Optimized Web-Crawling of Conversational Data from Social Media and Context-Based Filtering

Annapurna P Patil
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
annapurnap2@msrit.edu

Rajarajeswari S
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
raji@msrit.edu

Gaurav Karkal
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
gauravkarkal@gmail.com

Keerthana Purushotham
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
keerthupuru@gmail.com

Jugal Wadhwa
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
jugaldeepak@gmail.com

K Dhanush Reddy
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
dhanushreddy1014@gmail.com

Meer Sawood
Department of Computer Science and Engineering
Ramaiah Institute of Technology
Bangalore, India
sawoodrocket@gmail.com

Abstract

Building Chatbot's requires a large amount of conversational data. In this paper, a web crawler is designed to fetch multi-turn dialogues from websites such as Twitter, YouTube and Reddit in the form of a JavaScript Object Notation (JSON) file. Tools like Twitter Application Programming Interface (API), LXML Library, and JSON library are used to crawl Twitter, YouTube and Reddit to collect conversational chat data. The data obtained in a raw form cannot be used directly as it will have only text metadata such as author or name, time to provide more information on the chat data being scraped. The data collected has to be formatted for a good use case, and the JSON library of python allows us to format the data easily. The scraped dialogues are further filtered based on the context of a search keyword without introducing bias and with flexible strictness of classification.

1 Introduction

Real-world data remains a necessary part of training system models. The digital streams that individuals produce are quite useful in the Data Analysis domain, like natural language processing and machine learning. Social networking applications like Twitter, YouTube and Reddit contain a large volume of data that are quite useful for various algorithms. Naturally, the need to make information easily accessible to all leads to deploying a conversational agent. In order to build a chat model, a huge volume of conversational text data is required.

Twitter is a microblogging service that allows individuals to post short messages called tweets

Proceedings of the ICON 2020 Workshop on Joint NLP Modelling for Conversational AI, pages 33–39
Patna, India, December 18 - 21, 2020.

that appear on timelines. These tweets were limited to 140 characters, which has been later expanded to 280 characters and prone to change again in the future. Tweets consist of two kinds of metadata that are entities and places. Tweet entities are hashtags, user- mentions, images, and places in the real world's geographical locations. Metadata and short prose add to fewer than 280 characters can link to Webpages, Twitter users. Twitter timelines are categorized into the home timeline and user timeline. Timelines are collections of tweets in chronological order. Twitter API uses Representational State Transfer (REST) API to crawl and collect a random set of sample public tweets. The API allows users to explore and search for trending topics, tweets, hashtags, and geographical locations.

YouTube, a video-sharing website, allows users to view, upload, rate, report on videos. It contains a wide variety of videos such as TV show clips, music videos, documentaries. It also provides a platform for users to communicate and describe their thoughts about what they watch through comments.

Reddit is a social news platform where registered users may submit links, images, text, posts and also upvote or downvote the posts posted by other users. Posts are organized based on boards created by the user called subreddit. It is also a platform for web content rating and discussions. Reddit stores all of its content in json format which can be viewed on the browser by extending the reddit link with the extension '.json'.

Real-time datasets are needed to build a model to generate accurate output. As the available datasets are insufficient and do not contain realistic examples, there is a need to build a crawler which would scrape conversational data. Building crawlers for each website would allow collection of conversational data. This would help in the creation of datasets of conversational data.

2 Literature Survey

A Focused crawler is designed to crawl and retrieve specific topics of relevance. The idea of the focused crawler is to selectively look for pages that are relevant while traversing and crawling the least number of irrelevant pages on the web.

Context Focused Crawlers (CFC) use a limited number of links from a single queried document to obtain all relevant pages to the document, and the said obtained documents are relevant concerning the context. This data obtained is then used to train a classifier that would detect the context of documents and allow classification of them into categories based on the link distance from a query to target. Features of a crawler are-

- Politeness
- Speed
- Duplicate Content

Each website comes with the inclusion of a file known as robot.txt. It is a standardized practice where robots or bots are communicated with the website through this protocol. This standard provides the necessary instructions to the crawler about the status of the website and whether it is allowed to scrape the data off of the website. This is used to inform crawlers whether the website can be crawled either partially or fully, if the website cannot be crawled as per the robot.txt then the server blocks any such requests and can even lead to blocking of IP's.

