Media Conflict and DEMocratisation

Walid Al-Saqaf

Mecodify: A tool for big data analysis & visualisation with Twitter as a case study

November 2016

Media, Conflict and Democratisation (MeCoDEM)

ISSN 2057-4002

Communications, power and governance in democratisation conflicts

Copyright for this issue: ©2016 Walid Al-Saqaf

WP Coordination: Stockholm University / Christian Christensen

Editor: Katy Parry

Editorial assistance: Charlotte Elliott-Harvey

University of Leeds, United Kingdom 2016

All MeCoDEM Working Papers are available online and free of charge at www.mecodem.eu

For further information please contact Katrin Voltmer: k.voltmer@leeds.ac.uk

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 613370. Project Term: 1.2.2014 – 31.1.2017.

Affiliation of the authors:

Walid Al-Saqaf
Stockholm University
walid@al-saqaf.se

Table of contents

Abstract 1

1. Aim and motivation 2

2. Data structure and storage 3

3. Data visualisation 9

4. Limitations 12

Bibliography 15

Abstract

This paper describes *Mecodify*, an open-source tool aimed at codifying and presenting online data primarily for the use of social media researchers. Although the first stage of the platform’s development is confined to Twitter as a source of data, the tool has the capacity to develop so it could use data from other platforms including but not limited to Facebook, Instagram, and even news websites. This essay identifies potentials of growth and development as well as limitations of the tool.

1. Aim and motivation

Mecodify emerged as a tool that enables the extraction, codification, analysis and presentation of Twitter data to be analysed by researchers within the Media Conflict and Democratization (MeCoDEM) project,[1] which aims at investigating the role of traditional media and ICTs during and after times of conflict and assessing whether and how they were used to promote certain campaigns, trigger social change or/and lead to political action. It is therefore no coincidence that *Mecodify* uses the first five letters of MeCoDEM in its name as a tribute to the project that started it.

Unlike traditional media, studying the use of ICTs necessitates dealing with big data stored on external servers. Doing so requires technical skills to automate the data-gathering process, save the extracted data, do extensive analysis and provide means of visualisation. While some may argue that ICTs, with the exception of software like NVivo, are generally not needed for qualitative research (Roberts and Wilson, 2002), they are rather vital for quantitative studies that involve big data gathered from social media like Twitter and Facebook.

To address the need to objectively study ICT use during and after conflict, *Mecodify* was initially used internally within the MeCoDEM group to analyse Twitter data. It was also intended to be one of the project's outputs to be published as an open-source project for the possible use by others in the future. Twitter was chosen as the first source of data for *Mecodify* because almost all of its content is publicly accessible and is composed of short elements of data called tweets,[2] making it the preferred platform for posting quick and short public messages. Unlike Facebook, the search function of Twitter allows it to have many research applications (Ovadia, 2009). It also has a well-documented set of application interfaces (APIs) that facilitate the automation of data extraction. One can then conclude that media and communication scholars should seize the opportunity by studying Twitter's use, especially as more media and users around the world are increasingly using it for disseminating and receiving information (Deller, 2011).

The creation of *Mecodify* was motivated by the following:

[1] See http://mecodem.eu

[2] A tweet is a 140-character message that anyone with a twitter account could post publicly.

1) The need to provide a simple and effective way to explore, analyse and visualize data obtained from social media on various topics. For example, it could help answer questions on the use of Twitter in connection to particular incidents or conflicts.

2) Taking advantage of available skills within MeCoDEM instead of having to recruit experts or software developers from outside the project.

3) The value of complementing the theoretical approaches of the project by using a practical tool that could strengthen the project's impact beyond its lifetime. Through *Mecodify*, MeCoDEM could engage researchers, members of civil society, journalists and policy makers who wish to tap into its database well after the project ends.

4) The opportunity to enhance team interaction and collaboration and allow different researchers to use the tool to strengthen their outputs and share ideas and experiences on the analysis of big data.

5) Making use of the vast open and structured data that Twitter provides, particularly as social media research has become an important area of study due to the rapid expansion of their user base.

