SMART-Portal: A data tracking tool for research purposes from social media and news websites

Stefan Stieglitz
University of Duisburg-Essen
stefan.stieglitz@uni-due.de

Ali Sercan Basyurt
University of Duisburg-Essen
ali-sercan.basyurt@uni-due.de

Milad Mirbabaie
Paderborn University
milad.mirbabaie@uni-paderborn.de

Abstract

Social media data has become an important source of information for different parties including researchers from various research fields. Rarely is information given regarding the methods used to acquire social media data to a satisfying degree that would allow the reproduction of research methods and results. With the goal of providing researchers with a method to collect social media data as easily as possible, this paper describes the development of a scientific tool to acquire such data and to use as a method to conduct future research which can be referenced in their work. This tool is named SMART-Portal and it provides researchers with the ability to track data online. Furthermore, it allows the exportation of data in multiple formats for different analyses in addition to allowing researchers to track data from different predefined news websites and Twitter.

1. Introduction

Social media has become an important tool for various parties from different domains, such as businesses [1], [2], crisis managers [2], [3], or politicians [4], [5]. It offers them opportunities to share information, beliefs and opinions, promote products, or the ability to engage their target group directly. By simply registering to such websites, and sharing content in form of images, videos, or text messages a massive amount of data and information is created and made available to others [6]. This data can be sorted into two categories [7]. The first category is focused on quantitative data meaning it includes objective facts and numbers such as the number of followers of a Twitter account or the number of retweets a message on Twitter has. This category is titled structured data. In contrast, the remaining category is named unstructured data and it concerned with qualitative data that needs to be interpreted for example textual data such as comments or a tweet.

The term big data is used in data science to refer to very large datasets containing a massive amount of relevant information. They are frequently used in various research disciplines to study a multitude of topics such as human behavior. In this regard, big data collected over social media is named as a significant contributor to the analysis of human behavior [8]. Such data has been labeled as social media big data and it has become a possibility through the increasing popularity and the significant growth of social media platforms [6].

The rise of social media platforms and with it the increase in available data provided by individuals and organizations led to the need for a discipline that is able to handle such data as competently as possible. This challenge led to the emergence of the field of “Social Media Analytics.” It is focused on the analysis of social media data by creating and modifying existing methods with the purpose of exploring data collected from social media platforms such as Twitter or Instagram [9].

Besides the value that social media data has for a multitude of different research disciplines, organizations and individuals it is of utmost importance to acquire the data by utilizing methods and technologies which are aimed at data collection from social media platforms. Twitter provides researchers with a valuable method to collect social media. The advantage of collecting data from Twitter for researchers as opposed to accessing data for example from Facebook is that it allows researchers to collect data from all registered accounts that are not blocked on the platform [11]. After the Cambridge Analytica scandal in 2018, the changes to Facebooks guidelines require researchers and developers to obtain different permissions from Facebook to access specific data from their platform.

As valuable as social media data is for researchers a big problem with research that has been conducted using data from Twitter or similar sources can be identified. Often researchers use different ways to acquire datasets or do not precisely report how they were collected [12]. Moreover, researchers from non-IT related disciplines often lack skills to collect data from social media. Additionally, to our knowledge a tool that could be leveraged by the whole research community for data collection from sources such as Twitter and news websites does not exist. These matters make it difficult to acquire or reproduce the data analyses that were performed and their results.

With this issue in mind, the tool “SMART-Portal” was implemented. The term “SMART” is an acronym for “Social Media Analytics Reporting Tool” which is used in the field of social media analytics. The developed tool addresses the matters mentioned above by providing its users, which are exclusively researchers with a web application where they can define particular search terms with a specific period of time to track these search terms as individual projects. These terms then are tracked in content sent as tweets on Twitter, retrieved, and stored in a database. The data stored includes information such as usernames, comment tweets for each tweet, and the timestamp at which a specific tweet was sent. Afterward, the researchers are able to export the collected data in multiple data formats. The goal of this tool is to make the process of tracking certain situations through search terms on Twitter and other sources easier for researchers. This tool is currently dedicated to retrieving data from Twitter. However, it is also able to retrieve articles and comments from different news websites. It is aimed at providing an approach to data collection that is easy to comprehend, easy to reference in future projects in addition to providing a look into the collection of data itself. Existing data collection tools such as Twi-FFN [13] and Storywrangler [14] are focused on providing researchers with specific data such as a list of followers, trending topics or the daily usage of specific terms while the SMART-Portal provides researchers with a dataset of Tweets, Tweets information such as retweets and likes in addition to user information for example their follower count.

Therefore, we intend to answer the following research question with our work:

RQ: How can a scientific tool that helps researchers to collect data long term from multiple sources based on predefined parameters be developed?

The remainder of our paper is structured as follows. First, a literature review for social media analytics and tools in that field is given. Then the design goals that were followed while designing and implementing the SMART-Portal are presented. Next, the system architecture of the presented tool is described which is followed by an explanation of the user interface of the SMART-Portal. Afterward, the Twitter-Fetcher and then the RSS-Fetcher component of the tool is introduced. We conclude our paper by depicting the novelty of our tool for researchers as well as describing work that will be performed on the tool in the future.

