks through the website data and extracts the social media content from it.

This method also has some serious limitations:

- It requires technical/programming knowledge.
- It takes more time since there are rate limits which restrict the rate at which web pages can be downloaded.
- There are often security measures in place to prevent scraping.
- It can be difficult to query or fetch a true random sample of data.

However, these limitations can be overcome. For the purposes of this project, this was the only possible solution, since the other two options were simply not affordable given the volume of Tweets required.

3 Scraping Data from X

For this project the scraper approach was used. Scraping is the only approach that is appropriate for this project since the purpose of this project is to investigate whether X can be used as a *low cost* data source for public opinion as an alternative to polling, and the other two options are simply too expensive to meet this constraint.

3.1 Fetching Tweets from X About a Topic

There are large datasets available online containing the Tweet IDs of Tweets about to a specific topic, which can be accessed and used freely. The datasets do not contain the actual tweet data themselves, but we can look up the Tweet contents using the IDs in the dataset. This process is known as *hydrating*, since it is essentially transforming a list of IDs into a list of Tweets. In this case, the core purpose of the scraper is to hydrate a precompiled list of Tweet IDs.

A dataset from the Harvard Dataverse[3] was selected to provide a full set of Tweet IDs of Tweets about Brexit. The dataset contains all the Tweets mentioning Brexit from January 2016 to September 2019. The dataset also provides some other metadata along side the Tweet ID, including their own sentiment analysis scores. However, since there is no way to validate these sentiment analysis scores, and since we require the full hydrated Tweets anyway for more in depth analysis of keywords, these sentiment values were ignored.

> A `csv` file was downloaded from the Harvard Dataverse[3] with all the Tweet IDs of Tweets pertaining to our desired topic (in this case Tweets relating to 'Brexit' between 2016 and 2020). This data set contains `tweet_id`, `user_id`, `sentiment` and `stance` fields. The `tweet_id` field is the only important field since it gives us a way to reference the Tweets that we're looking for. The `stance` field was found to be inaccurate on many of the Tweets so was ignored.

3.2 Using Nitter as a Proxy for X

The process of hydrating requires a way to get the content of a Tweet from its Tweet ID. The problem with using the X website directly is that there are rate caps restricting the number of Tweets that a user can access. These rate limits are so restrictive that scraping using the X website is not feasible.

The website 'nitter.com' provides an alternative way to fetch Tweets. Nitter is a lightweight *skin* for X that was designed to be less distracting and addictive. The benefits of using Nitter are:

- There is no requirement to sign in.
- The Nitter rate cap for fetching Tweets is higher than X's (via scraping).
- The Nitter web pages are much smaller in size which means they take up less internet bandwidth.
- There is less bot detection.

The faster the Tweet hydration rate the better. Since we are fetching the data from a web page, we want the web page to contain as little unnecessary data as possible (e.g. advertisements, images, scripts, user interface elements).

> Unlike X, Nitter does not hide the Tweet behind any javascript and can therefore be scraped very efficiently. Furthermore, the like, retweet, quote and comment counts are all easily parsed.
>
> Looking up Tweets on Nitter is very similar to X. The url takes the schema `https://nitter.net/<user>/status/<tweet_id>`. The `<user>` path segment is actually ignored by the server and does not have to match the user of the Tweet being searched. In fact, this can be any arbitrary string and the request should still work.

The Nitter Tweet fetch rate is server limited to approximately 180 Tweets per minute per IP address. However, if they detected any IP addresses that were consistently making requests in suspicious ways, they deny access to those addresses.

3.3 Routing the Requests Through Tor

The Tor network, also known as the dark web, can be used in order to overcome the rate cap and suspicious IP address denial. The Tor network provides almost complete anonymity which gives it an appeal for criminals and civilians living under censorship. These benefits are also useful for scraping, since it allows us to avoid some of the usual access prevention methods that websites use to stop bots from getting access and overloading them with requests. Tor can be thought of as a special kind of proxy server, meaning that network data is routed via a 'middleman'. The consequence of this is that the website being accessed does not see our IP address, and instead only sees the IP address of this middleman. This allows us to change the IP address we use to access Nitter if it is blocked.

In total, **22 006 475 Tweets were fully hydrated** out of 50.8 million Tweet IDs (about half were unavailable as they were deleted by the user or the sender's account was removed).