While it would have been possible to purchase an account in one of the many commercial Twitter data analysis platforms, creating a new tool provides a higher degree of transparency given that all tasks from data gathering, to analysis and visualisation are done in-house. Furthermore, relying on a company with proprietary software is more expensive and limits the prospects of using the data beyond the lifetime of the project. Using *Mecodify* helps enhance the learning experience of its users when dealing with complex data gathering, storage and visualisation processes, which is a strong educational benefit for the teams involved. Finally, *Mecodify* has the potential to evolve and develop to accommodate new features and use data from other online sources. It is also scalable and adaptive, allowing it to become a valuable tool for use in other future projects, which would be a positive contribution of MeCoDEM's to the research community.

2. Data structure and storage

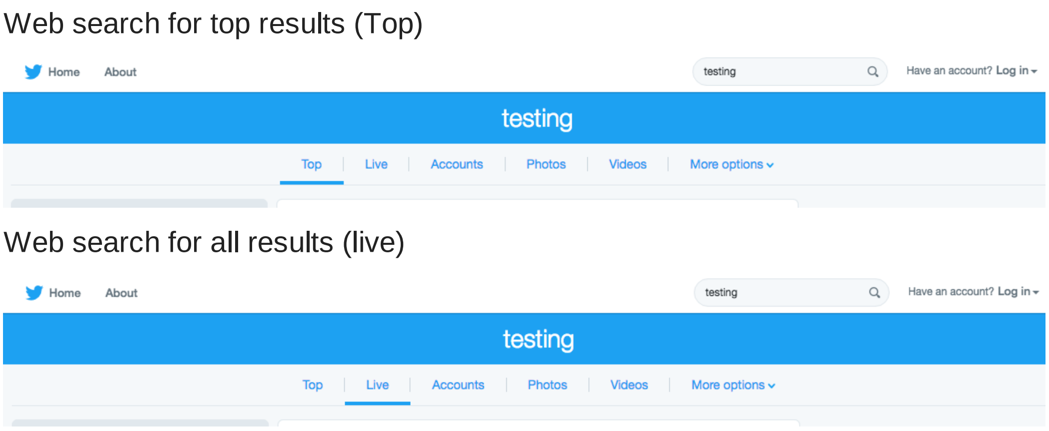
Given that that *Mecodify* is, at least for the first stage, using Twitter as the source of its data, it is important to understand what types of data are offered by Twitter to create the appropriate code book. But before doing so, it is important to identify the parameters necessary for the Twitter search procedure. In essence, *Mecodify* allows the creation of a *case*, which requires the following information:

1) ID (an alphanumeric name to be assigned to the case)
2) *Name* (an easy to understand short title)
3) *Search query (including operators such as AND, OR, etc.)*

Additionally, there are some optional parameters that could also be passed:

1) *Search method (Search API or Web Search)*
2) *Search criteria (Overview or full results)*
3) *Details (to include more information about the case if needed)*
4) *URL (any external link that could provide more information about the case if any)*
5) *Privacy (Private or Public)*

In the case of Twitter, the *search method* could either be the Search API or Web Search (https://Twitter.com/search). The Search API utilizes the Twitter Search API and is relatively fast and quite efficient. The Web Search method, however, is slow and requires a two-step process which involves crawling the website for the initial results based on a regular search query and then using the retrieved tweet IDs as input to a Twitter API to retrieve the full details of the tweets. While cumbersome, inefficient and time-consuming, the web search method allows searching tweets older than seven days, which is the limit that the Search API is confined to. The search *criteria* can be *top results only* mode, which are retrieved based on Twitter's algorithm that is used when selecting *top* tweets that Twitter claims to represent popular tweets based on the search query. If the 'top results only' option is not chosen, all search results that come up using the *live* mode of Twitter will be displayed as shown in Figure 1. The *details* and *URL* are mainly to provide readers with some context on the case. Finally, the *privacy* settings indicates whether only the creator of the case is able to view the results or that all members of a predefined group are also able to view it.


Web search for top results (Top)

Web search for all results (live)

Figure 1: The search method options when using *Mecodify* to create Twitter cases

Once the above parameters are set, search queries using Twitter's standard search are composed. For the web search method, the date range is split to individual days so that the search would occur for each day separately and sequentially from the last to the first. For example, if the period is from 20-Feb-2015 to 25-Feb-2015, individual search queries would then be carried out for the dates: 25, 24, 23, 22, 21, 20 Feb in the given order.