2. Literature Review

With the increase in popularity of social media platforms more data is available that can be analyzed in order to gain valuable information. The data provided by users of these platforms is therefore very valuable for different parties such as businesses which in return leads to an increase in demand to collect and analyze this type of content [15]. This demand for data collection, analysis, and insight led to the creation of the discipline of social media analytics. It is described as a methodology that is aimed at collecting data from social media networks followed by analyzing the online activities of users which includes their sentiment and the data they generate online with the purpose of interpreting the data to gain insights that are otherwise not accessible[16], [17]. The main reason to do so and for social media analytics itself is to obtain information, the opinion of users on certain topics and to analyze data through the tool with the motive of then making improved and educated decisions[15]. For this purpose, this interdisciplinary field combines a variety of techniques such as machine learning, statistical methods, and web scraping which allows the extraction of data from websites [18]. The insights derived from the data for businesses can be for example the identification of particular parts of specific products with which customers are either satisfied or unsatisfied [19] or it can provide information that can be used to reveal opinion leaders or influencers. They can for instance be identified by performing social network analysis, sentiment analysis, and by examining their network of followers [20]–[22]. The influencers could then be utilized by companies for the purpose of advertising their own products[23]. Other use cases for such data are the measurement of a company’s reputation[24], the detection of topics in social media conversations [18], [25] or to perform spatial analysis of data with geolocation information[18]. This increase in interest in social media analytics led to an upsurge in the development of tools that are capable of performing these tasks [26]. However, existing tools to collect social media data provide researchers with specific forms of data such as Twi-FFN as a module of the Twiscraper project [15] provides researchers with a list of followers of a user and people followed by a user while Storywrangler provides information on daily usage of specific words, language volume, real time word usage, trending topics and most used words on [16].

Social media analytics tools are of interest to a variety of different parties that are concerned with data from social media platforms and in the knowledge, they can acquire regarding the topic of mass communication and about different user groups by analyzing social media data [6]. These tools are either provided by the owners of such social media platforms or by third parties that implemented their own applications to provide their clients with the tools necessary to monitor and analyze social media data[26]. In the research domain, social media analytics has been applied in a variety of different research disciplines with great success [27]. A challenge that these tools must overcome is the fact that a standardized access method for gaining access to the data of social media platforms does not exist. Each platform has its own methods and requirements. In order to retrieve the user-generated data from such platforms application programming interfaces (APIs) can be used but each of these APIs has its own restrictions which makes it difficult to develop a tool for multiple platforms [9], [27]. APIs represent a access point for datatransfer by providing programmers with a set of functions and commands which they can use to implement software that is able to interact with an external system such as Twitter or YouTube.

3. Design goals

For the development of the SMART-Portal several design goals were set in order to provide the development process of the tool with certain guidelines. These goals were defined by expressing our expectations and wishes towards the developed tool in order to address our aspirations when working with data from Twitter and news websites.

As the first design goal, the accessibility of the application was addressed. The aim was to make the tool as accessible as possible for the end-user with as little effort as necessary for them. This does not only mean that the setup required to use the tool should be reduced to a minimum, but it also includes that different users should be able to use the application on different operating systems without having to perform additional actions. Therefore, the SMART-Portal was implemented as a web application that is accessible through a regular web browser such as Mozilla Firefox or Google Chrome. By doing so we ensured that the Portal is available on every technological device that has access to the internet. This increases the accessibility of this tool by only requiring the users to know the web address to the application in addition to only requiring researchers to register with an e-mail-address at the Portal. By implementing the tool as a web application all the computation is done on the server-side which leads to the hardware requirements for the users being also minimal. Additionally, only open-source or freely available resources were used.

As the second design goal, the ease of use of the tool was selected. The goal here was to minimize the effort it takes from the user to track data on Twitter via the SMART-Portal. Therefore, the information required to track data was reduced to the essential information necessary needed. This includes the definition of search terms, a period in which tweets should be tracked, and the language in which the terms should be tracked. To export the dataset the user is required to request the dataset through the user interface. Afterward, the dataset is sent to them attached to an e-mail.

Third, the tool should allow users to see the status of their project. For this purpose, an additional view was implemented that displays all projects that are being tracked in addition to presenting information such as the period in which Twitter data is being tracked, the name of the project, and its status in regard to if the tracking process has been completed or if it is still ongoing.

Fourth, the tool should enable the user to adjust the information provided to the system for projects that are still being tracked. This is accomplished by selecting a project from the previously mentioned project overview named in the third design goal. The user is able to adjust the name as well as the start and end date for the tracking process in addition to adding and removing search terms that should be tracked in that specific project.

The fifth design goal is that the researchers should be able to export the collected data in multiple formats. The SMART-Portal, therefore, offers the possibility of exporting the dataset as an Excel sheet in the CSV format in addition to providing the researcher with the capability of exporting the dataset as a CSV file adjusted for the usage with Gephi to perform social network analysis.

Lastly, the tool should structure the exported data according to a predefined structure. The Twitter data is stored in a database with several tables that are linked with each other. The portal combines the data from these tables according to the association between them and creates the export file. The links are stored in the according tables in from of each other’s ids. Depending on the selected export format the structure of the exported data differs from each other.

4. SMART-Portal Architecture

The entirety of the application that enables researchers to track data consists of three components that work with each other. These components can be seen in figure 1.

