Web Harvesting for Smart Applications

Detecting Novel Topics Related to Twitter News Events
Primary Topic: SEMI, Secondary Topic: IENLP
Course: 2022-1A – Group: 37 & 65 – Submission Date: 05-10-2022

Roel Leenders
University of Twente
r.c.leenders@student.utwente.nl

Leander van den Heuvel
University of Twente
l.p.w.vandenheuvel@student.utwente.nl

ABSTRACT

Twitter is often considered as a valuable source for journalists. It is frequently used to find additional information, new topics of interest, or to gather statements of eyewitnesses. However, due to its high volume of information, there is a risk of information overload. Therefore, this study aims to provide easier access to possible topics of interest for journalists. In order to achieve this, a smart web harvesting application has been developed that is able to extract segments from Tweets, which are then ordered by frequency and displayed to the journalist. The segmentation method is inspired by SEDTWik and compares n-grams from the tweet text to Wikipedia Titles. Although the method proposed in this study is able to extract frequently occurring segments at a reasonable performance, a follow-up study needs to be conducted to evaluate if the segments are actually relevant for journalists.

KEYWORDS

Web Harvesting, Twitter, Event Detection, SEDTWik

1 INTRODUCTION

Using Twitter as a source in journalism can be a tricky affair. Although Twitter can provide a platform for valuable information directly from the source of the event [5, 9], there is the risk of misinformation and information overload [10]. A plethora of surveys on the use of Twitter by journalists suggest frequent use of the social platform for different purposes, like finding new topics of interests, gathering statements of eyewitnesses, and rapid information collection in case of breaking news events [2, 7, 9, 13]. Although Twitter already provides an intuitive interface, the identification and organization of potential relevant sources and novel information remains highly time consuming [8]; a high variance in the validity of different sources and information overload due to the large amount of Tweets [14] often prevent journalists from having a clear overview of the event. A possible solution to this problem may be a smart application that is able to provide journalists with a clear overview of these novel Twitter events. To aid the development of this smart application, this paper attempts to answer the following research question:

- **RQ**: To what extent can relevant additional topics related to news events on Twitter be detected using web harvesting?

To answer the research question, this paper will first provide the relevant background information and related work. Subsequently, an elaboration of the final approach will be provided. Furthermore, the various attempted segmentation and grouping experiments will evaluated. Finally, the results of the experiment and their limitations will be discussed.

2 BACKGROUND AND RELATED WORK

The development of the smart application can be broken down into a variety of sub-topics. This section will provide the necessary background information and related work for each of these topics.

2.1 Smart Applications

The software industry has made various attempts to develop smart applications that may aid journalists in their writing process. Among popular applications are, for example, *Twitterfall* [16] which provides users with a real-time overview of incoming Tweets filtered by a variety of criteria; *Followerwonk* [6], which analyses Twitter bios and enables journalists to more easily connect with other journalists, researchers, and high-profile Twitter users; and *Tweet Archivist* [1], which provides historical insights related to certain Twitter search keywords and enables journalists to grow their own brand. Surprisingly, with regard to the difficulties journalists face while using Twitter (see section 1), it seems that no smart application has been developed that specifically aids journalists in the identification and organization of potential relevant sources and novel information.

2.2 Web Harvesting

Web harvesting refers to the extraction of data from the World Wide Web. For the development of the proposed smart application it is necessary to extract information from Twitter. Fortunately, the Twitter API [17] enables programmatic access to developers with the ability to, among other things, retrieve large amounts of Tweets. Although the API has a limit to request 500.000 Tweets, it should be sufficient to develop and test the smart application's ability to detect and organize novel information from Twitter. Moreover, for the retrieval of Tweets, the Twitter API uses a query language that allows users to query for specific criteria. This may enable journalists to query, for example, information from a specific country, in a specific language, and containing certain keywords. Please see figure 1 for an example query. For a full overview of the possibilities of the Twitter API, please see the provided documentation on the Twitter developer website [1].

```
(Russia OR Ukraine) lang:en place_country:GB
```