A crawler script written in PHP will then be triggered to fetch the first page of results by accessing the standard Twitter search page that starts with https://twitter.com/search?q. Once the first batch of results is obtained, the script then checks to see if there are further results also available. If so, it would fetch each of the pages by accessing the address embedded in the returned results, which usually start with https://twitter.com/i/search/timeline. Each call will be made with a delay of one second in between successive HTTP GET requests.

Each returned page of results contains permanent links to the returned tweets. Those links are identical to the ones a user would get when clicking on the 'More' link beneath any tweet and then selecting 'Copy Link to Tweet'. The link is then saved to memory for the extraction of the *tweet id*, which is a unique long number as indicated in Twitter's API documentation (see: https://dev.Twitter.com/overview/documentation). Those IDs are then stored in the database and a text file log associated with the query is saved to the server to mark the IDs that were already stored so that if a process is interrupted for any reason, the search would resume from where it stopped.

Once a batch of pre-defined number (default is 100) of tweet IDs is obtained, *Mecodify* starts using the official Twitter API, which allows fetching the complete information for each of the obtained tweet IDs. The open-source PHP library TwitterAPIExchange is used for that process (https://github.com/J7mbo/Twitter-api-php). The script then iteratively reads the database to extract the tweets. Tweet IDs are used as parameters to access Twitter's database through its *GET statuses/lookup* API to fetch the rest of the data related to the particular tweet. Each record includes a special variable called *tweeter id* representing a *tweeter*, which corresponds to the user that posted that particular tweet. The tweeter id is then used to create a new record in a separate dataset for tweeters. The two datasets interact with each other through a relational query using the common tweeter id variable as illustrated in Figure 2.

While Figure 2 shows eight variables in addition to the key unique *tweet id* and tweeter id, there are many other variables that are fetched through the API[3] but which are not mentioned in this essay because they are not yet fully utilized by *Mecodify*.