Websites have robot.txt, which prevent the use of crawlers that attempt to scrape large data from their website. Any request for a large amount of data is blocked almost immediately. To prevent such a case where the crawler should not be blocked, a list of publicly available proxy servers as described in Achsan (2014) is used to scale and crawl the website. Twitter is a popular social media platform used for communication. Crawling such a website can be useful to gain conversational data, and such information can be targeted based on the topic; one such example is a perception on the internet of things as shown in Bian J (2017) or Assessing the Adequacy of Gender Identification Terms on Intake Forms as described in Hicks A (2015).

3 Proposed System

3.1 Twitter Crawler

Twitter API provides the tweets encoded in JSON format. The JSON format contains key-value pairs as the attributes along with their values. Twitter handles both the users and as well the tweets as objects. The user object contains attributes including their name, geolocation, followers. The tweet object contains the author, message, id, timestamp, geolocation etc. The JSON file can also contain additional information in the media or links present

in the tweets, including the full Uniform Resource Locator(URL) or link's title or description.

Each tweet object contains various child objects. It contains a User object describing the author of the object, a place object if the tweet is geo-tagged, and entities object, which is an array of URLs, hashtags, etc. Extended tweets include tweets with longer text fields exceeding 140 characters. It also contains a complete list of entities like hashtags, media, links, etc. They are identified by the Boolean truncated field equals true, signifying the extended tweet section to be parsed instead or the regular section of the tweet object.

The retweet object contains the retweet object itself as well as the original tweet object. This is contained in the retweeted status object. Retweets contain no new data or message, and the geolocation and place is always null. A retweet of another retweet will still point to the original tweet.

Quote tweets contain new messages along with retweeting the original tweet. It can also contain a new set of media, links or hashtags. It contains the tweet being quoted in the quoted_status section. It also contains the User object of the person quoting the tweet.

The Twitter REST API method gives access to core Twitter data. This includes update timelines, status data, and user information. The API methods allow interaction with Twitter Search and trends data.

The Twitter streaming API obtains a set of public tweets based upon search phrases, user IDs as well as location. It is equipped to handle GET and POST requests as well. However, there is a limitation on the number of parameters specified by GET to avoid long URLs. Filters used with this API are-

- follow- user IDs of whom to fetch the statuses
- track- specific words to be searched for.
- location- filter tweets based on geo location.

Workflow of Twitter Crawler:

1. Crawling for public tweets

   This project uses Streaming API access to crawl and collect a random sample set of public tweets. These tweets are crawled in real time. These tweets can be filtered with geo location to crawl tweets with respect to a specific region. These sample sets of public tweets are outputted to a json file as seen in Figure 1.


```
{
  "created_at": "Thu Jul 11 14:55:30 +0000
2019",
 "id": 1139331385288495104,
 "id_str": "1139331385288495104",
    "full_text":    "@basura_inutil_    \"For
example, fat ciswomen are less likely to
receive cervical cancer screening... breast
cancer screening... and colorectal cancer
screening than non-far ciswomen.\"\n\nSource
(with citations): https://t.co/gKTPyRZ8s8",
 "truncated": false,
 "display_text_range": [16,231],
 "entities": {
   "hashtags": [],
   "symbols": [],
   "user_mentions": [
     {
       "screen_name": "basura_inutil_",
       "name": "Yung Sough",
       "id": 252837878,
       "id_str": "252837878",
       "indices": [0,15]
     }
   ]
 }
}
```


Figure 1: The above figure depicts a tweet object. It contains all attributes contained within a tweet

2. Crawling for tweets while searching and monitoring a list of keyword

   For searching and monitoring, a list of keywords, search/tweets is used. Streaming API's status/filters are not used as it does not provide previous tweets at all. Search/tweets are used to crawl and provide tweets from at most a week back. This function continues to endlessly crawl for tweets matching the query. Another advantage of using this method is that it does not limit the number of keywords it can track, unlike statuses/filters, which require separate or new instances to search for different keywords.

3. Filtering Tweets with replies

   The crawler filters through the collected public sample of tweets to find specific tweets with replies. It finds the same by checking the 'in_reply_to_status_id' attribute of the collected tweets. It then proceeds to crawl the parent tweet using the 'in_reply_to_status_id'. It outputs the parent and the reply tweet together in the JSON file, as seen in Fig. 2.

