Mining Online Sources to Analysis Socio-economic Impacts of Hamun Lake Desiccation

Mahdi Akbari   
Water, Energy and Environmental Engineering Research Unit, Faculty of Technology, University of Oulu, Finland

Gituhub address of the research: <https://github.com/MahdiZizou/Hamun-Lake-NLP-project.git>

Email:

[mahdi.akbari@oulu.fi](mailto:mahdi.akbari@oulu.fi)

ORCID iDs:

Mahdi Akbari: <https://orcid.org/0000-0002-6598-9994>

**Abstract**— Desiccation of Hamun Lake, in Sistan region, has led into unemployment in the region. Sistan region is very controversial region in border of Iran and Afghanistan affected by terrorism activities in recent years. Economic and medical consequences of Hamun Lake desiccation are discussed a lot in literature and media of Iran. Social media in crisis situations, such as environmental disasters, have been recognized by scholars and practitioners as key communication channels that can complement traditional channels. However, there is limited empirical examination from the user perspective of the functions that social media play and the factors that explain such uses. In this study we examine Twitter use to study how Twitter users have reacted to desiccation of Hamun Lake. We analyzed frequency of words in tweets to gain insight on what aspect of Hamun Lake is more important for users. Also, we applied some other Natural Language Processing techniques such as sentiment analysis, lexical categorization and semantic analysis. The results showed that twitter users reacted to Kamal-Khan dam inauguration majorly because this dam can affect Hamun Lake considerably by cutting inflow to this lake. Also, sentiment analysis of tweets revealed that negative feeling is more probable in retrieved tweets related to Hamun Lake and Sistan Region. Lexical categorization of tweets showed how economic consequences of Hamun lake desiccation is reflected in tweets. Finally, semantic analysis of online sources using Google search engine showed that “Desiccation of Hamun Lake”, “Fishing halt” and “Dust in Zabol” has most similarity using WebJaccard similarity index. This similarity shows how Hamun Lake desiccation has led into dust problem in Zabol, the closest city to Hamun Lake, and unemployment in the Sistan region.

**Keywords**— *Twitter API, Google API, NLTK, WordCloud, SentiStrength, Empath Client, WebJaccard Similarity*

# Introduction

Desiccation of water bodies due to drought or anthropogenic activities is a widespread phenomenon such as Caspian Sea Marginal Gulfs [1] and Aral Sea [2] in Eurasia, and Hamun Lake [3] and Urmia Lake [4] in Iran. This phenomenon has environmental and economic consequences.

Hamun Lake is located on the Iran-Afghanistan border in the Sistan region (Figure 1-a). This lake is consisting of 4 water bodies known as Hamun-i Puzak, Hamun-i Sabari, Hamun-i Hirmand and Gaud-i Zirreh (Figure 1-a). They are in Helmand Basin which is a transboundary basin and the most important river flowing to these water bodies is Hirmand (or Helmand) River. Sistan region has four cities and 980 villages, with a population of more than 400,000 [3]. Hamun Lakes are the biggest fresh lake in all over the Iran platue and are identiﬁed in the Ramsar Convention [5] which are very important for the economy and environment of the region [3]. The majority of settlements around Hamun Lakes are in Iran [6], majorly in Zahedan and Zabol cities (Figure 1-b). Therefore, desiccation of Hamun Lakes affects this country more than Afghanistan. For example, Kamal-Khan Dam (Figure 1-b) made a conflict between Iran and Afghanistan in recent years. Iran claims that this Dam will affect Hamun Lakes considerably [7]. Annual inflow from Hirmand River to Iran is shown in Figure 1-c. we can observe that recently annual inflow average has decreased from 3600 to 1900 MCM. In 1970s, a treaty on Hirmand River monthly inflow to Iran, known as Water Protocol (Figure 1-d), was signed between Iran and Afghanistan in which 820 Million Cubic Meter (MCM) (Figure 1-d) annually inflow to Iran was guaranteed by Afghanistan government [8]. This protocol is the only official document between two governments on Hirmand River inflow [9].

In the Sistan region, life quality of 400,000 people including economy and medical aspects are depended on Hirmand River inflow and Hamun Lakes [10]. This region has attracted scientiﬁc interest because of being a major dust source in southwest Asia [11]. Zabol as one the most important cities in the Sistan region and most close one to Hamun Lakes (Figure 1-b) had the most polluted air of the world in year 2012 where mean annual and [12]. A study showed that 63% of the people in Zabol suffer from respiratory diseases, whereas the health damage and medical costs for patients exceeded 166.7 million U.S. Dollars during the period 1999–2004. [13]. In Zabol city, from September 2010 to July 2011 (316 days), only 21 days (less than 7 percent) was in healthy condition [3]. In 1977 more than 55 percent of Sistan people in Iran were working in agricultural sector but this ration has reduced to less than 22 percent in 2015 due to drought periods [14]. Drought has negatively impacted ﬁsheries which have been brought to a halt [3] and caused high unemployment in Sistan region. Unemployment is one of the most important reasons of terrorism activities in Middle Eastern and North African known as MENA [15]. For example, Grishk Dam (shown in Figure 1-a) had two damages which would cost 3 million US dollars to repair by a group of twenty Taliban fighters in 2005.

Social media in crisis situations, such as natural disasters, have been recognized by scholars and practitioners as key communication channels that can complement traditional channels [16]. Considering that there is still a lack of clarity about the role of traditional media in crisis and risk communication [17], it comes as no surprise that there is even more limited understanding about the role of social media in these contexts [18]. Studies in this area mostly deal with the use of social media from an organizational perspective [19]. Such studies have documented cases of effective and ineffective uses of social media in crisis, such as a university response during an earthquake [20], or in global issues such as climate change. This stream of research has led to the development of best practices for organizations from a public relations perspective, such as: communicate quickly, be credible, be accurate, be simple, be complete, and communicate broadly [21]. Similarly, governmental organizations have realized the potential of social media in dealing with crises. For example, the Organization for Economic Cooperation and Development (OECD) and the U.S. Congress have recently developed reports from their institutional perspectives outlining beneﬁts and challenges of social media for crisis managers. Studies of social media use by lay populations, however, remain scarce. We argue that there is an equally pressing need to understand social media use during natural disasters from the perspective of ordinary users and public communicators, consistent with how much the audience has changed. An important body of research in information and computational sciences has explored this area [22], while researchers are only starting to investigate the implications of social media in disaster relief situations [16].

In this study we investigated Tweets to see how people react to Hamun Lake desiccation in order to highlight the most important effects of this environmental disaster. In this research we examined text mining techniques such as text categorization, sentiment analysis, semantic similarity, etc. to develop some information of how people feel about Hamun Lake desiccation based online on text resources. Using statistical analysis, we determined most frequent words of tweets and WebJaccard as an index to show similarity to understand what main consequences of Hamun Lake desiccation are. Proposed approach by this research can help to have more insights about what is happening in Sistan society to cover lack of social data in the region.