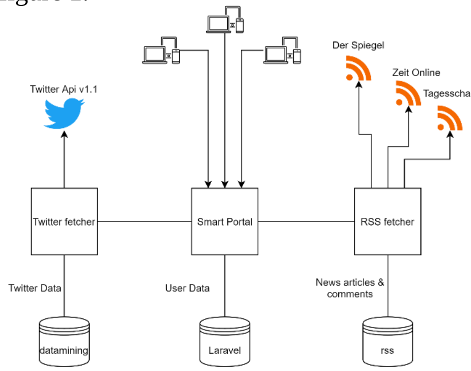


Figure 1System components

The SMART-Portal is located at the center of this application. It is the part of the application that the end-users interact with directly. This portal is the front end of the application and therefore the only component of it that possesses a user interface that the user utilizes with the intent of tracking data from Twitter or news websites. The portal is directly connected to a MySQL database that stores user account information that the users provide in the registration process. This database is checked when a user tries to log in to the portal with their credentials in order to verify if the user trying to log in has an account for the portal.

The portal itself is implemented using the languages PHP, JavaScript, HTML, and CSS and it utilizes multiple libraries and tools that can be seen in figure 2.

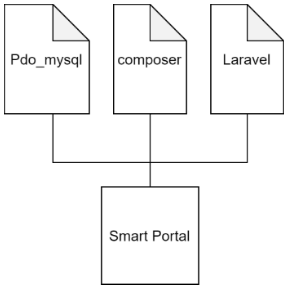


Figure 2 SMART-Portal libraries & tools

The composer library[1] is utilized for the purpose of managing dependencies and libraries of the SMART-Portal as a PHP-based software. In order to access the MySQL databases that are required for the portal, the PHP-based library Pdo_mysql[2] is used. Furthermore, Laravel[3] was utilized as a basis for the development of the SMART-Portal. It is a PHP web framework that follows the Model-View-Controller (MVC) principle and provides its users with multiple tools to improve their web development experience. In addition, it specifies how the application is structured by providing a folder structure for the implementation of the application. This makes it a key component of the SMART-Portal.

The MVC principle is a software design principle used for designing graphical user interfaces. It divides the interface into three components that are interconnected. The model component represents and manages the data of the application whereas the view visualizes the data. The controller reacts to user input on the view and performs actions or passes commands and information to the model.

The second component of our tool is the Twitter-Fetcher. This component is the main backend component of this application. It is implemented using the programming language Java and it utilizes several Java libraries to access and store Twitter data. These libraries can be seen in figure 3. For the implementation Java was chosen due to the familiarity of the developers with this language as well as with existing libraries and their documentation.

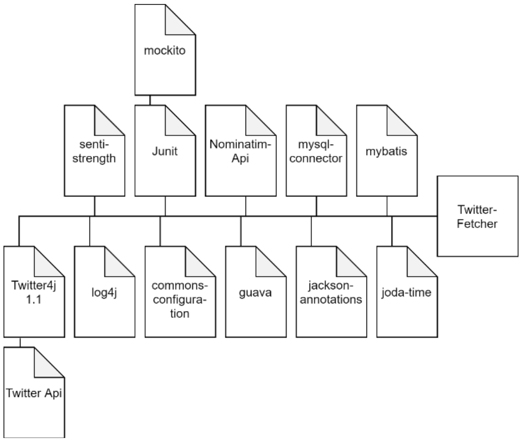


Figure 3Twitter-Fetcher libraries & tools

In order to store the data that the Fetcher retrieves from Twitter through the Twitter API, it uses a MySQL database with multiple interconnected tables. This is due to the relational nature of the data that is retrieved through the API which makes the usage of a relational database an appropriate approach. This approach is justified for instance through the fact that a tweet has a single user as its author, but other users might retweet that specific tweet which creates a relation between a retweeting user and that specific tweet. This database is also the database where the projects with their search terms and their tracking period are stored after they are created through the SMART-Portal. Therefore, the SMART-Portal itself as a component has direct access

[1] https://getcomposer.org/download/ on 13.04.2021

[2] https://www.php.net/manual/de/ref.pdo-mysql.php on 13.04.2021

[3] https://laravel.com/ on 13.04.2021

to two databases. In order to access the Twitter API with the goal of retrieving tweets and other Twitter data the Java library, Twitter4J[4] is used. Twitter4J is a Java library that allows developers to access the functionalities of the Twitter API through the Java programming language by enabling the integration of the Twitter API into a Java application. It provides developers with predefined methods to access the endpoints of the Twitter API in order to access the available Twitter data. The Twitter API currently used within the Twitter-Fetcher is its standard 1.1 version. The 1.1 version is also available in an enterprise and premium version which provides developers with more features and access to data but both versions are not as accessible compared to the standard version due to both of them not being free to use. The API consists of different endpoints which provide developers and researchers with different methods and access to different types of data for example the GET users / show-method retrieves a single user object that contains information about a specific user such as their screen name or their user id. One big limitation of the Twitter API is that it restricts every user and application to 180 requests within a window of 15-minutes on the GET / SEARCH endpoint that is used to retrieve tweets from Twitter. After each 15-minute window passes this limit is reset which allows the application to make 180 new requests through the API. Each request made is capable of returning up to 100 tweets which means that an application is able to retrieve up to 1.7 million tweets in a day. In order to counteract this limit set by the API, the SMART-Portal is able to retrieve data from Twitter using multiple Twitter accounts.