Figure 1: Twitter API query for requesting tweets containing the words 'Russia' or 'Ukraine', written in English and published in Great Britain

[1] https://developer.twitter.com/en/docs/twitter-api

2.3 Detecting and organizing novel information from Tweets

Several studies have made an attempt to automatically detect Twitter trends. Weng et al. [18], for example, made an attempt to detect Twitter events by clustering wavelet-based signals. Using wavelet analysis, their method is able to construct signals from the frequency of the occurrence of the words. Subsequently, these signals are used to filter out trivial words and to cluster the remaining words into seperate events. Additionally, a study conducted by Morabia et al. [11] attempts to automatically detect Twitter events by first segmenting the Tweets using Wikipedia titles, after which they extract bursty segments. Bursty segments are segments with a specific frequency that are identified within a fixed time window. After the extraction of the bursty segments, the authors cluster them to detect specific events. The method proposed by Morabia et al. is able to achieve state-of-the-art results.

3 APPROACH

As shown in figure 2 (see appendix A for a larger overview), to reach the desired result, the final architecture of the smart application consists out of various components. The section aims to describe the purpose and the inner workings of these components.

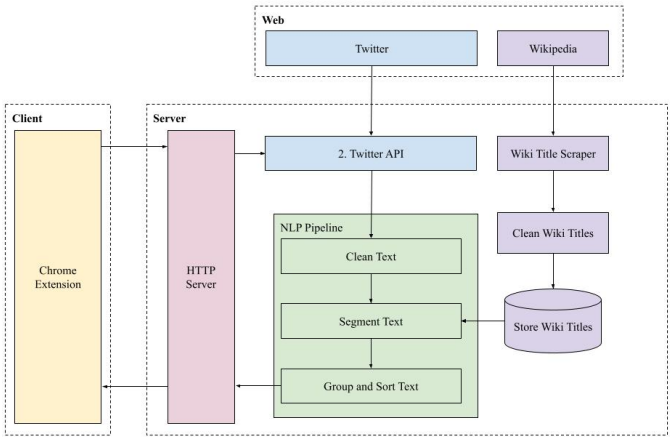


Figure 2: An overview of the architecture of the smart application

3.1 Chrome Extension

First of all, the Chrome extension serves as an graphical user interface for the user to use (see figure 3). The interface allows the user to enter Twitter search queries as described in section 2.2. Furthermore, since the smart application attempts to search for novel topics, the user can select any text on any web page filter out information the user already knows. For example, while writing an article about the Ukrainian war, the user select the text he has written while at the same time searching for Tweets related to the article. If the user's article, for example, contains the segment "Black Sea Fleet", the application will not display the exact topic as in the article but only other possible relevant topics. Once the smart application has processed the user request, the interface will display the extracted topics based on their frequency. Subsequently, the user is able to click the extracted topics in order to navigate to the original Tweets. The reason why the interface was developed as a Chrome extension is that it allows the user to more easily selected text from any web source while using the application.

Figure 3: Graphical user interface of the smart application

3.2 Twitter API

As described in section 2, the Twitter API was used to more easily retrieve the necessary information from Twitter. Based on a given search query, the smart application is able to retrieve a predefined amount of recently published and or popular Tweets. Since the API provided by Twitter can only request up to 100 Tweets per request, a custom function had to be implemented that would perform multiple requests in subsequent order; this allowed the smart application to use more Twitter data for its analysis.

3.3 NLP Pipeline

In order to extract novel topics related to breaking news events on Twitter, a custom Natural Language Processing (NLP) pipeline had to be built. This pipeline is partly based on the methodology proposed by Morabia et al. [11]. Morabia et al. propose a variety of Tweet cleaning methods that improve the overall quality of the segmentation. To achieve similar results, the NLP pipeline proposed in this paper cleans the retrieved Tweets by, first of all, removing non-ascii characters (e.g. emoticons), hyper-links, digits, punctuation marks, Twitter Hashtags, and user names. The words that remain get tokenized. Besides improving the quality of the eventual results, cleaning the tweets also improves the speed of the pipeline as it reduces the overall words that need to be processed in later (more computationally heavy) steps.