A picture containing chart

Description automatically generated

Figure 1. Studied area: a) Hamun Lakes and Helmand Basin with the location of dams, inflow gauges and cities of the basin, b) DEM of Hamun Lakes and close water bodies to Hamun Lakes, c) Annual inflow of Hirmand River to Iran from 1946 to 2016 and d) Water Protocol on Hirmand River inflow to Iran

# Methodology

This study is consisting of two sections: section 1 uses Twitter Application Programming Interface (API) to retrieve text data from tweets and section 2 uses Google search API to extract information from online websites. The flowchart of each section is separately depicted in Figure 2-a and b respectively. The implementation of each section is separately presented in appendices sections. Description of each step is presented below.

## Twitter API

Twitter API module in python known as Tweepy [23] was used in this study. To communicate with the API, we ﬁrst registered a developer account on [https://developer.twitter.com/en](https://developer.twitter.com/en%20) to get tokens and secrets. We then used Tweepy module in python as the tool to communicate and access the API to clone tweets from users. In comparison with other libraries such as Twint, rearchtweets and GetOldTweets, Tweepy has more sufﬁcient documentations with top-notch tutorials and easy guides for newbies. As also being known as a power tool, Twint, has overcame tweepy limitation in number of cloned tweets (3200 tweets per user). However, its minimal documentation was a struggle for us to approach in this way.

In this study, we utilized the user timeline API by conﬁguring the extended mode to get full tweets message. At the same time, all retweeted posts were also excluded from searching result to get a more accurate dataset. The working flow is depicted in Figure 2-a and more details on implementations are explained in Appendix A and B.

## Google search API

Google does give the opportunity to scrape information. However, whatever scraping that would be done has to be through an API. In order to be able to use the Google search API, we would be needing a Custom Search Engine ID. Therefore, we would have to create a Custom Search Engine first (<https://cse.google.com/cse/all>). Custom Search Engine provides API key and search engine ID which should be used to benefit from Google search API (see Appendix C for more detail). The working flow of this section is depicted in Figure 2-b.

## Datasets

For section 1 (Twitter API) of this study, we analyzed content of tweets shared on Twitter about Hamun Lake and Sistan region. First, we build a positive dataset () and negative dataset () from tweets in sense of users’ emotions showed in tweets by defining queries based on some keywords (Table 1). Then, using Tweepy in python we retrieved tweets from twitter. It should be noted that because most tweets related to studied topic are in Persian (popular language in Iran and Afghanistan around Hamun Lakes), so keywords were defined in Persian. Then, we translated all tweets into English using Google Translate module in python known as googletrans [24]. Based on defined queries by OR operator between keywords of each datasets (Table 1), excluding retweets, and were built (more detail in Appendix A).

Finally, datasets of section 2 (Google search API), we used Google search API in python to count number of pages containing queries shown in Table 2 (more detail in Appendix C).

## Preprocessing of Tweets

Persian alphabet is completely different than Latin alphabet so if any Latin character is used in Persian tweets, it should be omitted. After building and stored here in a variable named as tweets\_data, we first deleted any Latin character in tweets to make clean data set named as cl\_tw. After this step, we translated cleaned tweets in first step from Persian (or Farsi) to English to make cl\_tw\_trs variable. Translator is imported from googletrans module. Finally, in translated tweets, there should not be any Persian alphabet, so we again filter Persian letters from translated tweets and made cl\_tw\_trs\_cl variable. This process is applied on all tweets in in a for loop and stored finally in tw\_final list.

1. tw\_final = []
2. for tweet in tweets\_data:
3. cl\_tw = "".join([char for char in tweet if char not in 'qwertyuiopasdfghjkl\:zxcvbnm/ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890@#$%^&\*\_=-\_,:.;?!']).strip()
4. cl\_tw\_trs = translator.translate(cl\_tw, src='fa', dest='en').text
5. cl\_tw\_trs\_cl = "".join([char for char in clean\_tweet\_trans if char not in ',:;،ضصثقفغعهخحجچپگکمنتالبیسشظطزرذدئوې']).strip()
6. tw\_final.append(cl\_tw\_trs\_cl)

## Tweets tokenization and detecting most frequent words

After building cleaned and translated tweets for and (tw\_final as mentioned before), we tokenized them. MWETokenizer is imported from nltk module in python [25].

1. tk = MWETokenizer(mwe\_words)
2. tokenized\_tweets = []
3. for tw in tw\_final:
4. tw = tweet.lower()
5. tok =

tk.tokenize(tw.split())

1. tokenized\_tweets.append(tok)

In this step, we deleted stop words and then drew histogram of the most common words for each of mentioned datasets.

1. stop\_words = set(stopwords.words('english'))
2. filt\_tw = []
3. for tw in tokenized\_tweets:
4. filtered = [w for w in tw

if not w in

stop\_words]

1. filt\_tw.append(filtered)

## Part of speech (POS) tagging

After tokenizing all tweets and deleting stop words, we tagged them all by POS tagger in NLTK module in python. Finally, using tokenized and POS tagged words from tweets in and , we determined most frequent noun and verbs of each data set.

  Table 1. Keywords of Queries to collect positive and negative tweets related to Hamun lake

| Dataset | Key word | Key word Translation |
| --- | --- | --- |
|  | زیبایی سیستان | The beauty of Sistan |
| زیبایی زابل | The beauty of Zabol |
| توریسم سیستان | Sistan Tourism |
| دریاچه هامون زیبا | Beautiful Hamun Lake |
| هامون پر آب | Full-of-water Hamun |
|  | تشنگی دریاچه هامون | Thirsty Hamun |
| گرد و غبار زابل | Dust in Zabol |
| بیکاری سیستان | Sistan unemployment |
| خشکسالی سیستان | Sistan drought |
| تشنگی سیستان | Thirsty Sistan |
| تشنگی زابل | Thirsty Zabol |

## Graphical representation of datasets by WordCloud

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites [26]. We used this technique in our research to have a graphical representation for and .

## Sentiment analysis of datasets by SentiStrength

SentiStrength estimates the strength of positive and negative sentiment in short texts, even for informal language. It has human-level accuracy for short social web texts in English, except political texts. More negative scores from SentiStrength reports more negative feeling in the text [27]. We applied this technique in our retrieved data set from twitter API. We expect that negative score be given to and positive score to .

## Lexical categorization by Empath

Empath is a tool that can generate and validate new lexical categories on demand from a small set of seed terms (like “bleed” and “punch” to generate the category violence). Empath draws connotations between words and phrases by deep learning a neural embedding across more than 1.8 billion words of modern ﬁction [28]. We applied this tool on and to see what main lexical categories of these datasets are.

## Semantic analysis by WebJaccard similarity index

Page counts for the query P AND Q, can be considered as an approximation of co-occurrence of two words P and Q on the Web. We modify popular co-occurrence index of Jaccard to compute semantic similarity using page counts retrieved from Google search API in python [29]. For the rest of this paper we use the notation H(P ) to denote the page count for the query P in Google search engine. WebJaccard coefﬁcient (WJ) between words P and Q is deﬁned by:

*WJ(P,Q)*, 

here, PQ denotes the conjunction query P AND Q. Given the scale and noise in web data, it is possible that two words may appear on some pages purely accidentally. In order to reduce the adverse effects attributable to random co-occurrences, we set the WJ to zero if the page count for the query PQ is less than a threshold c.