Further libraries used in the Twitter-Fetcher include the Nominatim-API[5] for the purpose of geocoding - turning addresses into geographical coordinates – and reverse geocoding - turning geographical coordinates into addresses and the SentiStrength[6] library. The SentiStrength library is used to perform sentiment analysis by determining if tweets express positive or negative sentiment. If determinable each tweet is given a value between -4 for very negative sentiment to +4 for very positive sentiment. This is done by checking tweets for words from a dictionary file including positive and negative words and performing calculations defined within the SentiStrength library. The sentistrength library was preferably chosen over alternatives such as AFINN due to being focused on short web message and being already applied in research projects [28] and evaluated in peer reviewed scientific articles [29] For the purpose of performing unit tests the Junit[7] library together with the Mockito[8] library is used. Unit tests are tests performed in software development where single functional components of the software are tested.

Mockito is a unit test library in Java that allows the usage of test objects that can be used as stand-in objects for a unit test. It allows the simulation of such test objects for example when the real object that is simulated by the test object is not fully implemented yet. For a unit test, the developer can specify the behavior of the simulated object when the tested unit accesses it.

In order to connect to a MySQL database, the Java library MySQL-connector is utilized. It not only enables Java applications to connect to a MySQL database but also to store and retrieve data and to execute other queries on the database. The Java library mybatis[9] is utilized in collaboration with the MySQL-connector library. It enables the application to execute SQL queries that have been stored in XML files.

The Jackson-annotations library is used to map Java objects to JSON objects and JSON objects to Java objects which is necessary due to tweet objects being retrieved as JSON objects through the Get/tweets endpoint of the Twitter API. This endpoint is used to access and retrieve Tweet data such as the content of a tweet or the date the tweet was created. This endpoint is not directly utilized but through the corresponding methods provided by the Java library Twitter4J that is used in this application.

For the purpose of logging information, the log4j2[10] library is utilized. This library is simple to configure and provides a wide variety of possibilities for the purpose of logging information. Additionally, the Commons-configuration[11] library is used to enable the Fetcher to read configuration information of the database that has been stored into a configuration file. This library provides an interface with the necessary methods to access this information from different sources. The Twitter-Fetcher also includes guava[12] which is a set of libraries for Java from Google that incorporates for example new collection types or utilities for caching, strings, etc. Another important tool for the Twitter-Fetcher but also for the RSS-Fetcher is Maven[13]. This tool is used to manage these Java projects and the dependencies of them. Lastly, the Fetcher incorporates the Joda-Time[14] library which provides the application

[4] http://twitter4j.org/en/ on 13.04.2021

[5] https://nominatim.org/release-docs/latest/api/Search/ on 13.04.2021

[6] http://sentistrength.wlv.ac.uk/ on 13.04.2021

[7] https://junit.org/junit5/ on 13.04.2021

[8] https://site.mockito.org/ on 13.04.2021

[9] https://mybatis.org/mybatis-3/getting-started.html on 13.04.2021

[10] https://logging.apache.org/log4j/2.x/javadoc.html on 13.04.2021

[11] https://commons.apache.org/proper/commons-configuration/ on 13.04.2021

[12] https://github.com/google/guava on 13.04.2021

[13] https://maven.apache.org/what-is-maven.html on 13.04.2021

[14] https://www.joda.org/joda-time/ on 13.04.2021

with the necessary functionality for date-time calculations.

The last component that is linked to this application is the RSS-Fetcher which is the second back-end component of the SMART-Portal application. The main purpose of this component is to retrieve news articles as well as comments for each article in addition to comments on other comments. These articles and comments are then stored in an independent MySQL database. The Fetcher itself is implemented in Java and it utilizes multiple Java libraries that are visualized in figure 4.

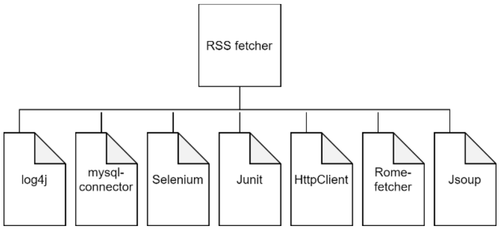


Figure 4RSS-Fetcher libraries & tools

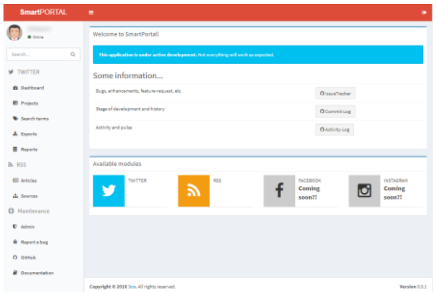
The RSS-Fetcher uses several tools and libraries that are also utilized in the Twitter-Fetcher. These are the MySql-connector, Log4j2, Junit, and Maven. In addition to these libraries and tools, the RSS-Fetcher incorporates the Selenium[15] library which enables the Fetcher to emulate user behavior on a webpage. This is useful for cases where comments are only available after selecting another comment. The HttpClient is used to retrieve and send resources over a network. As an additional library, the Rome-Fetcher library is included in the RSS-Fetcher[16]. It enables the Fetcher to retrieve RSS-Feeds. In combination with this library, the Jsoup[17] library is utilized. This library empowers the application to parse through HTML content that has been retrieved by the RSS-Fetcher and to extract information from it.