Subsequently, after the cleaning step, the tokenized Tweets get passed to the segmentation step within the pipeline. In order to

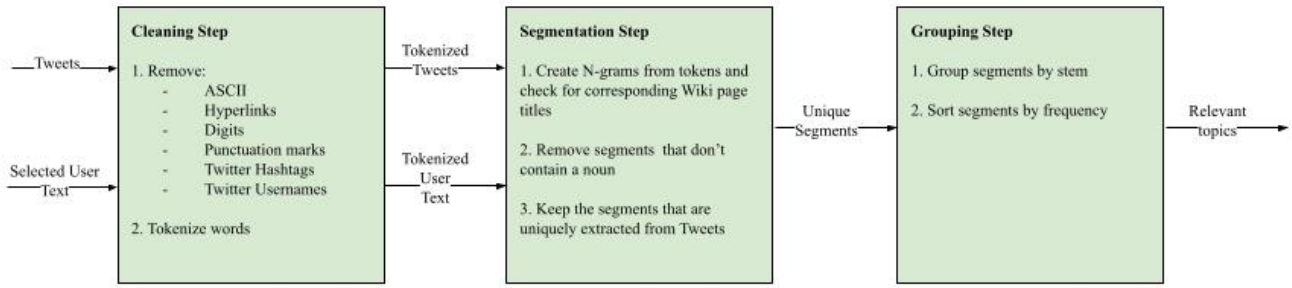


Figure 4: An overview of the NLP pipeline

successfully segment the Tweets, various segmentation methods have been tested, after which an alteration of the method proposed by Morabia et al. [11] seemed to be the most effective (see section **??** for the comparison). Using the tokens provided by the cleaning step, both the Tweets and the article text selected by the user are segmented using Wikipedia page titles. A segmentation is made by creating various lengths of n-grams from these tokens and comparing them to the list of Wikipedia titles. For example, the segmentation "Black Sea Fleet" can be found by allowing the segmentation algorithm to search for a combination of up to three words in the list of titles; although the Wikipedia titles "Black", "Sea", and "Black Sea" do exist, by allowing the algorithm to search up to, for example, three words, it is able to achieve more descriptive segments. However, since Wikipedia contains over more than six and a half million pages [20] increasing the maximum size of the n-gram will result in a significant of performance. Furthermore, since the average length of words in a Wikipedia page title is two words [11], we kept the max n-gram at 3 words. Additionally, since there are Wikipedia page titles that result in non-descriptive segments (e.g. "my" or "on"), only the segments that contained a noun were kept. To evaluate if a segment contained a noun the Python library TextBlob was used [15]. Furthermore, to only return the segments that are not included in the selected article text, an additional filter was developed to only return the segments that were unique to the Tweets.

The final step in the NLP pipeline aims to group and sort the segments by frequency. In order to better count the frequency of the segments, various stemming and lemmatization techniques were evaluated. As a result of the evaluation, the Porter Stemmer, provided by NLTK [12], was used. Due to this stemmer, segments like 'helicopters' and 'helicopter' were considered the same and therefore grouped together. Additionally, the grouped segments were sorted by frequency since the most occurring segments are most likely the most relevant for the user (see fig. 3).

3.4 Wikipedia Scraper

Since the NLP pipeline uses Wikipedia page titles for the segmentation of the text, a web scraper was developed to periodically update the available list of titles; as new topics get introduced on both Twitter and Wikipedia on a daily basis, it is necessary to update the page titles used in the app as well. Fortunately, Wikimedia [19] provides daily Wikipedia dumps that are easily accessible. As a result, the server automatically downloads the most recent page title information on a daily basis. Subsequently, when the server finished downloading, it cleans the page titles in a similar fashion to how the Tweets are cleaned (e.g. removing ASCII, stop words, digits, etc.) after which they are stored in a text file.