All above-mentioned steps are taken using python programming language. All codes are available in Github page of the project: <https://github.com/MahdiZizou/Hamun-Lake-NLP-project.git>.

# Results and Discussions

We could find very low numbers of tweets for (less than 10), but included more than 300 tweets by defined query. Most common words for are “water”, “Alborz” and “upright”, and most common words for are “dam”, “Kamal-Khan”, “third phase” and “water” shown in Figure 3.

The low number of positive tweets related to Hamun Lake and Sistan region, shows how negatively this region is affected by Hamun Lake desiccation. Having “water” as most frequent word in both and shows that water is very important issue in the region and most people talk about it in their tweet regardless of showing positive or negative feelings. In Iran, dams are one of the most important causes of environmental issues like Urmia Lake desiccation [30]. This is reflected in because “dam” word is very popular among tweets. Another important point in , is presence of “Kamal-Khan” and “third phase”. As shown in Figure 1-a, on Hirmand River we have 4 dams and most close one to Iran border is Kamal-Khan dam. This dam is in third phase of construction and has inaugurated experimentally. This fact is also affected in tweets of considerably.

Also, after POS tagging of all words of tweets in each positive and negative datasets, most common nouns and verbs of and are plotted in Figure 4. Ashraf Ghani, president of Afghanistan, has recently visited Kamal-Khan Dam and wished it’s operation happens soon [31]. The verbs “open” and “inaugurate” and noun “Ghani” are popular in because of mentioned fact about Kamal-Khan dam.

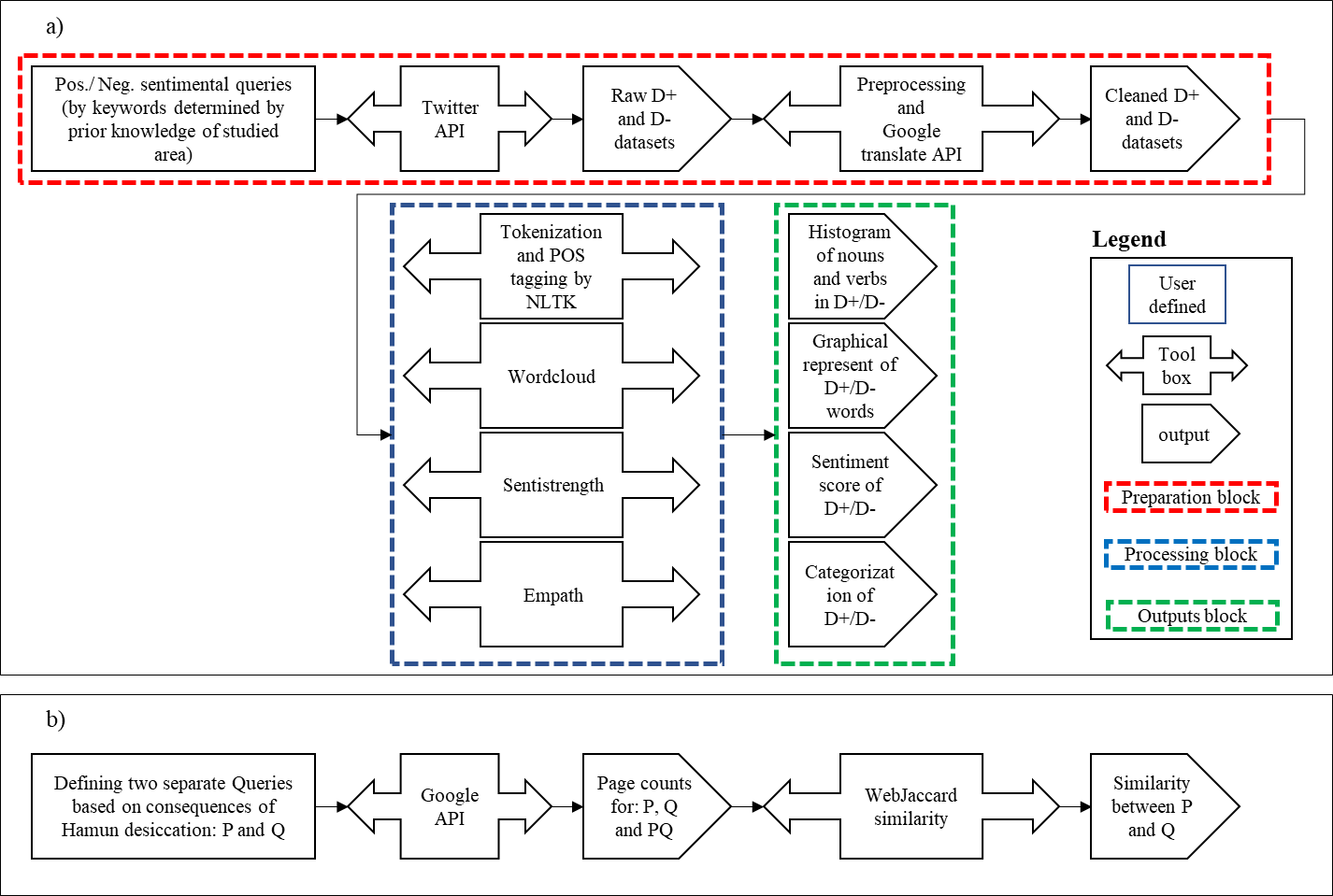


Figure 2. Flowchart of sections in this study a) section 1 by Twitter API and b) section 2 by Google search API

|  |  |
| --- | --- |
| a)  b) | (b) |

Figure 3. most common words in a) D+ and b) D-

Based on graphical representation of most common words from Word Cloud (Figure 5), positive words like “upright”, ”peace”, “beauties”, “full”, “celebration” and “fairness” are popular in . However, words “Kamal-Khan”, “Ashraf-Ghani”, “inaugurated”, “” and “third phase” can reflect negative feeling of people, majorly Iranian, in about damming of Hirmand River by Afghanistan.

After applying SentiStrength on each of and , overall score of them found to be 0 and -1 respectively (Figure 6). It is consistent with what we expected. It should be noted that we expect to have positive value for but its score is 0 which shows even in positive dataset, user do not show very positive feelings.

Furthermore, applying Empath showed that main lexical categories for are “Beach”, “Water”, “Swimming”, “Sailing”, “Cleaning” and “Ocean”. However, major categories for are “Negative emotion”, “Economics”, “Poor”, “Aggression”, “Water” and “Government”. The lexical categories for both and also includes “water” which shows how water resource management problem is crucial for Hamun Lake and Sistan region. Also, negative economic consequences of Hamun Lake are reflected clearly in determined categories of by Empath module.

|  |
| --- |
| a)  b)  c)  d) |

Figure 4. Most common nouns in a) D+ and b) d- with most common verbs in c) d+ and d) d-

|  |
| --- |
| a)  b) |

Figure 5. output of word cloud for a) D+ and b) D-

Figure 6. Sentistrength score for D+ and D-

The result of WJ similarity index for different queries (shown in Table 2) shows that most similar queries using Google search engine are “Desiccation of Hamun Lake”, “Fishing halt” and “Dust in Zabol”. This is consistent with our prior knowledge from Sistan region because two important main consequences of Hamun Lake desiccation are destruction of fishing industry of the region and long dust storms of the region especially in Zabol city [3], [10].