5. User Interface of the SMART-Portal

The current version of the SMART-Portal is online and can be accessed through the network of the University of Duisburg-Essen. In order to access the Portal through that network, researchers have to be connected to the university's network either by being located at the university and connecting to it directly or by using a VPN to connect to it remotely.

The SMART-Portal is accessible through a web browser by navigating to the web address of the portal. Afterwards, the researcher has to register to the SMART-Portal by filling out the registration form that is prompted by pressing the registration button. After the user has submitted the form an administrator of the portal has to approve the registration request. When the application is approved the researcher is able to access the portal and its functions by logging into the portal using the e-mail address and the password that they provided in the registration form as their login credentials.

After the registration process and logging into the portal the user is referred to the homepage of the portal where they are met with a welcome message, available modules such as the Twitter-Fetcher, and additional information for example links to the GitHub repository of the portal where they can access the development history in addition to expressing their ideas for enhancements to the Portal. The homepage can be seen in figure 5.



Welcome to SmartPortal!

This application is under active development. Not everything will work as expected.

Some information...

Bugs, enhancements, feature-request, etc

Stage of development and history

Activity and pulse

Available modules

Coming soon?!

Coming soon?!

Copyright © 2016 3cs. All rights reserved.

Version 0.0.1

Figure 5 SMART-Portal Homepage

Located on the far left of the website is the navigation bar of the portal. This bar allows the users to access different areas of the portal by pressing the buttons located within it. Each button is labeled with a text that simplifies the access to different areas and functions of the portal such as the export function or access to the area where all projects that are being tracked or have been tracked previously are located.

The navigation bar can be divided into three sections. The first section is dedicated to Twitter. It includes a dashboard section that is still in development, a projects section that shows the priorly mentioned overview of all projects that are currently being tracked as well as the projects that have been previously tracked, and a search terms section that shows an overview of which search terms are being tracked in which project. Furthermore, a button to request the exportation of projects that have finished being tracked over Twitter as a dataset is also located in this section. Lastly, a reports button is located in the Twitter area which refers to a section where descriptive statistics about the tracked data are presented. The project overview includes a button at the top to add a new project that should be tracked on Twitter. After clicking that button, the user is referred to a form that is used to acquire the necessary information to track a project via Twitter. This

[15] https://www.selenium.dev/ on 13.04.2021

[16] http://rometools.github.io/rome-fetcher/ on 13.04.2021

[17] https://jsoup.org/ on 13.04.2021

information includes a name for the project, the search terms, the priority of the project, the start and end date for the tracking process, and the language for the search terms that indicates which language the tweets that are tracked should be in. The user has the option to select either German, English, Spanish, or the selection of no language restrictions for tweets.

The second section of the navigation bar is dedicated to retrieving data from news sites including news articles and comments for each article. This is achieved by accessing the RSS feeds of each news website. In its current state, this area of the navigation bar includes an overview of articles and an overview of available news sources. By activating a source the articles from that source are tracked.

The last section within the navigation bar is dedicated to the maintenance functionality of the SMART-Portal. This includes an admin area that is only accessible for administrators of the portal, a button to report bugs, a button referring to the GitHub repository of the portal as well as a button that refers to the documentation of the Portal.

6. Twitter-Fetcher

The SMART-Portal is the user interface that researchers interact with in order to create projects that they want to track on Twitter. The Twitter-Fetcher on the other hand is the component of the Portal that is used to access the user data from Twitter which includes tweets, dates, usernames, etc. The main task of the Fetcher is to access the database where the projects are located and to retrieve Twitter data according to the specifications of these projects from Twitter itself. Afterward, the Fetcher is tasked with storing said data into multiple interconnected tables within the designated database. The relevant specifications to retrieve Twitter data are for example search terms, the period for which these terms should be tracked, and the language that they should be tracked in.

The architecture used to implement the Twitter-Fetcher is based on the master-worker principle where the master thread has the task of coordinating the worker threads, assigning them tasks to perform in addition to being in charge of collecting the results retrieved by the other threads. Within the configuration file of the Fetcher, an interval is set that defines the rate at which the application checks the database for search terms that need to be searched for on Twitter. After identifying the search terms that are currently not being tracked the application orders them by their priority, set by the researcher that created the project over the interface of the SMART-Portal. For the search terms that have the same priority level, the application sorts them by the interval beginning with the shortest, and those with the same interval are then sorted by their creation date starting with the newest search terms. After the search terms are sorted the application checks if a Twitter account that is registered at the application is available and has not reached its limit to make requests yet. If such an account is available, the master spawns a worker thread.

These worker threads are referred to as minions and they perform the task of executing the queries that are priorly defined in the code of the application or as XML files. This is done in order to retrieve the Twitter data through the Twitter API for the available search terms. Minions send queries to the Twitter API until their rate limit is reached. Afterward, they save the returned tweets for these queries.

The master thread receives these tweets from the minions that he spawned and stores them until a limit is reached that is set in the configuration file. After that limit is reached the master thread spawns a different type of worker thread - a treasurer thread. The task that the treasurer fulfills is that of receiving the tweets that the master thread accumulated and connecting to the database followed by storing these tweets permanently in said database.