> We establish n Tor circuits each on ports $[7000, 7000 + n)$ which act as proxy servers. To do this, we need a separate working directory for each tor instance each with its own `torrc` configuration file. The optimum number of circuits depends on the bandwidth and computing power available, since running many tor circuits which are each handling a steady stream of requests is highly performance intensive. Between 150 and 250 circuits seemed optimal, however anything above that was too unstable to use. The maximum download/hydration rate achieved in testing was around 12000 Tweets per minute with 200 circuits running. However, for the sake of the Nitter servers it was run at around $5000 - 7000$ Tweets per minute while hydrating the Tweets used for this project.
>
> For every circuit, a separate task was created inside an async runtime using Rust's async functionality. Each of these tasks independently manages its own circuit, including sending requests and receiving responses. The Tweet IDs are read from the dataset and added to a queue which is read by each task when it is ready to send a new request. Given the very large latency through the tor network, multiple requests are sent per circuit asynchronously, meaning that we do not have to wait for a response before we send the next request. A maximum of 8 requests per circuit in progress at any one time was found to be the optimum - any more than this did not yield any performance benefits, probably due to a bottleneck on the bandwidth of the Tor circuit.
>
> Each Tor circuit task also tracks its performance such that if it is substantially below the average download rate of all the circuits, that circuit will be recreated. This did not seem to improve performance much, and actually reduced it if the threshold at which the circuit is recreated was too sensitive, likely due to the increased bandwidth and processing required to initialise new Tor circuits.
>
> Once responses are received by the tor circuit tasks, they are all sent back to a single task which is responsible for scraping the response and hydrating the Tweet. The hydrated Tweet is then written to the output `csv` file containing the hydrated Tweets.

4 Processing the Tweets

Once the list of hydrated Tweets is acquired, they are then processed, which assigns a language (e.g. English, French) and a sentiment score between $[-1.0, 1.0]$ with -1.0 being the maximum negative sentiment and 1.0 being the maximum positive sentiment. The `DistilBERT`[5] Sentiment Analysis model was used in to determine these sentiment scores. The analysed Tweets are then saved to another `csv` file containing the language and sentiment score.

4.1 Aggregating the Sentiment Data

The data can then be aggregated into a time series with an arbitrary number of data points. To do this, an average of all the sentiment scores within each time span is taken which gives us the value for that data point. For this project, a time period of 4 days per data point was chosen which resulted in anywhere between 1000 and 10000 Tweets per data point. This data was written to another `csv` file and graphed (see Figure 1 as an example - the blue line is the aggregated sentiment score).

4.2 Key Word Extraction

Key word extraction involves identifying the key words in Tweets to help us determine the topic of the Tweet. Most commonly it is done using Natural Language Processing techniques[6], however due to the large volume of Tweets at our disposal, a different approach was taken. The most commonly occurring words among all the Tweets in each time period (of 4 days) were collected and established as the 'key words' during that period, ignoring any neutral words that commonly appear such as prepositions, pronouns and common verbs.

> A list of words to ignore is used to filter out the common words. For each time period, a hash map is constructed, and every word (ignoring the words in the ignore file) is added to the map with the word itself as the hash key. If the word already exists then the occurrence value is incremented by one. The words with the 20 highest occurrence values are selected as the 20 key words of that data period. It is important to note that all words are normalised to lower case before any analysis is done, so 'Brexit' and 'brexit' are both treated as equivalent.

5 Findings from the Brexit Dataset

Polling data for the period from 1st Janurary 2016 to 31st December 2019[4] was used for comparison. This poll was taken after Brexit took place, and was chosen because there was more variation in it than in many other topics which have a large dataset of compiled Tweet IDs that are publicly accessible. It is politically relevant to the UK and of particular interest given the importance of social media in the Brexit campaign.

The polling data was split into 3 parts - pro-leave, pro-remain and undecided with each value being the estimated proportion of the population supporting each view. Since we require only one value for the correlation only the pro-leave value was used. This was decided because we are investigating the sentiment towards 'Brexit' so a higher proportion of the population supporting pro-leave should in theory result in a more positive sentiment towards 'Brexit' on X and vice-versa.

5.1 Correlation Between Sentiment and Polls

One of the main aims of this project was to investigate whether there exists a correlation between X Sentiment and Polling data on the topic of Brexit. Unfortunately there was no significant correlation found between the aggregated sentiment data and the polling data. Figure 1 shows the Brexit sentiment data on X in blue and the polling data in magenta for comparison. Analysis was done to test for a correlation between the datasets, giving a correlation value of -0.1980 (the opposite direction to what we would expect) at a significance level of 0.6704 (67.04%), which is not significant. The correlation was repeated by using larger time periods (e.g. 10 days) for the sentiment aggregation. This will smooth the trend and eliminate some noise, but it still did not yield any significant correlation.