Table 2. WebJaccard coefﬁcient (WJ) for various queries

|  |  |  |
| --- | --- | --- |
| **Query** | **Translation** | **WJ(%)** |
| **Q**=خشکی دریاچه هامون | Desiccation of Hamun Lake | 0.53 |
| **P**=بیکاری سیستان و زابل | Unemployment in Sistan and Zabol |
| **Q**=خشکی دریاچه هامون | Desiccation of Hamun Lake | 20.6 |
| **P**=توقف ماهیگیری سیستان | Fishing halt |
| **Q**=خشکی دریاچه هامون | Desiccation of Hamun Lake | 2.9 |
| **P**=کمبود آب شرب | Domestic water deficit |
| **Q**=خشکی دریاچه هامون | Desiccation of Hamun Lake | 20.3 |
| **P**=گرد و غبار زابل | Dust in Zabol |
| **Q**=خشکی دریاچه هامون | Desiccation of Hamun Lake | 5.9 |
| **P**=کاهش گردشگری سیستان و زابل | Decrease in tourism in Sistan and Zabol |

Main limitation of this study was the small size of datasets because we had to use Persian language in Twitter API to cover more tweets. Sistan region in developing region and internet access is an issue in that region. Majority of population in that region are in Iran not in Afghanistan. Twitter is filtered by Iran government which makes possibilities of tweets from that region lower. Although we also defined some English keywords to make and as largest as possible, we could only retrieve 327 tweets. Therefore, inferences from retrieved datasets can be more robust if size of increases. This fact is due to some reasons: 1) the Hamun Lake desiccation is a disaster in Sistan region and inherently it is not likely to have positive tweets about it, 2) Hamun Lake is close to border of Iran and Afghanistan. Sistan Region in both countries is very undeveloped region and affected people from Hamun Lake desiccation do not have access to internet commonly and 3) even in other developed citied of Iran such as Tehran, Hamun Lake desiccation is not hot topic because of security issues related to Sistan area which is affected by terrorist activities a lot in recent years. However, another desiccated lake in north west of Iran, Urmia Lake, received more attentions in media and academic communities compared to Hamun Lake.

# Conclusion

In this research we found that Natural Language Process techniques can help to gain insight about undeveloped regions affected by environmental disasters like water body desiccation covering the lack of in-situ data by state-of-art approach to use twitter API. Economy of Sistan region is deeply affected by Hamun Lake desiccation and this fact was also shown by Empath lexical categorization. Some important facts of the region like inauguration of Kamal-Khan dam by president of Afghanistan was also found by analysis of most common words in tweets. Finally, from sentiment analysis we found that “Hamun Lake” and “Sistan region” as main keywords of our queries for retrieving tweets, are more probable to show negative feeling among twitter users than positive sentiment. Finally, the most important consequences of Hamun Lake desiccation based on literature was found by WJ index which are Zabol dust and fashioning industry ruin.

##### **Acknowledgment**

The authors are thankful to Eng. Nabil Arhab who provided insights and expertise that greatly assisted the research.

##### **References**

[1] M. Akbari *et al.*, “Vulnerability of the Caspian Sea shoreline to changes in hydrology and climate,” *Environ. Res. Lett.*, 2020.

[2] A. AghaKouchak *et al.*, “Aral Sea syndrome desiccates Lake Urmia: call for action,” *J. Great Lakes Res.*, vol. 41, no. 1, pp. 307–311, 2015.

[3] A. Rashki, D. G. Kaskaoutis, C. J. d. W. Rautenbach, P. G. Eriksson, M. Qiang, and P. Gupta, “Dust storms and their horizontal dust loading in the Sistan region, Iran,” *Aeolian Res.*, vol. 5, pp. 51–62, 2012.

[4] M. Akbari, A. Torabi Haghighi, M. M. Aghayi, M. Javadian, M. Tajrishy, and B. Kløve, “Assimilation of satellite-based data for hydrological mapping of precipitation and direct runoff coefficient for the Lake Urmia Basin in Iran,” *Water*, vol. 11, no. 8, p. 1624, 2019.

[5] A. Moghaddamnia, M. G. Gousheh, J. Piri, S. Amin, and Dja. Han, “Evaporation estimation using artificial neural networks and adaptive neuro-fuzzy inference system techniques,” *Adv. Water Resour.*, vol. 32, no. 1, pp. 88–97, 2009.

[6] M. Pesaresi and S. Freire, “GHS Settlement grid following the REGIO model 2014 in application to GHSL Landsat and CIESIN GPW v4-multitemporal (1975-1990-2000-2015),” *Eur. Comm. Jt. Res. Cent.*, 2016.

[7] BBC, “Afghanistan: Dam construction in Afghanistan should not be a concern for Hassan Rouhani,” 2017.

[8] Ministry of Enegry (MoE) and Yekom Consultancy Company, “Update of Sistan River studies (hydrological report),” 2014.

[9] ICANA, “25% migration of Sistan people due to drought,” 2015. [Online]. Available: khabaronline.ir/news/587938. [Accessed: 25-Aug-2020].

[10] A. Rashki, D. G. Kaskaoutis, A. S. Goudie, and R. A. Kahn, “Dryness of ephemeral lakes and consequences for dust activity: The case of the Hamoun drainage basin, Southeastern Iran,” *Sci. Total Environ.*, vol. 463–464, pp. 552–564, 2013.

[11] A. S. Goudie and N. J. Middleton, *Desert dust in the global system*. Springer Science & Business Media, 2006.

[12] WHO, “Global Health Observatory data repository,” *World Health Organization*, 2016. [Online]. Available: https://apps.who.int/gho/data/view.main.AMBIENTCITY2016?lang=en. [Accessed: 08-Nov-2020].

[13] A. Miri, H. Ahmadi, A. Ghanbari, and A. Moghaddamnia, “Dust storms impacts on air pollution and public health under hot and dry climate,” *Int J Energy Env.*, vol. 2, no. 1, pp. 101–105, 2007.

[14] Ministry of Cooperatives Labour and Social Welfare Iran, “Employment status of Sistan and Baluchestan province,” 2017.

[15] A. Bagchi and J. A. Paul, “Youth unemployment and terrorism in the MENAP (Middle East, North Africa, Afghanistan, and Pakistan) region,” *Socioecon. Plann. Sci.*, vol. 64, no. December 2017, pp. 9–20, 2018.

[16] B. Takahashi, E. C. Tandoc, and C. Carmichael, “Communicating on Twitter during a disaster: An analysis of tweets during Typhoon Haiyan in the Philippines,” *Comput. Human Behav.*, vol. 50, pp. 392–398, 2015.

[17] A. Af Wåhlberg and L. Sjöberg, “Risk perception and the media,” *J. Risk Res.*, vol. 3, no. 1, pp. 31–50, 2000.