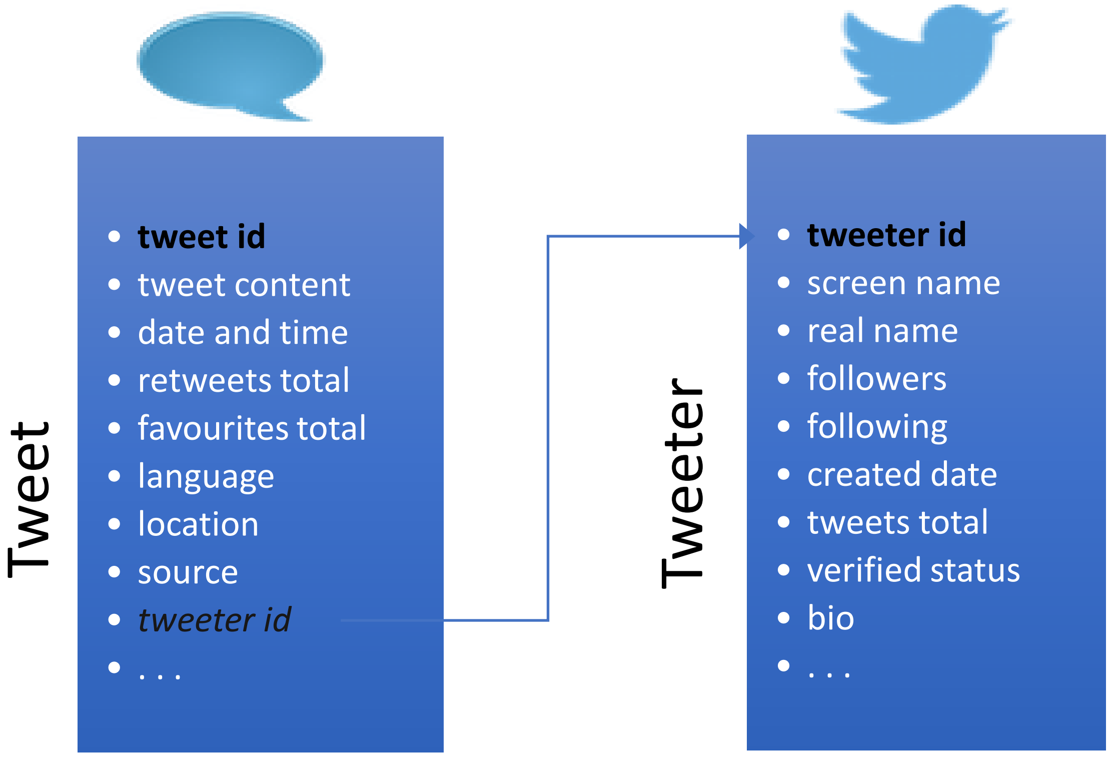


Figure 2: The two datasets acquired from searching Twitter

Below is a more detailed explanation of the main variables and their value from a research standpoint along with their variable type.

Tweet variables

- ***Tweet id:*** This is a unique number used to fetch elements of an individual tweet. *(number, nominal)*
- ***Tweet content:*** This includes the text and associated links or media in the body of the tweet. It is possible to derive specific data from the text including hashtags, mentions of other tweeters, links to external pages, etc. It holds no other metadata apart from that. This variable is crucial for content analysis as it helps understand the sentiment, tone, focus, and other attributes of the actual message. *(string, nominal)*
- ***Date and time:*** This is the timestamp indicating when the tweet was originally posted. It is in GMT/UTC time. *(datetime, ordinal)*

[3] The complete specifications of the data structure are available at: https://dev.twitter.com/overview/api/tweets

- ***Retweets total***: The number of times the tweet was retweeted by others. It helps identify how popular and resonating the tweet was. *(integer, scale)*
- ***Favourites total:*** The number of times the tweet was marked as *favourited* or *liked* by others. While this is less common than retweets as a metric, it is often the case that it correlates positively with the number of retweets. *(integer, scale)*
- ***Language:*** This is the language that Twitter assigns to the tweet based on its own algorithm. On certain occasions, this value may be incorrect due erroneous machine language detection. It may be useful to use this value to filter out noise or irrelevant tweets that may use the same hashtag in another country. *(string, nominal)*
- ***Location:*** This is an optional field that users could enable to track where they are by latitude and longitude values. The majority of users have this geo-location tracking option disabled. This makes the use of this variable rather limited. However, each tweeter also has the ability to set his/her location on his/her profile page as well. *(string, nominal)*
- ***Source:*** The platform used to post on Twitter. It is useful to know if the tweeter used a mobile device or a computer. It is useful to verify if a person tweeting near an area where a conflict is taking place was indeed on the move when he/she tweeted. It could also help examine the relationship between engagement/influence and the device used. For example, would people who have iPhones be more influential than those using Android devices. *(string, nominal)*
- ***Tweeter id:*** This is a number pointing to the person who posted the tweet. It is essential to include as it is used to find the particular tweeter record using Twitter's APIs. *(integer, nominal)*

Tweeters

- ***Tweeter id:*** This is a unique number used to fetch elements of an individual tweeter. *(integer, nominal)*
- ***Screen name:*** This is another unique identifier that could be up to 15 characters long and sometimes includes the name of the tweeter. It often starts with @ when it is referred to in tweets. *(string, nominal)*
- ***Real name:*** The real name of the tweeter as provided by him/her when setting up the account. Using this variable, it is possible to identify the role, affiliation and other details of the tweeter that would then be helpful in connecting his/her action on Twitter to the real world. *(string, nominal)*
- ***Verified status:*** Twitter introduced this status in 2009 as a way to limit identity theft and abuse of its service. Most celebrities have their Twitter account verified. It would be useful from a research standpoint to examine whether there is an association

between the level of engagement and popularity of posts from verified account vis-à-vis non-verified accounts. *(boolean, ordinal)*