3.5 HTTP Server

In order to allow the user client to access the functionality available on the server, a simple HTTP server was implemented using the Python library FastAPI [3]. Using this library, the user is able to send the Twitter API search query and the selected text in the JSON body of a HTTP POST request after which the HTTP server returns the extracted Twitter topics.

4 EXPERIMENTS

To answer the question if the detected topics are actually novel and relevant for the journalists, it would be preferable to evaluate their opinion. However, since a user study is not a part of the assignment, the experiments presented in this paper will mainly attempt to provide an overview of the various methods used and their results compared to other methods. These experiments mainly revolve around the improvement of the NLP pipeline. More specifically, how the pipeline can detect new topics.

4.1 Segmentation

Although the methodology provided by Morabia et al. [11] provides a possible solution to extract relevant segments from Tweets, various other segmentation methods have been tested. In particular, we evaluated the duration of the segmentation process and the quality of the output. The segmentation methods evaluated are Stanford NER [4], TextBlob [15] and SEDTWik [11]. They are all evaluated using the same sample data set contain 50 Tweets retrieved using the Twitter API using the query from figure 1.

4.1.1 Stanford NER. [4] is a Named Entity Recognition software library that is trained to recognize and label persons, locations and organizations amongst a variety of other classes. Please see table 1 for the results of the experiment.

4.1.2 TextBlob. [15] is a Python library for text processing that provides functionality like, amongst others, noun phrase extraction,

Duration	31.5900 seconds
Sample output	('loss', 'O'), ('crimea', 'O'), ('russia', 'LOCATION'), ('already', 'O'), ('looking', 'O'), ('forward', 'O'), ('russia', 'LOCATION'), ('dlc', 'O'), ('even', 'O'), ('though', 'O'), ('coming', 'O'), ('month', 'O'), ('hope', 'O'), ('soon', 'O'), ('get', 'O'), ('popular', 'O'), ('opinion', 'O'), ('peer', 'O'), ('pressure', 'O'), ('get', 'O'), ('see', 'O'), ('russia', 'LOCATION'), ('dlc', 'O'), ('resurrected', 'O')

Table 1: Results segmentation Stanford NER; the results are grouped by (segment, label). 'O' denotes that no corresponding class has been found.

part-of-speech tagging and sentiment analysis. For the detection of possible relevant segments, the noun phrase extraction funcionality was used. Please see table 2 for the results of the experiment.

Duration	0.0177 seconds
Sample output	'loss crimea eastern provinces consequence', 'violent overthow', 'government bombing civilians regard post', 'govt illegitimate europe', 'sufficient resources', 'russia dlc', 'month hope', 'popular opinion peer pressure', 'eager fight ukraine', 'ethiopia starvation', 'starvation hoa', 'day night un sanction eritrea ethiopia lunatic', 'president zelensky morning', 'military support ukraine', 'aircraft defence'

Table 2: Results segmentation TextBlob

4.1.3 SEDTWik. is the name of the segmentation method proposed by Morabia et al. [11], which uses Wikipedia page titles to detect multi-worded segments in any given string of text. Please see table 2 for the results of the experiment.

Duration	2.5587 seconds
Sample output	'eastern provinces', 'popular opinion', 'peer pressure', 'computer game', 'president zelensky', 'anti aircraft defence', 'drone attack', 'gon na', 'terrorism war', 'time zones', 'utility services', 'hypersonic missiles', 'russian government', 'russian state'