[18] A. R. Binder, “Figuring out #fukushima: An initial look at functions and content of US twitter commentary about nuclear risk,” *Environ. Commun.*, vol. 6, no. 2, pp. 268–277, 2012.

[19] S. R. Veil, T. Buehner, and M. J. Palenchar, “A Work-In-Process Literature Review: Incorporating Social Media in Risk and Crisis Communication,” *J. Contingencies Cris. Manag.*, vol. 19, no. 2, pp. 110–122, 2011.

[20] N. Dabner, “‘Breaking Ground’ in the use of social media: A case study of a university earthquake response to inform educational design with Facebook,” *Internet High. Educ.*, vol. 15, no. 1, pp. 69–78, 2012.

[21] K. Freberg, K. Saling, K. G. Vidoloff, and G. Eosco, “Using value modeling to evaluate social media messages: The case of hurricane irene,” *Public Relat. Rev.*, vol. 39, no. 3, pp. 185–192, 2013.

[22] A. Acar and Y. Muraki, “Twitter for crisis communication: Lessons learned from Japan’s tsunami disaster,” *Int. J. Web Based Communities*, vol. 7, no. 3, pp. 392–402, 2011.

[23] J. Roesslein, “Tweepy: Twitter for Python!,” *URL https//github.com/tweepy/tweepy*, 2020.

[24] S. Han, “googletrans: googletrans for Python!,” *URL https//github.com/ssut/py-googletrans.git*, 2020.

[25] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. “ O’Reilly Media, Inc.,” 2009.

[26] S. Kadam, “Generating Word Cloud in Python,” 2020. [Online]. Available: https://www.geeksforgeeks.org/generating-word-cloud-python/. [Accessed: 11-Oct-2020].

[27] SentiStrength, “SentiStrength.” [Online]. Available: http://sentistrength.wlv.ac.uk/. [Accessed: 11-Oct-2020].

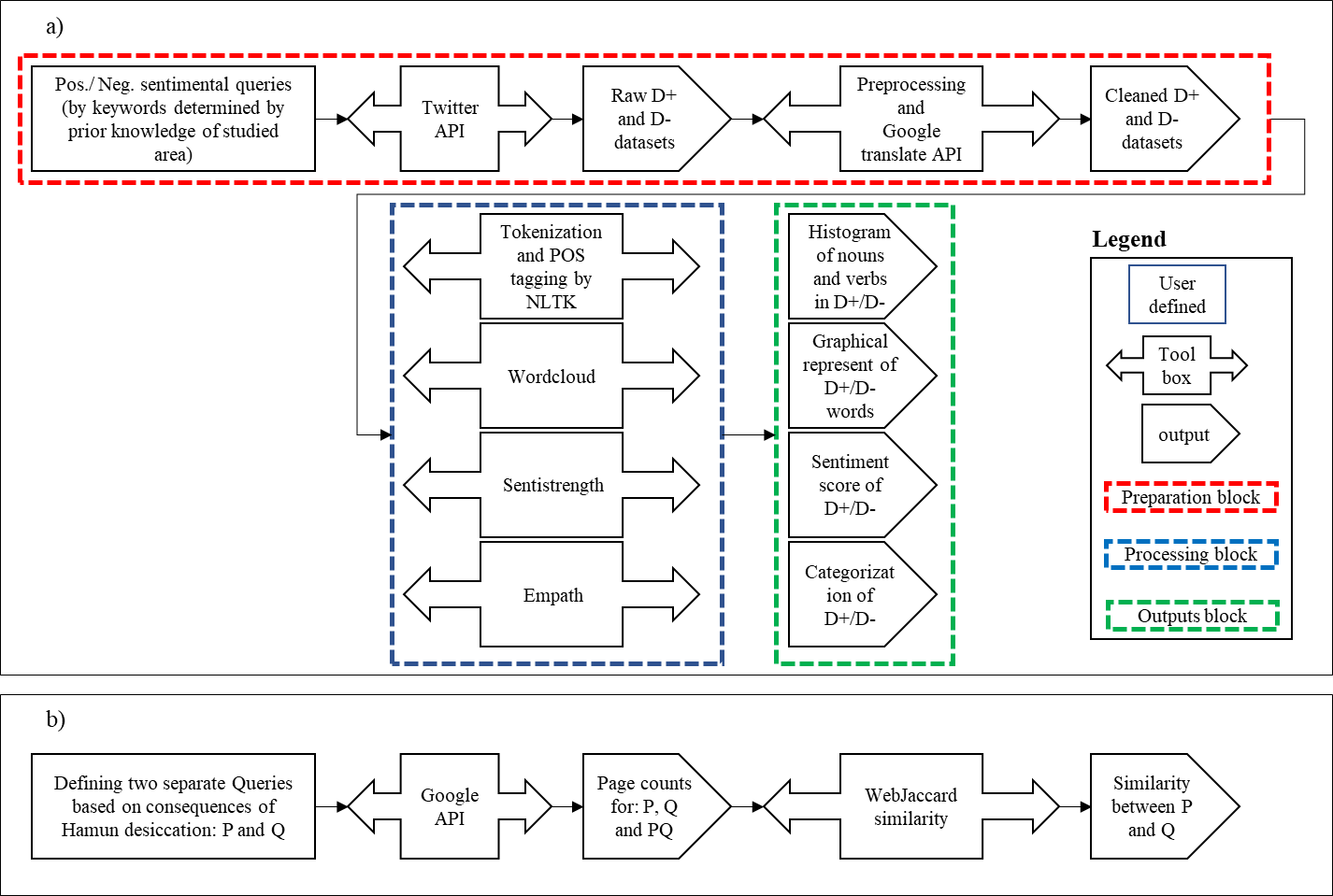
[28] E. Fast, B. Chen, and M. S. Bernstein, “Empath: Understanding topic signals in large-scale text,” *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 4647–4657, 2016.