- ***Following***: The number of Twitter users that the tweeter follows. This variable helps show how eager a user is to keep track of tweets by others. *(integer, scale)*
- ***Followers:*** The total number of Twitter users following the tweeter. This is often an indicator of popularity and prominence on Twitter. Most high-profile individuals and media accounts for example have a high number of followers. *(integer, scale)*
- ***Created date:*** The date the account was created. *(date, ordinal)*
- ***Tweets total:*** The total number of tweets the tweeter has posted on Twitter since the account was created. It is a good way to assess the level of activity and time spent on Twitter. *(integer, scale)*
- ***Bio:*** Information provided by the tweeter about themselves. It includes links, images or other metadata useful to know about the background of the tweeter. While it is possible to use keyword searches to know if the tweeter belongs to a particular political party, or has other affiliations, it is not always reliable to use text detection to draw conclusions from this field. Therefore, manual research may be required. *(string, nominal)*

There are also a few additional variables that are not described in this paper for the sake of space. The datasets holding the values of those variables are stored in a MySQL database that is backed up daily. Adding a new case with its own search query is relatively easy and straightforward.

A note about replies and mentions

There are two very similar types of interaction on Twitter, replies and mentions. It is important to note here that while replies are directly connected to the button clicked by a tweeter in response to a particular tweet or user, mentions do not require having a tweet at all. It is possible for a tweeter to mention a particular screen name even if that name has never contributed to the dataset with any tweet. Furthermore, a tweeter could mention multiple accounts but a reply is different in that Twitter embeds the response action taken by the tweeter into a hidden set of data that are not shown publicly but can only be identified when using the API. It is therefore possible that a reply could have no mention of the person being replied to. When collecting or analyzing a given data set, the user should thus be aware that where a certain account name is used as part of the search criteria (e.g. as a keyword), replies that do not mention the account name of the person being replied to would not be included in the data set.

Given its comprehensive filters and flexibility in accepting a wide range of search queries, it is possible to optimize the query in question so as to maximise the effectiveness of the search when seeking tweets connected to a particular tweeter. For example, if one seeks to get all the tweets related to a particular person with the Twitter screen name @user, last name LAST and first name FIRST, it may be useful to have a search query such as (from:@user OR @user OR "FIRST LAST"). The first part from:@user will capture tweets published by the person, @user would capture other tweets that mention him/her and "FIRST LAST" would capture the mention of the person even without referring to the user handle. If it is a prominent person, it might be enough to use FIRST OR LAST instead of "FIRST LAST".

3. Data visualisation

Mecodify had to be of use so that a user could examine and go through the data without the need for special technical skills. To achieve this, scripts utilizing HTML, Javascript, and PHP were constructed to link the database to a web-based interface that takes advantage of the HTML5 capabilities to draw native graphs. Open-source HighCharts and D3JS.org open-source charting libraries were used to render the graphics needed for data visualisation.

As Figure 3 shows, *Mecodify* has a left column where one can select the case and various parameters to display the graph with a timeline and number of tweets, retweets, and unique tweeters. Below the graph, there is a hashtag cloud in addition to the actual list of the tweets with all the necessary information provided in a tabulated sortable format.