In order to retrieve the tweets, the Twitter-Fetcher does not simply retrieve the 100 newest tweets since the last query was executed for a search term. This would lead to tweets being lost if since the last time tweets were retrieved for that search term more than 100 tweets with that term were made. Instead, a different approach is taken for the retrieval of data that has the advantage of ensuring that data for a search term is not lost. The approach used within the implementation of the Twitter-Fetcher is that of performing queries in iterations instead of simply retrieving the newest 100 tweets. When more than 100 tweets were posted for a search term since the last iteration, multiple queries are performed in order to retrieve every single tweet since then. In contrast, if less than 100 tweets were posted since the last iteration, then the 100 tweets that are retrieved will simply contain tweets that are already stored in the database which will be filtered out based on the id they receive from Twitter.

The first iteration of retrieving data is performed when a search term is added to the database by a researcher using the SMART-Portal overlay. This approach works backward through the Twitter timeline and it retrieves the 100 most recent tweets for that search term after it was added to the database. Each tweet retrieved by the Twitter API has its own unique id that it receives from Twitter. A minion identifies the id of the oldest returned tweet and sets it as the max_id parameter and sets for that search term that it is now in an iteration so that other minions do not also retrieve the same tweets for that search term. Then it retrieves the most recent tweets prior to that specific id. Afterward, it will now have the 200 most recent tweets. This sequence is performed by the minion repeatedly until the oldest tweet is older than the set search start date or the five-day limit. An iteration by going backward through the Twitter timeline in order to retrieve tweets is complete when this state is reached. These tweets then are returned to the master in order for him to hand them over to the treasurer so that he can store them in the database. Afterward, the minion calculates how long it will take for 100 new tweets to be posted. When this interval has expired the minion will start a new iteration backward through the Twitter timeline and it will retrieve the 100 most recent tweets for that search term until the start date of the last iteration or the five-day limit is reached. This is performed by the Twitter-Fetcher repeatedly in the background for multiple search terms from several different projects.

The interval is capped at five days because the Twitter API only returns tweets that are not older than six to nine days. This cap is set for this application in order to ensure that even search terms with few tweets are covered by the application completely.

7. RSS-Fetcher

The RSS-Fetcher is a backend component that retrieves news articles as well as comments for each article in addition to comments on other comments. These articles and comments are then stored in an independent MySQL database. The Fetcher itself is implemented in Java by utilizing multiple Java libraries.

Each news site needs to be incorporated into the application individually. There is no universal template to retrieve the articles and comments from the available RSS-Feeds of every existing news website. This is due to the fact that they differ in how each new agency structures its websites. This leads to the necessity of implementing each news website as their own Java class in order to access their articles and comments. The difference in how a website is structured can be a problem for example when retrieving the comments made for comments on a news article. Some news websites show all comments whereas others hide the comments on comments until the user clicks on them which makes it difficult to retrieve them. For this specific issue, we employ the selenium library which enables the application to simulate user interaction with the website through the web browser. By doing so the RSS-Fetcher is able to simulate a click of a user on a comment which leads to its comments being shown. By doing so the hidden comments become accessible for the Fetcher which enables it to retrieve them and store them into the designated database.

After each news website is implemented as its own Java class each website is added to the news sources in the database with information such as their web address and feed language. The RSS-Fetcher accesses the RSS feeds of every news source from the database where the status is set to active according to the specifications of each news website within their corresponding implementation as a Java class. In order to access the RSS feeds and retrieve the desired articles of different news websites the Java library, Rome-Fetcher is utilized. After accessing the RSS feed of a news website the RSS-Fetcher retrieves information such as the article title, the content of the article, the date of its creation, its comments by registered users, etc. and stores them into the database. In order to access this information for each article from the retrieved RSS feeds Jsoup as a Java library is used. It enables the Fetcher to parse through the HTML code obtained through the Rome-Fetcher library and to extract the specified data from a news article. Afterward, the information is stored in the database.

8. Conclusion

The goal of this work was to provide researchers with a guideline for the conceptualization of their own tool for similar purposes by illustrating how we created the SMART-Portal. The advantage of this tool is that it is developed with the purpose of being used by researchers to conduct their research. However, the approach described in this guideline is by no means the only approach to develop such a tool and should be treated as a suggestion intended to inspire the recreation of a similar tool, or the creation of new ones. Through the description of our tool a contribution to the existing knowledge base is made which can be leveraged to further develop tools inspired by our tool.

We addressed our research question which was “How can a scientific tool that helps researchers to collect data long term from multiple sources based on predefined parameters be developed?”, by demonstrating how we developed the SMART-Portal.

The purpose for collecting data is to provide researchers with datasets for different topics from news websites or Twitter so that they can analyze the data. The tool does not restrict the topics that can be tracked. Furthermore, it can be used by researchers of different research fields to acquire data from these sources.

As of 2020 Twitter API v2 was introduced. This Version of the API provides developers with access to more features and data such as current poll results and the new Academic Research product track. This track is limited to researchers and grants them free access to the full-archive of Tweets. Furthermore, it significantly increases the monthly cap of Tweets that a developer is able to retrieve in addition to granting access to more features. The new API is currently in early access and therefore constantly changing and new endpoints are being added over time. In a future version of the SMART-Portal the new API will be utilized to ensure that researchers can continue to use the tool.