This is not a hugely surprising result as there are several significant factors at play that would prevent us from observing a correlation. First and foremost, the unfortunate reality of polling data is that there is a lot of noise, it is not particularly accurate[7], and the underlying population opinion remains relatively stationary with many of the main issues of public discourse, including Brexit.

Secondly, the subset of the population who post on X are unlikely to be the 'swing voters' who will change their opinions. Therefore the sentiment on X will remain mostly stationary since the people posting on X

are going to continue to express the same feelings.

Thirdly, there is a significant shortcoming with the way the sentiment analysis works in this instance - the 'misinterpretation of sentiment' effect. While the sentiment model can reasonably detect whether the general tone of the Tweet is positive or negative, that does not necessarily equate to the Tweet's sentiment towards Brexit. For example, a Tweet that reads "I think Cameron's arguments about Brexit are spot on" would have a positive sentiment score attributed to it even though it actually expresses a negative sentiment about Brexit. Likewise a tweet that reads "Cameron trying to win this Brexit referendum by spreading fear" would have a negative sentiment score attributed to it even though it expresses pro-Brexit sentiment. Dealing with this problem is a much more complex issue, since simple sentiment models are not enough. There are some models that attempt to solve this, and these are referred to as aspect-based sentiment analysis models. This may improve accuracy, but it would still struggle with the above examples as it would not understand whether agreeing with Cameron is for or against Brexit.

5.2 Key Word Extraction for Brexit

The key word extraction method described can give an interesting insight into the direction of the Brexit debate on X. Some interesting keywords from the first few data points are shown in Table 1. Each keyword is followed by the total number of occurrences of the word in the time interval and rate of occurrence per Tweet (a value of 0.01 means the word appears on average once in every 100 Tweets). The entire key word files are published along side this paper.

Most of the common words are relatively unspecific such as 'vote', 'brexit', 'campaign' etc. which is as expected. However, there are more specific, topic related words that come up in different periods that demonstrate what particular things the discussions on X were focusing on. Table 1 demonstrates a very small sample of this. Further research could be done to try to associate these keywords with the topics discussed in the legacy media, to try to establish how similar the the discourse is between X and the legacy media, both on the topic of Brexit and on other issues.

5.3 Sentiment Analysis of Tweets Containing Key Words

It is possible to perform a more fine grained sentiment analysis on Tweets containing a common key word. This could give us an idea of how public sentiment around a specific topic affects public opinion on Brexit as a whole. Figure 2 shows the graph of the average sentiment score for Tweets containing the word 'Cameron' (most of which presumably refer to 'David Cameron'). This data would suffer from most of the same issues as the the overall sentiment analysis of Brexit Tweets described above. However, because the topic is more specific, there may be less of the 'misinterpretation of sentiment' effect.

6 Concluding Remarks

Given the value and price of large quantities of social media data in the current era, there is demand for developing methods to access this data for free. While there is no doubt that tools like this have the ability to be used maliciously, those with malicious intent are the most likely to be able to afford to pay for access to this data. It may not be in the interest of the large social media companies due to the money they make from selling user data, but opening this data up only levels the playing field - a goal which this project aims to contribute to.

In retrospect it is not surprising that no correlation was found between Brexit sentiment on X and the Brexit polling data due to the number of other variables at play and the fundamental failure of the sentiment analyser to specifically determine Brexit opinions. However, understanding this fact is still a useful finding, as it demonstrates limitations of the predictive power of sentiment analysis of X, especially with regards to large and complex topics such as Brexit. This does not render the data useless since there are many other types of analysis that can be done. Key word analysis an example of this - something that is more effective with very large datasets such as this one.

References

[1] Reveilhac, M., Steinmetz, S. & Morselli, D. A systematic literature review of how and whether social media data can complement traditional survey data to study public opinion. Multimed Tools Appl 81, 10107–10142 (2022). https://doi.org/10.1007/s11042-022-12101-0

[2] X Developer Platform, API Pricing Information, available at: https://developer.twitter.com/en/products/twitter-api (25/06/2023)

[3] Calisir, Emre; Brambilla, Marco, 2020, "Twitter dataset about Brexit", https://doi.org/10.7910/DVN/KP4XRP, Harvard Dataverse, V1, UNF:6:5wutKhAU/2JSByEfCbI2Tg== [fileUNF]

[4] What UK Thinks, Should the United Kingdom remain a member of the European Union, or leave the European Union? (Asked after the referendum). https://www.whatukthinks.org/eu/questions/should-the-united-kingdom-remain-a-member-of-the-european-union-or-leave-the-european-union-asked-after-the-referendum/

[5] Hugging Face, DistilBERT Sentiment Model, available at: https://huggingface.co/docs/transformers/model_doc/distilbert (20/06/2023)

[6] Biswas, S.K., 2019. Keyword extraction from tweets using weighted graph. In Cognitive Informatics and Soft Computing: Proceeding of CISC 2017 (pp. 475-483). Springer Singapore.