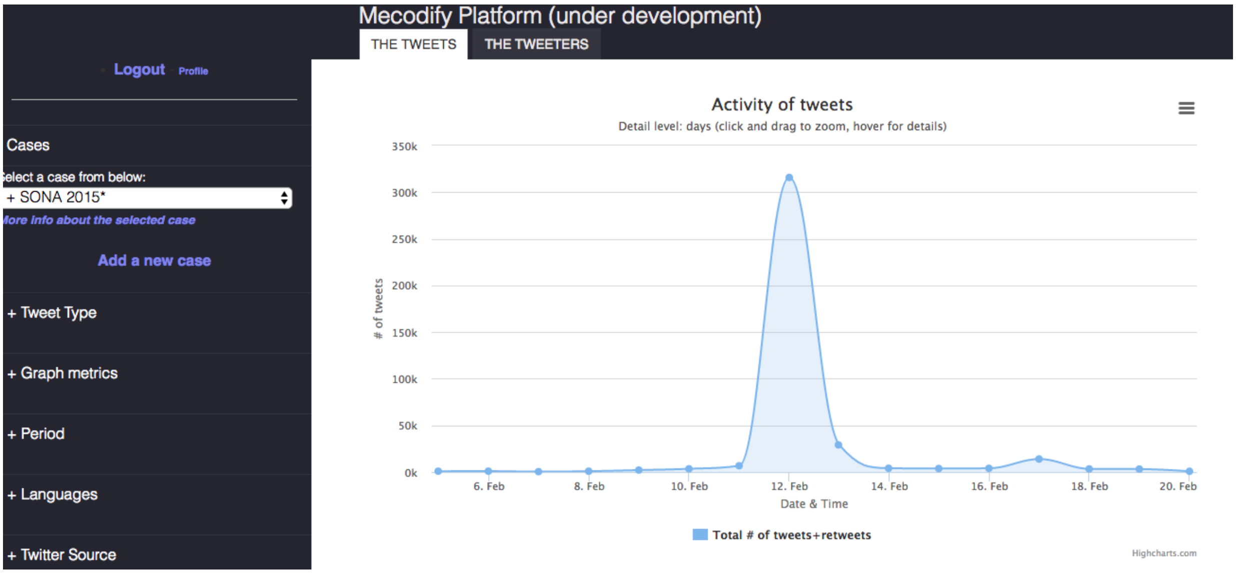


Figure 3: *Mecodify* being used within the MeCoDEM project

For Twitter data, *Mecodify* has the ability to produce two sets of data both as visualisations and raw comma separated value (CSV) files. The first is concerned with *tweets* and provides a timeline with a line graph illustrating the number of tweets and/or retweets of the whole dataset or a subset that can be defined using an extensive list of filters. The generated graph could be downloaded and embedded in an image, vector or PDF format.

The web interface allows access to two independent tabs. The first shows data on *tweets* contents over time and is useful to examine time-sensitive developments on Twitter. The second tab deals exclusively with *tweeters* by providing insight into their level of participation and interaction as well as the connections between them in terms of how they mention or retweet each other. Unlike the first tab, which is about the contents of the messages, the second tab deals with the senders of those messages and is more useful for the understanding of the social networking between individuals regardless of the timeline.

Mecodify currently only visualises Twitter data but may well be developed in the future to use data from other platforms such as Facebook or YouTube. The presentation of data in visual and standard textual format enables their integration into other spreadsheet and word processing applications. Furthermore, the CSV files could be imported into advanced software such as Tableau or Gephi for more refined processing.

The data being visualised

Mecodify allows the plotting a line graphs on a timeline axis showing the number of tweets or retweets representing a particular case. Once plotted, a hashtag cloud is displayed showing the most widely used hashtags during that period. Additionally, the list of most retweeted tweets is displayed in a table below the cloud. If the results yield more than 100 tweets, the table is displayed in pagination with 100 tweets per page. The table includes various fields such as whether the tweeter has a verified account, any images used, the actual text of the tweet, and information about the tweeter. It is possible to sort by any of the column values.

Once a case is created and enough data is gathered, it is possible to plot the graph with the default filter values by clicking on the *Visualise* button even if the data extraction process is still continuing. The filters allow having a more rigorous examination of the data. Below is a brief description of the different fields that the user can adjust as required.

Tweet Type

This filter allows limiting the displayed data to tweets that have specific keywords, images, links, or are from specific accounts, are tweeted from specific locations, include specific hashtags, etc. The variety of options is wide. The filter is simple enough to use by checking on checkboxes that define the content the user wants to see. However, if needed, a more complex query system is possible to implement at a later stage through the integration of boolean syntax similar to the one used by Google.

Retweets

It is possible to limit the graph to only original tweets. This is useful when one wants to measure the overall level of activity. However, when introducing retweets to the equation, one can also know the level of popularity of certain tweets, adding one more dimension. It is also possible to plot both values on the same graph by checking on the *overlay* checkbox that allows plotting both with different colours.

Period

In addition to the possibility of zooming into a graph dynamically by clicking and dragging along the x axis, the period option allows zooming even more to a particular period defined in terms of days, hours, minutes and even seconds. This filter becomes quite powerful when trying to scrutinize the detailed developments immediately after a particular attack or explosion for instance. It also makes it possible to know who started a particular hashtag.