Table 3: Results segmentation SEDTWik

4.2 Grouping

Besides extracting segments from the text provided by Twitter, it is necessary to detect how relevant these segments are to both the search query and the article text provided by the user. Relevancy, in this context, has been determined by the frequency of the occurrence of a specific segment. To increase the likelihood of a segment occurring more frequently, all the segments were stemmed using the Porter Stemmer provided by NLTK [12]. Different NLTK stemmers (e.g. Snoball, Hunspell stemmer) have been tested as well, however no significant difference could be found. Furthermore, since the outputs of various segmentation methods described in section 4.1 are different, it is necessary to evaluate how descriptive and relevant the segments are once they are sorted by frequency. Therefore, the grouping algorithm has been evaluated with both the SEDTWik and TextBlob segments (see table 4 5). Since, the segments provided by Stanford NER were merely one-worded and took relatively long to compute, we proceeded to not evaluate it any further. The sample output as noted in the tables can be understood as ('segment', 'stem of segment', 'frequency of occurrence').

Sample output
('shirt source', 'shirt sourc', 34), ('food crisis whats', 'food crisis what', 11), ('long world', 'long world', 11), ('terrorism russia place', 'terrorism russia plac', 11), ('international stage', 'international stag', 11), ('collective responsibility', 'collective respons', 11), ('monster act', 'monster act', 11), ('russia terrorist state place', 'russia terrorist state plac', 6)

Table 4: Results grouping TextBlob segments

Sample output
('food crisis', 'food crisi', 18), ('collective responsibility', 'collective respons', 11), ('terrorist state', 'terrorist st', 11), ('russia hoax', 'russia hoax', 6), ('ukraine war', 'ukraine war', 6), ('biden administration', 'biden administr', 5), ('gas prices', 'gas pric', 5), ('word year', 'word year', 5), ('ballistic missiles', 'ballistic missil', 5), ('strong reaction', 'strong react', 5), ('dmitry medvedev', 'dmitry medvedev', 4),

Table 5: Results grouping SEDTWik segments

5 DISCUSSION

While evaluating the results of the segmentation algorithms, a variety of observations can be made. First of all, Stanford NER is only able to extract one-worded segments (see table 1. Furthermore, the segmentation speed is relatively slow and only few of the segments actually are classified with a label; words like "Crimea" should have been classified as a location, however, Stanford NER did not recognize it. Regarding the segmentation results of TextBlob (see table 2), the sample output shows that it is able to detect multi-worded segments. Additionally, the segmentation speed is significantly faster compared to the segmentation performed by Stanford NER. The quality of the segments is debatable, however, as some segments (e.g. "loss crimea eastern provinces consequence") may be difficult to interpret. The results of the experiment with SEDTWik (see table 3) indicate a slightly slower segmentation speed compared to TextBlow, but still a significantly faster speed compared to Stanford NER. Furthermore, the amount of detected segments is relatively low compared to the other methods. The interpretability of these segments, however, does seem better compared to TextBlow's segments.

While evaluating the output of the grouping algorithm using the segments from both SEDTWik and TextBlob, a variety of things can be noted. First of all, since the SEDTWik method results in fewer

segmentations, the average frequency of the TextBlob segments is higher. However, the amount of segments that only occur only once is significantly higher in the TextBlob method as well. This may indicate that the TextBlob method introduces more 'noisy', and therefore less relevant, segments. Furthermore, since the TextBlob method extracts more segments, the computation time of the grouping algorithm increases from 0.019 to 0.113 seconds. Finally, looking at the relatedness of the most frequently occurring segments to the original search query, SEDTWik seems to provide more descriptive segments; 'food crisis', seems more descriptive than 'shirt source' with regard to Russia and Ukraine.

Since this paper does not contain a user study, it is difficult to evaluate if the extracted topics are actually novel and relevant to journalists. This introduces the biggest limitation with regard to this study. Since relevancy is currently determined by frequency of the segments rather than the opinion of journalists, it may be difficult to fully answer the research question. Therefore, to evaluate if the extracted topics are actually relevant, a qualitative user study may be required.