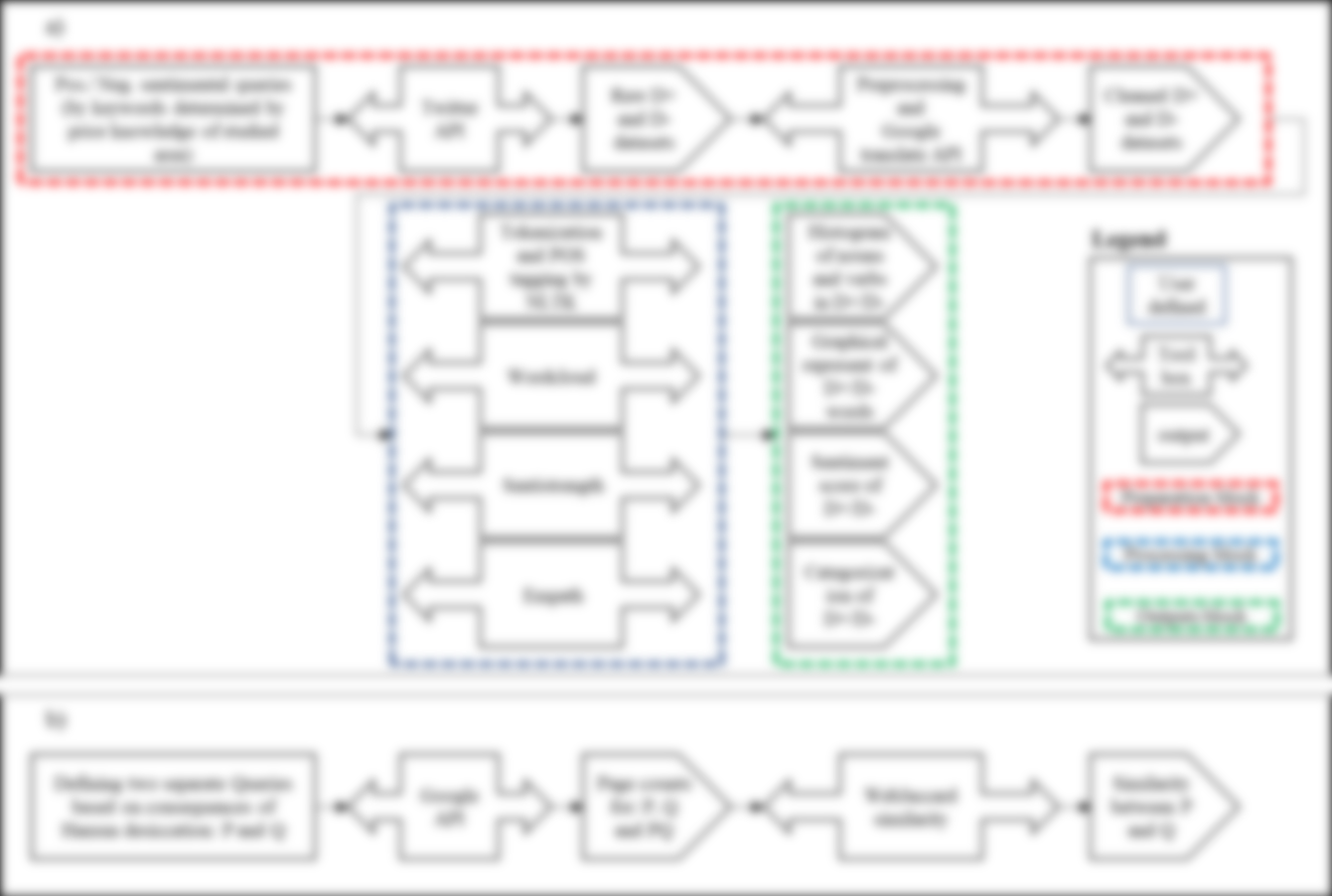
[29] Google, “Python Quickstart for Google API,” 2020. [Online]. Available: https://developers.google.com/docs/api/quickstart/python. [Accessed: 10-Oct-2020].

[30] M. Akbari, A. T. Haghighi, M. M. Aghayi, M. Javadian, M. Tajrishy, and B. Kløve, “Assimilation of satellite-based data for hydrological mapping of precipitation and direct runoff coefficient for the Lake Urmia basin in Iran,” *Water (Switzerland)*, vol. 11, no. 8, 2019.

[31] BBC, “Experimental water intake of Afghanistan’s Kamal Khan Dam began near the border with Iran,” 2020. [Online]. Available: https://www.bbc.com/persian/afghanistan-54854466. [Accessed: 10-Oct-2020].

**Appendix A- Implementation of preparation block of section 1**







Based on keywords shown in Table 1, queries for and are defined (excluding retweets). We defined a function as limit\_handled in which if we pass Tweepy limit, it pauses run for 15 min so that we can again use Tweepy. Using tokens provided for Twitter API after registration, we searched twitter data based by our query. Finally, we save retrieved data as csv files available in Github page of the project.

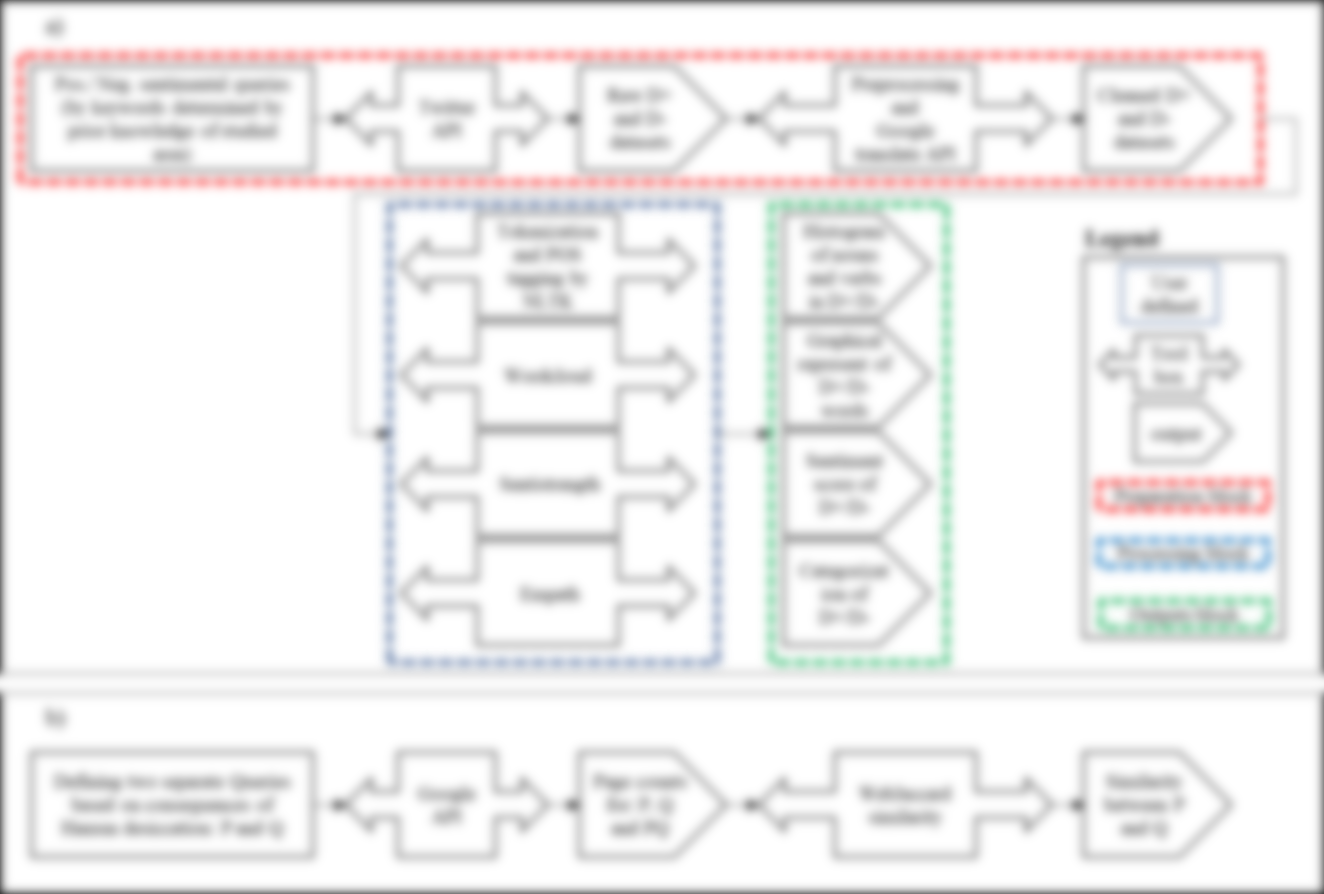
1. query=keyword1+' OR '+keyword2+' OR '+keyword3+' OR ' +keyword4+' OR '