At the time of writing this paper, we could not find any scientific tool which is dedicated to collecting data from Twitter and news websites through a user interface as it is described in this paper. Further, no scientific tool was identified that simplifies the process of tracking data for researchers which could be referenced as a method for acquiring research data by minimizing information needed for the tracking process. More often in existing studies, the method was not described, or commercially available tools were used, which often meant that information about how these tools function is not publicly available. Our application is meant to change that. Furthermore, our tool provides datasets that include a variety of data which to our knowledge existing tools do not provide together. The SMART-Portal as a tool provides several novelties for researchers.

First, by using the Twitter search API to retrieve data from Twitter researchers are able to acquire data for up to five days prior to the date of sending the query to retrieve the data. This enables researchers to track a topic retroactively through the SMART-Portal within a five-day period after they identified this topic. Through the SMART-Portal they do not necessarily need to start tracking the topic immediately when it occurs which in itself can be a problem because most often it takes time to identify a topic of interest. Therefore, it is not easy to track data for a phenomenon such as a natural disaster or a topic that goes viral immediately after they occur except when systems are set in place that are already tracking for such cases before they occur. Thus, the SMART-Portal provides researchers with a valuable opportunity to start tracking a topic even five days after it occurred. The tracking period can be set to start for example four days prior to the current day in order to track relevant data that has been previously posted. The only issue in regard to tracking Twitter data is that due to the usage of the search API and not the streaming API the SMART-Portal is not able to track live data. However, by using the search API instead this data is also stored eventually meaning that the live data is not lost and can be used within the research also.

Second, the SMART-Portal provides researchers with two formats for the export of tracked Twitter data. The first format is in form of a table as a CSV file that contains all relevant information collected such as tweets, usernames, retweet count, date, etc. Whereas the second format is specifically designed for further usage with Gephi in order to perform social network analysis. By providing multiple export formats the SMART-Portal further supports researchers in their research by providing them with multiple datasets in different structures that are still based on the same data collected. This empowers researchers to analyze different aspects of the tracked topic.

Third, this tool enables researchers not only to track data from Twitter but also from different news websites. It provides researchers with news articles, comments for each article, and answers of other users on article comments. Several popular news websites are already included.

Limitations for the SMART-Portal application are that it currently only collects data from Twitter and predefined news sources by the developers of the tool. But for future iterations of the tool, the addition of more sources for data collection is planned, such as YouTube and Reddit. Through the linked GitHub repository, the researchers using the tool are able to make suggestions for which sources they would like to see included in the tool. Another limitation for it is that currently, only two formats are available for data extraction which will be expanded in future iterations of the application. Furthermore, the SMART-Portal is currently only accessible for members of one university. In the future it is planned to extend accessibility to a wider range of researchers and to publish the source in repositories such as GitHub.

Future work on this tool will include the identification of additional relevant news websites. Appropriate websites will be incorporated into the RSS-Fetcher meaning they will receive their own Java class and will be trackable through the Fetcher. After identifying which data formats are also relevant for researchers they will be added to the portal in the future. Additionally, to the reports section that shows statistics for all projects and tweets collected a dashboard will be implemented. This dashboard will allow users of the portal to select a specific project that has been fully tracked. Afterward, it will visualize statistics for the selected project. This will include the visualization of collected data as charts for improved readability, descriptive statistics, and results of simple calculations such as the average frequency of tweets. The charts will present information such as the number of tweets for a specific search term over a selected period of time as a bar chart where each bar represents the number of tweets for a specific day. Moreover, we intend to evaluate the tool with researchers for its acceptance by users, novelty and its value for conducting research in a future study.

10. References

[1] M. Beier and K. Wagner, “Social media adoption: Barriers to the strategic use of social media in SMEs,” 2016.

[2] W. He, W. Zhang, X. Tian, R. Tao, and V. Akula,

"Identifying customer knowledge on social media through data analytics," *J. Enterp. Inf. Manag.*, vol. 32, no. 1, 2019, doi: 10.1108/JEIM-02-2018-0031.

[3] M. Mirbabaie, D. Bunker, S. Stieglitz, J. Marx, and C. Ehnis, "Social media in times of crisis: Learning from Hurricane Harvey for the coronavirus disease 2019 pandemic response," *J. Inf. Technol.*, vol. 35, no. 3, 2020, doi: 10.1177/0268396220929258.

[4] G. Enli, "Twitter as arena for the authentic outsider: exploring the social media campaigns of Trump and Clinton in the 2016 US presidential election," *Eur. J. Commun.*, vol. 32, no. 1, 2017, doi: 10.1177/0267323116682802.

[5] P. Singh, Y. K. Dwivedi, K. S. Kahlon, R. S. Sawhney, A. A. Alalwan, and N. P. Rana, "Smart Monitoring and Controlling of Government Policies Using Social Media and Cloud Computing," *Inf. Syst. Front.*, vol. 22, no. 2, 2020, doi: 10.1007/s10796-019-09916-y.

[6] S. Stieglitz, M. Mirbabaie, B. Ross, and C. Neuberger, "Social media analytics – Challenges in topic discovery, data collection, and data preparation," *Int. J. Inf. Manage.*, vol. 39, 2018, doi: 10.1016/j.ijinfomgt.2017.12.002.

[7] H. Baars and H. G. Kemper, "Management support with structured and unstructured data - An integrated business intelligence framework," *Inf. Syst. Manag.*, vol. 25, no. 2, 2008, doi: 10.1080/10580530801941058.

[8] Z. Tufekci, "Big Questions for social media big data: Representativeness, validity and other methodological pitfalls," 2014.