6 CONCLUSION

The study conducted in this paper attempts to extract relevant additional topics related to news events from Twitter. Using web harvesting technologies, a smart application was developed that allowed the user to request frequently occurring segments from Tweets based on a search query. Furthermore, partly using the segmentation method proposed by Morabia et al. [11] we were able to extract segments from Tweets in a relatively fast manner. However, since the relevancy of the segments is currently determined by the frequency of the occurrence of a segment, we strongly recommend to conduct a follow-up user study to evaluate the relevancy in a qualitative manner.

REFERENCES

[1] Tweet Archivist. 2022. Tweet Archivist. https://www.tweetarchivist.com/

[2] Jan Boesman, Leen d'Haenens, and Baldwin Van Gorp. 2015. Triggering the News Story. *Journalism Studies* 16, 6 (Nov. 2015), 904–922. https://doi.org/10.1080/1461670X.2014.953783 Publisher: Routledge _eprint: https://doi.org/10.1080/1461670X.2014.953783.

[3] FastAPI. [n. d.]. FastAPI. https://fastapi.tiangolo.com/

[4] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs Sampling. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05* (2005). https://doi.org/10.3115/1219840.1219885

[5] Caroline Fisher. 2018. News Sources and Journalist/Source Interaction. https://doi.org/10.1093/acrefore/9780190228613.013.849 ISBN: 9780190228613.

[6] Followerwonk. 2022. Followerwonk. https://followerwonk.com/

[7] Agnes Gulyas. 2013. The Influence of Professional Variables on Journalists' Uses and Views of Social Media. *Digital Journalism* 1, 2 (June 2013), 270–285. https://doi.org/10.1080/21670811.2012.744559 Publisher: Routledge _eprint: https://doi.org/10.1080/21670811.2012.744559.

[8] Alejandra Hernández-Fuentes and Angeliki Monnier. 2020. Twitter as a Source of Information? Practices of Journalists Working for the French National Press. *Journalism Practice* 14, 8 (2020), 1–18. https://doi.org/10.1080/17512786.2020.1824585 Publisher: Taylor & Francis (Routledge).

[9] Sanja Kapidzic, Christoph Neuberger, Felix Frey, Stefan Stieglitz, and Milad Mirbabaie. 2022. How News Websites Refer to Twitter: A Content Analysis of Twitter Sources in Journalism. *Journalism Studies* 0, 0 (June 2022), 1–22. https://doi.org/10.1080/1461670X.2022.2078400 Publisher: Routledge _eprint: https://doi.org/10.1080/1461670X.2022.2078400.

[10] Josephine Lukito, Jiyoun Suk, Yini Zhang, Larissa Doroshenko, Sang Jung Kim, Min-Hsin Su, Yiping Xia, Deen Freelon, and Chris Wells. 2020. The Wolves in Sheep's Clothing: How Russia's Internet Research Agency Tweets Appeared in U.S. News as Vox Populi. *The International Journal of Press/Politics* 25, 2 (April 2020), 196–216. https://doi.org/10.1177/1940161219895215 Publisher: SAGE Publications Inc.

[11] Keval Morabia, Neti Lalita Bhanu Murthy, Aruna Malapati, and Surender Samant. 2019. SEDTWik: Segmentation-based Event Detection from Tweets Using Wikipedia. *Proceedings of the 2019 Conference of the North* (2019). https://doi.org/10.18653/v1/n19-3011

[12] NTLK. [n. d.]. nltk.stem.porter module. https://www.nltk.org/api/nltk.stem.porter.html

[13] Arthur D. Santana and Toby Hopp. 2016. Tapping Into a New Stream of (Personal) Data: Assessing Journalists' Different Use of Social Media. *Journalism & Mass Communication Quarterly* 93, 2 (June 2016), 383–408. https://doi.org/10.1177/1077699016637105 Publisher: SAGE Publications Inc.