4. Filtering Quoted Retweets

   Quoted Retweets or retweets with comments are filtered from the collected set of

public tweets. This is achieved by parsing the collected tweets and searching for 'quoted_status'. The parent tweet is contained within the quoted retweet inside the 'quoted_status' attribute. The crawler outputs the quoted retweet with its contained parent tweet to the JSON file, as seen in Fig. 3.


```
{
 "metadata": {
   "iso_lanaguage_code": "en",
   "result_type": "recent"
 },
 "source": "<a
href=\"http://twitter.com/download/android\"
rel=\"nofollow\">Twitter for Android</a>",
 "in_reply_to_status_id":
1149330296774393863,
 "in_reply_to_status_id_str":
"1149330296774393863",
 "in_reply_to_user_id": 976575708494413824,
 "in_reply_to_user_id_str":
"976575708494413824",
 "in_reply_to_screen_name": "t3rrordactyl",
 "user": {
   "id": 976575708494413824,
   "id_str": "976575708494413824",
   "name": "Aranv Gupta died protesting US
imperialism",
   "screen_name": "t3rrordactyl",
   "location": "Occupied Ohline Territory",
   "description": "Black american bi polyam
cis woman. You're gonna see either
informative political matters or memes here.
Socialist will win. she/her.",
   "url": null,
   "entities" : {
     "description": {
       "urls": []
     }
   }
 }
}
```


Figure 2: In the above diagram, the 'in_reply_to_status' tag used to filter the tweets is highlighted.

3.2 Youtube Crawler

Cascading Style Selector and Python are the primary tools that are used. The crawler works to accept a CSS selector expression as input; the selector compiles to XPath, and many other libraries such as requests.

AJAX interchanges data with the server in the background enabling the web page to be updated. This allows certain sections to be updated as opposed to the entire page. Python is an object-oriented programming language for general-purpose programming. It helps enhance code readability by including the whitespace.

YouTube has server-side rendering with automation,it is impractical to wait for all comments to get loaded in order to extract them. This work uses the fact that YouTube sends an AJAX Request


```
{
 "geo": null,
 "coordinates": null,
 "place": null,
 "contributors": null,
 "is_quote_status": true,
 "quoted_status_id": 1149078505017286661,
 "quoted_status_id_str":
"1149078505017286661",
 "quoted_status": {
   "created_at": "Wed Jul 10 22:10:39 +0000
2019",
   "id": 1149078505017286661,
   "id_str": "1149078505017286661",
   "full_text": "\"Do not let your little
girl ever take this #vaccine. No matter what
happens, don\u2019t let her take this
vaccine\" - @RobertKennedyJr #hpvvaccine
https://t.co/TeeeqRkPE9",
   "truncated": false,
   "display_text_range": [0,141],
   "entities" {
     "hashtags": [
       {
         "text": "vaccine",
         "indices": [44, 52]
       }
     ]
   }
 }
}
```


Figure 3: In the above diagram, the 'quoted_status' tag used to filter the tweets is highlighted.

to a URL in order to get comments. By using that URL, the crawler makes a session with a user agent and sends the AJAX request to the server for a particular video id input given by the user; all the comments are downloaded and stored in a JSON format; each comment and its reptiles are identified by the id it has, replies have the id of the main comment prefixed with its id.

Begin:
  Initialize a new session with user agent
  Get YouTube page with initial comments
  Extract all the comments by sending an AJAX request
  Obtain remaining comments
  Obtain replies
  Allocate the replies cid's prefixed with main comment id
  Parse the json data in the required format
  Store it in a json file.
End

Figure 4: Algorithm Crawl Youtube Comments

3.3 Reddit Crawler

Reddit is a social media platform that serves the purpose of sharing the content of many forms, including multimedia content. A subreddit is defined to cover a topic of interest, for example, sports and users can make posts on sports, and others can comment on these posts.

Usually, scraping Reddit is difficult since Red-

dit easily blocks IP addresses after a certain number of requests. Hence the work done focuses on viewing Reddit pages as a collection of JSON objects that can be simply parsed for the required content type. This overcomes the issue mentioned above of having a single IP address blocked at multiple requests.

Reddit returns a JSON object when the link is extended with '.json', which is passed as an initial request to crawl, and a callback function is called with the response downloaded from the request. In the callback function, the response object is parsed, and the crawler retrieves the comments. Post links are retrieved based on the search term entered and sorted on hot, new, top, rising, or controversial. The number of post links is limited based on the limit factor. Each of these posts contains a set of permalinks containing information about the comments. Each of these permalinks is parsed, are traversed, and a JSON object is retrieved. Obtained JSON objects are parsed to retrieve the comments.