[9] S. Stieglitz, L. Dang-Xuan, A. Bruns, and C. Neuberger, "Social Media AnalyticsSocial Media Analytics," *WIRTSCHAFTSINFORMATIK*, vol. 56, no. 2, 2014, doi: 10.1007/s11576-014-0407-5.

[10] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," 2010, doi: 10.17148/ijarcce.2016.51274.

[11] B. Batrinca and P. C. Treleaven, "Social media analytics: a survey of techniques, tools and platforms," *AI Soc.*, vol. 30, no. 1, 2015, doi: 10.1007/s00146-014-0549-4.

[12] R. Tekumalla and J. M. Banda, "Social media mining toolkit (Smmt)," *Genomics and Informatics*, vol. 18, no. 2, 2020, doi: 10.5808/GI.2020.18.2.e16.

[13] D. Henry, "TwiScraper: A Collaborative Project to Enhance Twitter Data Collection," 2021, doi: 10.1145/3437963.3441716.

[14] T. Alshaabi et al., "Storywrangler: A massive exploratorium for sociolinguistic, cultural, socioeconomic, and political timelines using Twitter," Sci. Adv., vol. 7, no. 29, 2021, doi: 10.1126/sciadv.abe6534.

[15] O. G. Ayodeji and V. Kumar, "Social media analytics: A tool for the success of online retail industry," *Int. J. Serv. Oper. Informatics*, vol. 10, no. 1, 2019, doi: 10.1504/IJSOI.2019.100630.

[16] P. Garg, B. Gupta, S. Dzever, U. Sivarajah, and V. Kumar, "Examining the Relationship between Social Media Analytics Practices and Business Performance in the Indian Retail and IT Industries: The Mediation Role of Customer Engagement," *Int. J. Inf. Manage.*, vol. 52, 2020, doi: 10.1016/j.ijinfomgt.2020.102069.

[17] F. Mirzaalian and E. Halpenny, "Social media analytics in hospitality and tourism: A systematic literature review and future trends," *Journal of Hospitality and Tourism Technology*, vol. 10, no. 4. 2019, doi: 10.1108/JHTT-08-2018-0078.

[18] Z. Xiang, Q. Du, Y. Ma, and W. Fan, "A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism," *Tour. Manag.*, vol. 58, 2017, doi: 10.1016/j.tourman.2016.10.001.

[19] M. L. Jibril, I. A. Mohammed, and A. Yakubu, "Social Media Analytics Driven Counterterrorism Tool to improve Intelligence Gathering towards Combating Terrorism in Nigeria," *Int. J. Adv. Sci. Technol.*, vol. 107, 2017, doi: 10.14257/ijast.2017.107.03.

[20] M. Mirbabaie, C. Ehnis, S. Stieglitz, and D. Bunker, "Communication roles in public events – A case study on twitter communication," in *IFIP Advances in Information and Communication Technology*, 2014, vol. 446, doi: 10.1007/978-3-662-45708-5_13.

[21] M. Mirbabaie and E. Zapatka, "Sensemaking in social media crisis communication – A case study on the Brussels bombings in 2016," 2017.

[22] J. Golbeck, J. Gerhard, F. O'Colman, and R. O'Colman, "Scaling Up Integrated Structural and Content-Based Network Analysis," *Inf. Syst. Front.*, vol. 20, no. 6, 2018, doi: 10.1007/s10796-017-9783-x.

[23] Y. A. Argyris, Z. Wang, Y. Kim, and Z. Yin, "The effects of visual congruence on increasing consumers' brand engagement: An empirical investigation of influencer marketing on instagram using deep-learning algorithms for automatic image classification," *Comput. Human Behav.*, vol. 112, 2020, doi: 10.1016/j.chb.2020.106443.

[24] P. R. Spence, D. D. Sellnow-Richmond, T. L. Sellnow, and K. A. Lachlan, "Social media and corporate reputation during crises: the viability of video-sharing websites for providing counter-messages to traditional broadcast news," *J. Appl. Commun. Res.*, vol. 44, no. 3, 2016, doi: 10.1080/00909882.2016.1192289.

[25] A. Chinnov, P. Kerschke, C. Meske, S. Stieglitz, and H. Trautmann, "An overview of topic discovery in Twitter communication through social media analytics," 2015.

[26] L. E. Young, S. Soliz, J. J. Xu, and S. D. Young, "A review of social media analytic tools and their applications to evaluate activity and engagement in online sexual health interventions," *Preventive Medicine Reports*, vol. 19. 2020, doi: 10.1016/j.pmedr.2020.101158.

[27] A. Geissinger, C. Laurell, and C. Sandström, "Digital Disruption beyond Uber and Airbnb—Tracking the long tail of the sharing economy,"

Technol. Forecast. Soc. Change, vol. 155, 2020, doi: 10.1016/j.techfore.2018.06.012.

[28] T. Baviera, A. Sampietro, and F. J. García-Ull, "Political conversations on Twitter in a disruptive scenario: The role of 'party evangelists' during the 2015 Spanish general elections," Commun. Rev., vol. 22, no. 2, 2019, doi: 10.1080/10714421.2019.1599642.

[29] M. Thelwall, "Gender bias in sentiment analysis," Online Inf. Rev., vol. 42, no. 1, 2018, doi: 10.1108/OIR-05-2017-0139.