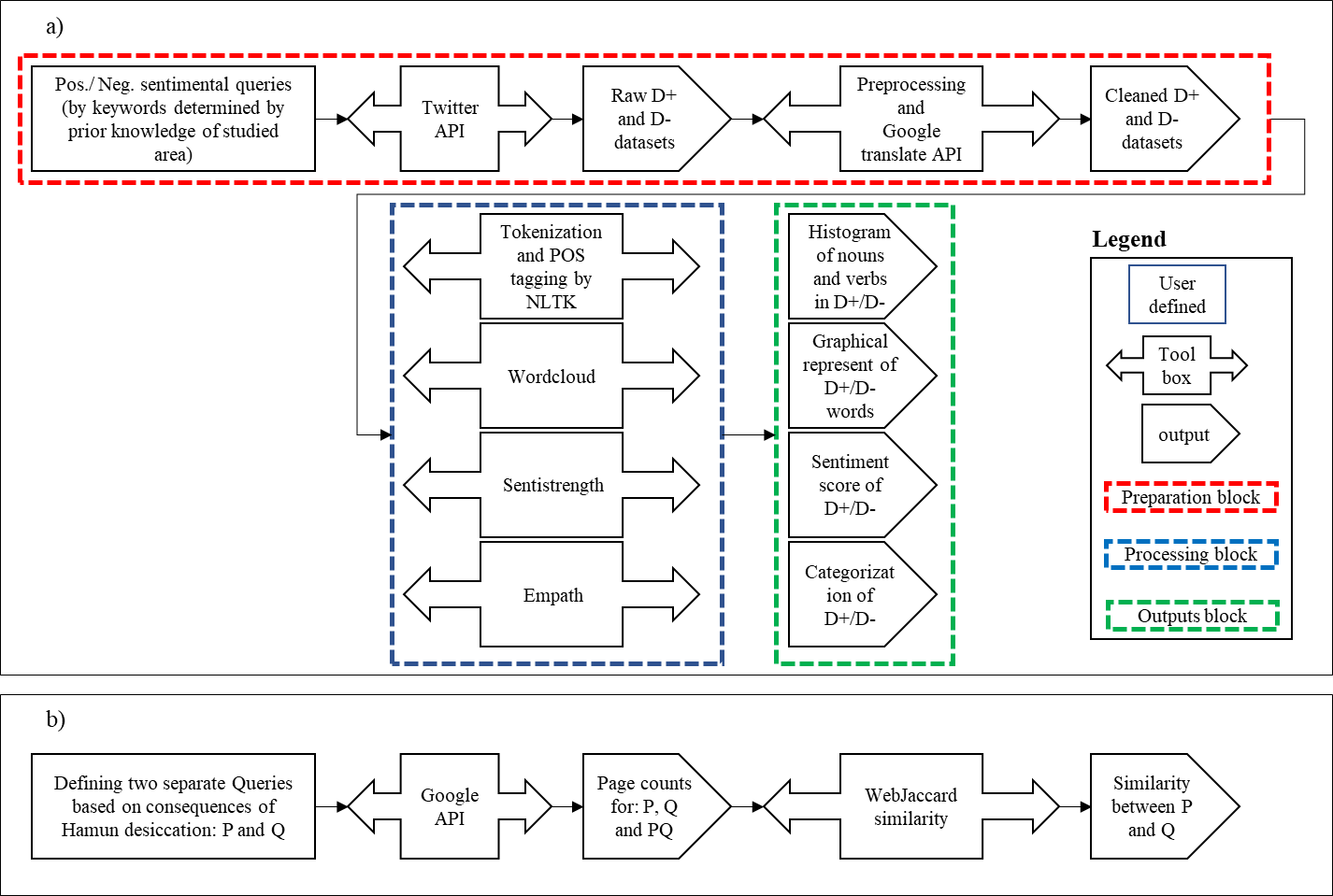
+keyword5 +' OR ' +keyword6+' OR ' +keyword7+' OR ' +keyword8+'

-is:retweet'

1. import tweepy
2. import pandas as pd
3. import googletrans
4. from googletrans import Translator
5. translator = Translator()
6. import time
7. import re
8. def limit\_handled(cursor):
9. while True:
10. try:
11. yield cursor.next()
12. except tweepy.RateLimitError:
13. time.sleep(15 \* 60)
14. except StopIteration:
15. print('Query is done :)')
16. break
18. consumer\_key='???'
19. consumer\_secret='???'
20. token\_access='???'
21. token\_secret='???'
22. auth = tweepy.OAuthHandler(consumer\_key,consumer\_secret)
23. auth.set\_access\_token(token\_access,token\_secret)
24. api = tweepy.API(auth, wait\_on\_rate\_limit=True)
26. tweets\_data=[]
28. for i,status in enumerate(limit\_handled(tweepy.Cursor(api.search, q=query).items())):
29. status\_dict=dict(vars(status))
30. tweet=status\_dict['text']
31. tweets\_data.append(tweet)
33. df = pd.DataFrame(tweets\_data)
34. query\_en=translator.translate(keyword, dest='en').text
35. name= query\_en + '\_raw\_' + str(len(tweets\_data)) + '.csv'
36. df.to\_csv(name, sep='\t', header=False, encoding='utf-8-sig')

**Appendix B- Implementation of processing block of section 1**





Here we tagged words and drew histogram of most frequent words. Please note that in methodology section, variable filt\_tw is explained which is tokenized, cleaned and translated tweets from which stop words has be deducted.

1. import tweepy
2. import nltk
3. from nltk.tokenize import TweetTokenizer
4. from nltk.tokenize import MWETokenizer
5. from nltk.probability import FreqDist
6. from nltk.corpus import stopwords
8. chain\_object = itertools.chain.from\_iterable(filt\_tw)
9. flattened\_tweets = list(chain\_object)
11. POS\_tweets=nltk.pos\_tag(flattened\_tweets)
13. nltk.FreqDist(tag for (word, tag) in POS\_tweets)
15. flattened\_tweets\_noun=[word for (word,tag) in POS\_tweets if tag=='NN' or word=='kamal\_khan']
16. flattened\_tweets\_verb=[word for (word,tag) in POS\_tweets if tag=='VB' or tag=='VBD' or tag=='VBN' or tag=='VBG']
18. fdist\_noun = FreqDist(flattened\_tweets\_noun)
19. most20\_noun=fdist\_noun.most\_common(20)
20. fdist\_noun.plot(20, cumulative=False)
22. fdist\_verb = FreqDist(flattened\_tweets\_verb)
23. most20\_verb=fdist\_verb.most\_common(20)
24. fdist\_verb.plot(20, cumulative=False)

 Then we applied Wordcloud on our data sets here:

1. from wordcloud import WordCloud, STOPWORDS
2. import matplotlib.pyplot as plt
3. import pandas as pd
5. tweets\_words = ' '
6. stopwords = set(STOPWORDS)
8. for tweet in tw\_final:
9. tokens = tweet.split()
10. for i in range(len(tokens)):
11. tweets\_words += " ".join(tokens) + " "
13. wordcloud = WordCloud(width=800, height=800,
14. background\_color='white',
15. stopwords=stopwords,
16. min\_font\_size=10).generate(tweets\_words)
18. plt.figure(figsize=(8, 8), facecolor=None)
19. plt.imshow(wordcloud)
20. plt.axis("off")
21. plt.tight\_layout(pad=0)
23. plt.show()

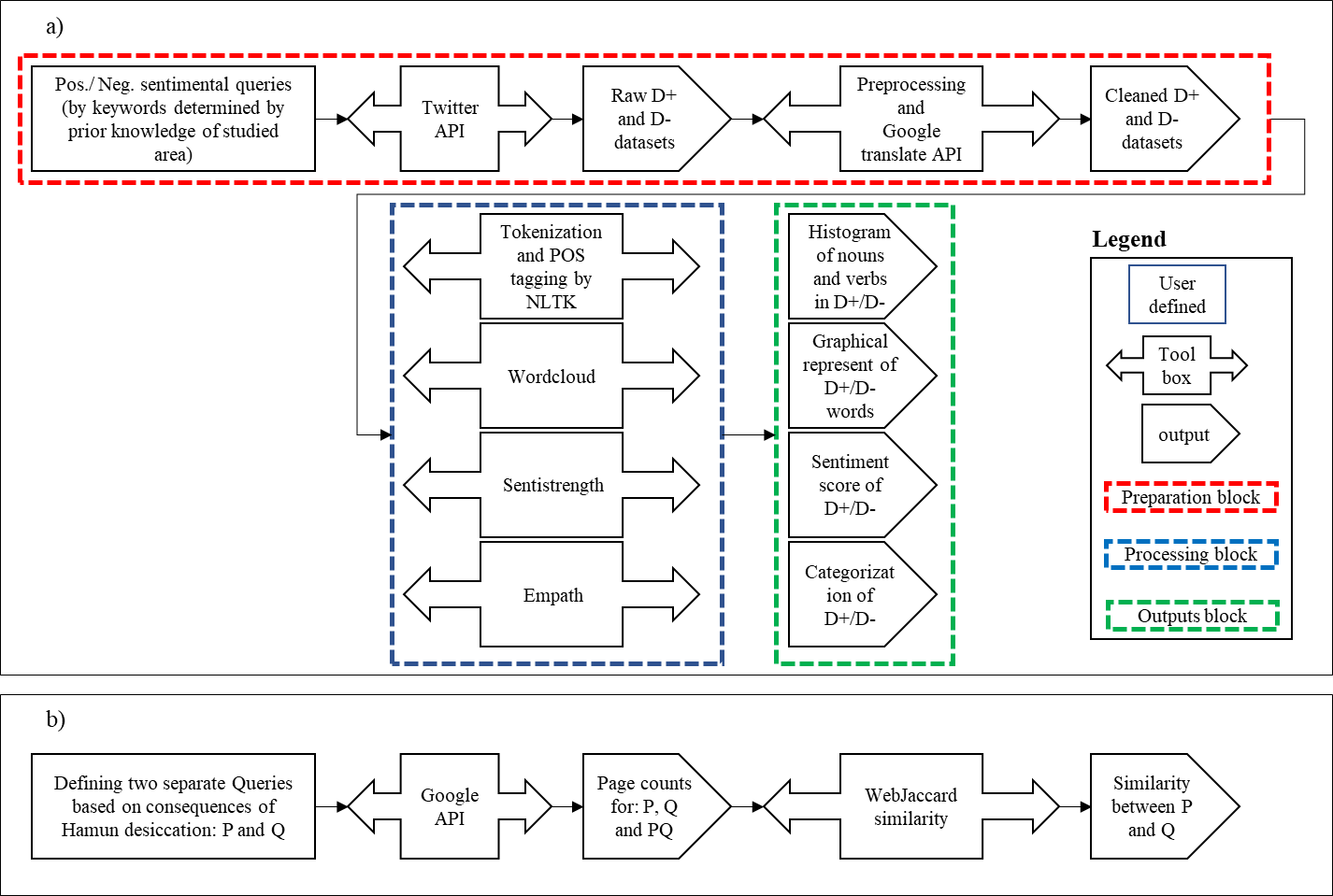
 Here we applied sentistrength on our data set. This module can be applied if java is installed. Also, SentiStrengthData folder and SentiStrength.jar directory should be given. These two files should not be in a same folder so that error is not generated. tweets\_words variable is introduced in above pseudocode box (SentiStrengthData folder and SentiStrength.jar are uploaded in Github page of this project).

1. import sentistrength
2. from sentistrength import PySentiStr
4. senti= PySentiStr()
5. senti.setSentiStrengthPath('directory of SentiStrength.jar???')
6. senti.setSentiStrengthLanguageFolderPath(' directory of SentiStrengthData/ folder???')
7. senti.getSentiment(tweets\_words)

Finally, here we categorized our dataset by Empath:

1. from empath import Empath
2. lexicon = Empath()
4. tweet\_cats=[]
5. for tweet in tw\_final:
6. categories = lexicon.analyze(tweet, normalize=True)
7. cats = [key for key, val in categories.items() if val >= 0.05]
8. tweet\_cats.append(cats)

**Appendix C- Implementation of section 2**



Detail on implementation of searching online websites using Google search API is presented below (for detailed toturial please see <https://linuxhint.com/google_search_api_python/>). query1 (or P) and query2 (or Q) are defined based on Table 2. query12 is defined as shown in line 12 in below pseudocode box. Finally, hits\_query1, hits\_query2 and hits\_query12 are number of pages when we use query1, query2 and query12 respectively. Finally, WebJaccard coefﬁcient is calculated based on equation 1 in line 23 as shown below:

1. from googlesearch import search
2. from googleapiclient.discovery import build
4. def google\_search(search\_term, api\_key, cse\_id, \*\*kwargs):
5. service = build("customsearch", "v1", developerKey=api\_key)
6. res = service.cse().list(q=search\_term, cx=cse\_id, \*\*kwargs).execute()
7. return res
9. my\_api\_key = '???'
10. my\_cse\_id = "???"
11. query12=query1+" "+query2
12. result\_query1 = google\_search(query1, my\_api\_key, my\_cse\_id)
13. result\_query2 = google\_search(query2, my\_api\_key, my\_cse\_id)
14. result\_query12 = google\_search(query12, my\_api\_key, my\_cse\_id)
16. hits\_query1=int(result\_query1['searchInformation']['totalResults'])
17. hits\_query2=int(result\_query2['searchInformation']['totalResults'])
19. hits\_query12=int(result\_query12['searchInformation']['totalResults'])
21. jsim12=hits\_query12/(hits\_query1+hits\_query2-hits\_query12)
23. print(jsim21\*100)