[14] Marcos A. Spalenza, Elias de Oliveira, Leopoldo Lusquino-Filho, Priscila M. V. Lima, and Felipe M. G. França. 2021. Using NER + ML to Automatically Detect Fake News. In *Intelligent Systems Design and Applications (Advances in Intelligent Systems and Computing)*, Ajith Abraham, Vincenzo Piuri, Niketa Gandhi, Patrick Siarry, Arturas Kaklauskas, and Ana Madureira (Eds.). Springer International Publishing, Cham, 1176–1187. https://doi.org/10.1007/978-3-030-71187-0_109

[15] TextBlob. [n. d.]. Simplified text processing. https://textblob.readthedocs.io/en/dev/

[16] David Somers Tom Brearley. 2022. Twitterfall. https://twitterfall.com/

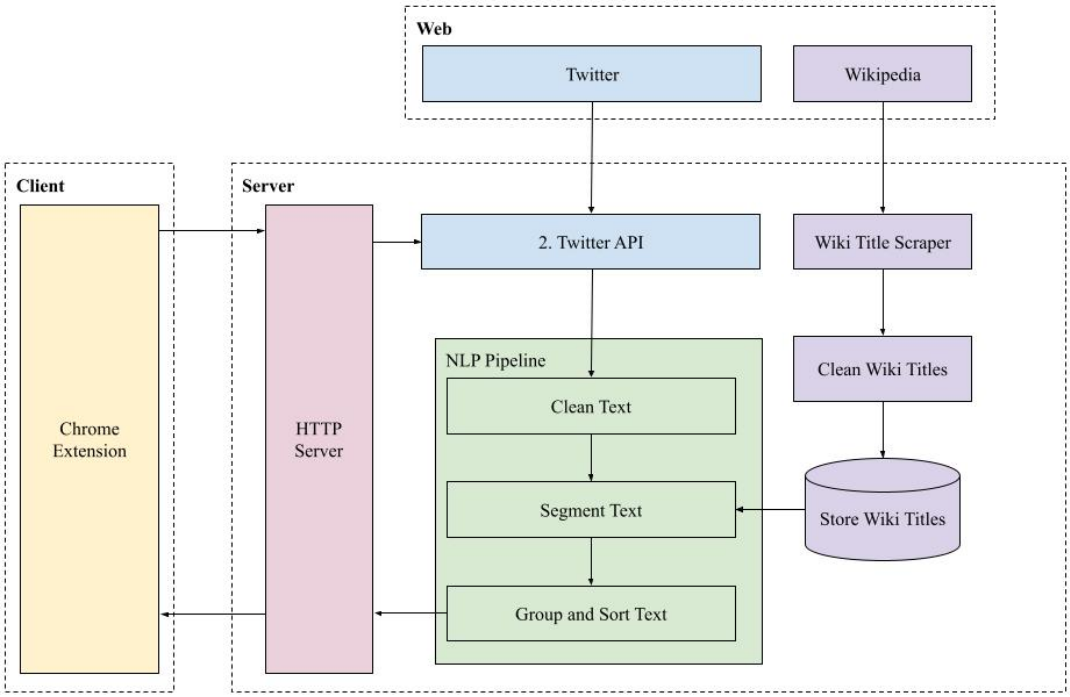
[17] Twitter. 2022. Twitter API. https://developer.twitter.com/en/docs/twitter-api

[18] Jianshu Weng and Bu-Sung Lee. 2021. Event detection in Twitter. *Proceedings of the International AAAI Conference on Web and Social Media* 5, 1 (2021), 401–408. https://doi.org/10.1609/icwsm.v5i1.14102

[19] Wikimedia. [n. d.]. Wikimedia. https://dumps.wikimedia.org/enwiki/latest/

[20] Wikipedia. 2022. Size of wikipedia. https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

A ARCHITECTURE OF THE APP


Web
Twitter
Wikipedia
Client
Server
Chrome Extension
HTTP Server
2. Twitter API
Wiki Title Scraper
NLP Pipeline
Clean Text
Segment Text
Group and Sort Text
Clean Wiki Titles
Store Wiki Titles