[7] Prosser, C., & Mellon, J. (2018). The Twilight of the Polls? A Review of Trends in Polling Accuracy and the Causes of Polling Misses. Government and Opposition, 53(4), 757-790. doi:10.1017/gov.2018.7

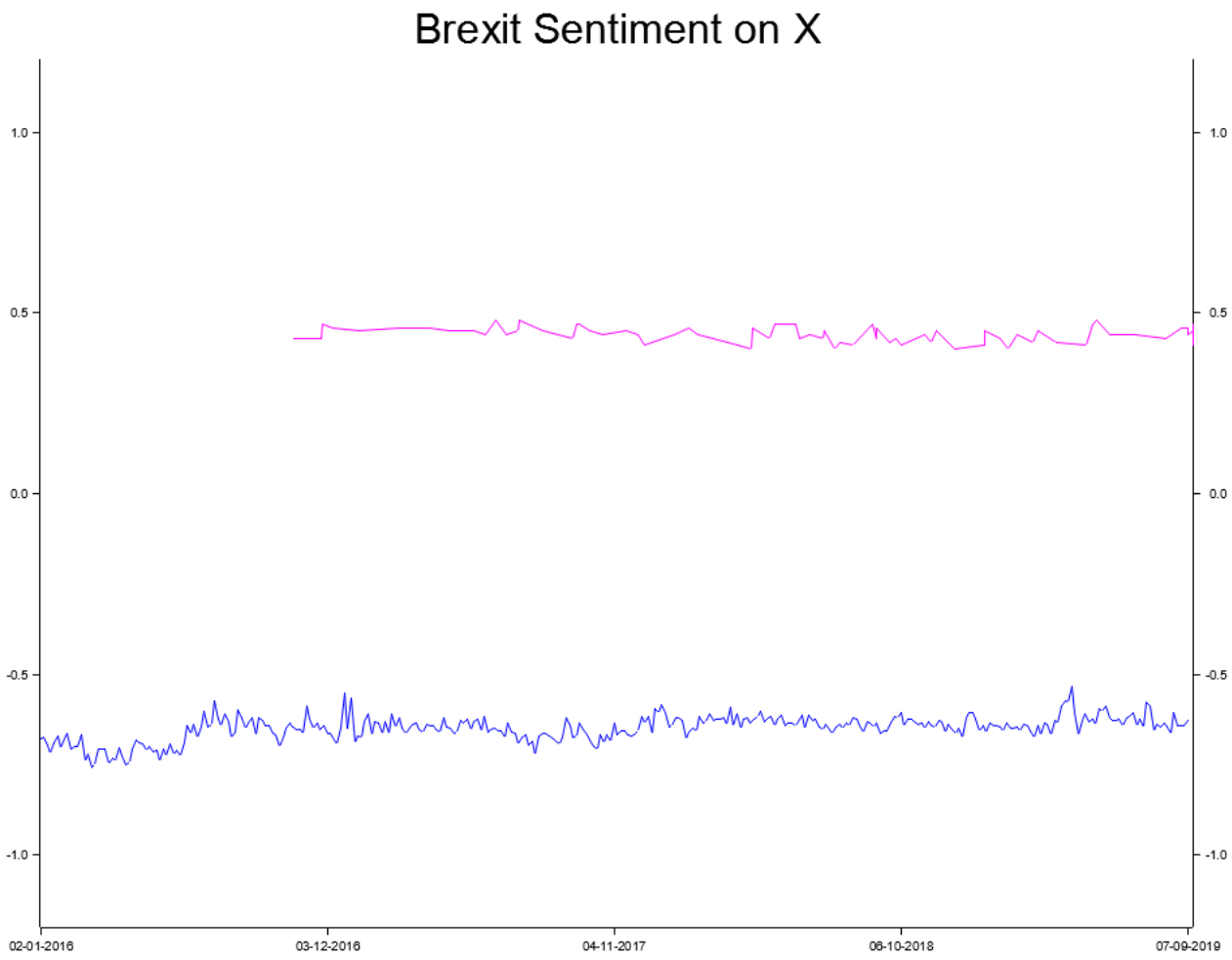

Figure 1: Sentiment vs Polling Graph

01/01/2016	`poland` (260, 0.1233), `nato` (210, 0.0998)
09/01/2016	`cameron` (372, 0.1086), `toyota` (198, 0.0578)
09/03/2016	`queen` (2926, 0.1345), `science` (779, 0.0358)
12/05/2016	`imf` (2301, 0.0703), `recession` (1193, 0.0364)
28/08/2016	`german` (1068, 0.0297), `chequers` (741, 0.0206)
16/02/2017	`blair` (5905, 0.1620), `lords` (1065, 0.0292)
14/10/2017	`deal` (4101, 0.1043), `hard` (1145, 0.0291)
13/10/2018	`ireland` (2362, 0.0357), `border` (2256, 0.0341)