Languages

Occasionally, it may be useful to only limit the data to tweets in a particular language. This may be important when trying to know, for instance, if minorities had contributed to a discussion or to filter out irrelevant tweets that seem to be using the same hashtag in a different country. It is possible to add several languages through the open text input box that accepts standard ISO two-letter language codes.

Twitter Source

Using this filter, it is possible to see when tweets were posted using mobile devices or particular clients. It is in fact possible to limit the data to tweets posted using particular apps that are installed on particular devices such as Android-based phones. This becomes handy when doing cross-country analysis as well as when trying to figure out the connection between the device used and the level of engagement and popularity on Twitter.

Additional options

There are three additional options that the user could set. One is to show or hide the tweet tables that appear beneath the graph by default. The other does the same but for the hashtag cloud. The third option allows stacking graphs on top of each other on the same page, which makes it possible to compare two or more graphs without having to open them on different tabs. There is certainly more room to add additional filters based on the demand of the community of users.

4. Limitations

Data source limitations

The biggest limitation for the first version of *Mecodify* is the lack of access to historical Twitter data through an API. Twitter had initially offered to provide free access of its full archived data to the research community years ago (Finley, 2014), but the door had since been closed[4]. To overcome this limitation, an approach of using a script to crawl web search results on twitter.com was found to be the remaining option to retrieve the tweet IDs and then using a Twitter API to get the details of each. This presented some technical issues and ethical questions as indicated in the next section. Apart from being lengthy and resource-hungry, the web search method is less reliable because it depends on what is shown to the public via the web interface, which is different than what is revealed through the API although the differences are not apparent and not properly defined by Twitter. The limitations below specify those that we are aware of, based on Twitter's documentation and previous research.

Twitter Search API limitations

Searching using the standard Twitter Search API has a number of limitations of which perhaps the most important is the restriction to tweets posted in the last seven days. While this may be sufficient when studying recent developments, it is not of use for studies that require historical data. Another technical limitation is the need for API keys to access the Search API, which means that there is a limit as to how many persons use the tool on the same platform at the same time. Furthermore, the API has a strict rate limitation as it can only allow a maximum of

[4] See: https://engineering.twitter.com/research/data-grants-closed

180 queries per 15 minutes. *Mecodify* overcomes this problem by enabling the entry of multiple access tokens so that if one times out, it could use the other and so on.

Additionally, the search API focuses on relevance and not completeness, leading to the possibility of missing some tweets or tweeters that Twitter's proprietary algorithm sees as irrelevant[5]. Twitter also stresses that there are no guarantees that the search API would yield the exact same results that are obtained using the web-based search. To what extent a given set of search results is complete and how Twitter defines relevance is thus unknown.

Web search limitations

Unlike the Search API method, the web search relies on using the web interface (twitter.com/search) by plugging in the search data parameters and getting the results using a slower and less efficient approach that requires parsing through the returned HTML files. This causes slower performance but allows going beyond the seven-day limit of the Search API. Additionally, the web search method does not include retweets as individual entries in the returned results. It only allows identifying the total number of retweets of original tweets. Hence, it is not possible to use the web search method to track how a particular tweet got retweeted over time and who contributed to its popularity. This can, however, be done using the Search API.

Unlike the search API, the web search method does not have a rate limitation but the robot.txt file indicates that crawlers need to give a one second pause between each automated request. Given the need to download and parse large HTML files, the web search method requires greater hardware resources in terms of bandwidth, storage and processing power. To reduce the use of those resources, the search query covering a particular period of time is divided to separate search queries for each day in that period. However, for some very popular subjects, the number of tweets for a single day may well be in the hundreds of thousands. In such a situation, the web search would have to run for a long period of time, perhaps for days on end, which could strain some servers with limited capacity.

Finally, unlike the Search API, which has a tracking mechanism based on the tweet ID, the web search method requires restarting the search for that particular day if the crawling process was interrupted for any reason. However, it does have a mechanism to track the completed days and resume from the day that was interrupted, which helps reduce the burden on the server.

[5] Details available in the API documentation: https://dev.twitter.com/rest/public/search

The web search approach is believed to obtain all tweets resulting from a given query, with only the limitation that retweets will be missing. Therefore, a search that results in a smaller number of tweets is likely to obtain a complete data set where queries with very high numbers of tweets in a short time period may not return a complete data set.

Incomplete responses

It is not always possible to see full conversations between a particular tweeter and those who respond to him/her because for any reply to be included in the dataset, it has to meet the search query's criteria. For example, if a query requires having the hashtag #SONA, a response to a tweet that meets the criteria will only be included in the data extraction if it too has that hashtag. So a response like "I support your view on this issue" will not be recorded while "I support your view on the issue of #SONA" will be included. There is no way for *Mecodify* to collect a tweet and all the replies it received regardless whether they meet the criteria or not.

Inability to track dynamic changes

Finally, dynamic changes such as deletions of tweets, changes of screen handles, profile images, bio information, etc. are not possible to track in real time. Once they are indexed in the database, they remain fixed indefinitely until a new update is triggered manually. One of the implication of this limitation is the inability to know for certain the number of times a particular tweeter was mentioned in other tweets when that tweeter changes his/her user screen name. It was evident that Twitter does not change mentions in older tweets, which may occasionally constitute a reliability problem. For example, when retrieving a list of tweets having the #SONA2015 hashtag, it was found that many tweets mentioned the tweeter @EconFreedomSA. However, looking at the tweeters list, that tweeter seems to have changed the user screen name to @EFFSouthAfrica, which meant that a measure of the number of mentions for it would result in zero, while that tweeter was mentioned extensively under a different name. As noted earlier, a mention is simply the case when a particular tweet text includes the screen handle of another tweeter. If the mentioned tweeter changes his/her screen handle, the mention does not change with it. So far, there is no reliable way to solve this problem because it is related to how Twitter works and the fact that it does not seem to have a method to track changes that have occurred over a particular period of time. While a reply to a particular user or tweet is fixed and permanent and it is possible to track replies even if the tweeter changed his/her screen name.

Implications of web search limitations

The inability to display retweets using the web search method highlights lack of clarity from Twitter particularly as there is a checkbox in the advanced search page which if checked, should supposedly allow the display of retweets but it was found that checking it or not did not make any difference. The only thing one can know from the automated web search results is the total number of retweets. However, it is still possible to know the most recent 100 retweets of a particular tweet but that would require extensive resources because each tweet would need to be run individually by calling the GET Statuses/Retweets API. The extent of time and effort required to do this for hundreds of thousands of tweets is too costly for this stage of *Mecodify*'s development. Furthermore, there is simply no way to know the full list of all the persons who retweeted a particular tweet if the total was more than one hundred retweets. This means that the ability to track from start to finish how a particular tweet propagated or grew using retweets is very difficult.

It is noteworthy that *Mecodify* obtains edited retweets, i.e., retweets that include a comment or change to the original tweet, which Twitter labels as *quote tweets*. However, they are very few in number compared to regular retweets, which are merely replicas of the original except that they are triggered by different tweeters. It is possible to identify quote tweets using the 'Quotes another tweet' filter under 'Tweet Type' in *Mecodify*.

While Twitter's Terms of Service (https://twitter.com/tos) do not prevent robots from crawling the website, they do prohibit 'scraping of Services'. *Mecodify* crawls Twitter's web source page to identify and extract tweet IDs that emerge from web searches[6], which may or may not be seen as 'scraping' depending on the definition of the term. Questions sent to Twitter on this matter went unanswered.

Bibliography

Deller, R. 2011. Twittering on: Audience research and participation using Twitter. *Participations.* **8**(1), pp.216-245.

Finley, K. 2014. Twitter opens its enormous archives to data-hungry academics. June 2, 2014 at 16:41. *Wired.com - Business.* [Online].

Ovadia, S. 2009. Exploring the Potential of Twitter as a Research Tool. *Behavioral & Social Sciences Librarian.* **28**(4), pp.202-205.

Roberts, K.A. and Wilson, R.W. 2002. ICT and the Research Process: Issues Around the Compatibility of Technology with Qualitative Data Analysis. *2002.* **3**(2).

[6] This is an approach similar to the one taken by other academics such as Tom Dickinson. See: http://tomkdickinson.co.uk/2015/01/scraping-tweets-directly-from-twitters-search-page-part-1/