Table 1: Interesting examples of key words

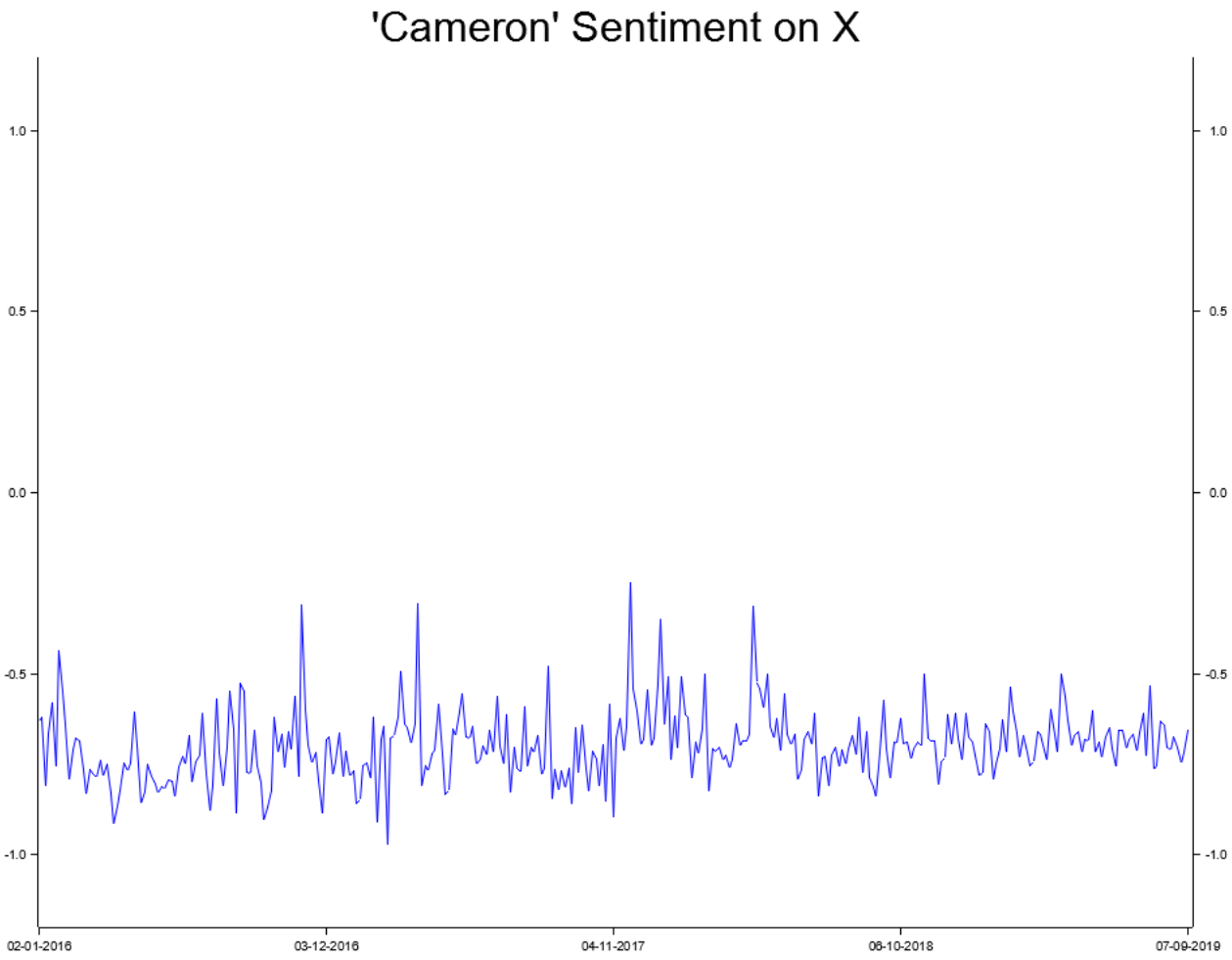

Figure 2: 'Cameron' Sentiment Graph