Scrapy, a web scraping framework, is used to extract structured content from the Reddit page. Libraries such as JSON, time are used. The JSON library provided by python is used to parse JSON objects returned from Reddit links. The crawler output can be seen as in Fig. 5.


```
  {
   "id": "exrxaiu",
   "user-name": "K-369",
   "text": "They should open a category for
performance enhanced athletes only. Only ban
them for serious narcotics but allow hormone
enhancers and steroids.\nThen we'll see the
human body seriously pushed to the limit",
   "time-stamp": "Fri Aug 23 07:43:58 2019",
   "likes": 2,
   "original":
"https://www.reddit.com/r/sports/comments/ct
yn6o/the_worlds_fastest_man_may_be_banned_fr
om_the/exrxaju/.json",
   "topic": "science",
   "depth": 1,
   "replies": [
     {
       "id": "exsgaef",
       "user-name": "VilleKivinen",
       "text": "That category already exists,
it's called the Olympics.",
       "time-stamp": "Fri Aug 23 15:11:43
2019",
       "likes": 1,
       "original":
"https://www.reddit.com/r/sports/comments/ct
yn6o/the_worlds_fastest_man_may_be_banned_fr
om_the/exrxaju/.json",
       "topic": "science",
       "depth": 0,
       "replies": []
     }
   ]
  }
```


Figure 5: Crawler output in JSON format

4 Context Based Dialogue Filtering

The tool used in the comment filtering module is python libraries centered around handling and manipulating the data obtained from the crawlers. The filtering model uses the pandas' library to read the raw data from a .csv form of the output json data from the crawlers. The filtering model uses the nltk library to clean our data, tokenize the data, and remove stop words.

The outputs collected from the crawlers are all initially in JSON files. These are then converted into a uniformly structured csv type file. The main technology used is in the form of the bag of words model used to analyze the importance of each generated token within the context of the extracted data.

In the implementation, note that input is the csv file, and output is a cleaned and appended list of comments and replies. The comments and replies are first to read into a data frame following which the following cleaning methods are applied: Convert to lowercase "[/()\[\]\—@,;" symbols are replaced by a space "0-9a-z +_" symbols are removed

Stopwords are removed according to the 'English' stopwords from the nltk stopwords library

The first 30 comments are then analyzed to generate a list of tokens, and their frequencies are counted. Tokens with "" are given high preference, and a high-frequency list of words is taken as a subset of the original list. Then all the comments and replies from the data frame are cross-referenced. Entries that do not contain any of the words in the list of high-frequency words are rejected. The remaining entries which have been filtered are the output.

Begin
Convert .json files to a standard .csv file.
Extract the comments and replies columns into a dataframe.
Clean the raw data.
Tokenize the entries in the dataframe and calculate the word frequency for each unique word.
Generate a high frequency word list as a subset of the former.
Cross-refer the remaining entries and filter in those that include a token from the high frequency word list.
This output is cleaned, filtered data.
End

Figure 6: Algorithm Comment filtering

5 Analysis and Comparison

5.1 Twitter

Standard twitter crawlers use the Streaming API statues/filters, which do not provide old tweets. The other caveat is that it can only track a limited number of keywords. So, if the user has a lot to track, he will need to have many separate instances, each tracking different parts of the keywords.

The proposed Twitter crawler is beneficial in that it uses search/tweets to get old tweets. It searches for a portion of the keyword list at a time. The Streaming API is used to collect a random sample of real-time public teams. The REST API is used to filter and obtain tweets with respect to specified filters. The implemented algorithm filters the tweets to obtain tweets relevant to conversational data.

5.2 YouTube

Selenium, a vast tool for scraping, can crawl YouTube. YouTube has server-side rendering, which loads the website first upon which it can be scraped for data. This approach's problem is that the website has to be loaded upon which JavaScript has to be used to load more comments on the web page.

The novelty in this approach lies in that by analyzing how YouTube retrieved its comments, and the same approach can be used by sending AJAX requests to the URL https://youtube.com/id/comments. Since sections of the page that does not include comments are not downloaded, the crawler saves time and resources over the downloaded contents.

5.3 Reddit

Reddit being a dynamic website and communicates through rest API; selenium is a powerful web scraping tool that can be used to scrape dynamic websites. While simulating the browser, crawlers can mimic the scrolling of the pages, clicking to read more comments. Since the speed of retrieval will depend on the speed at which each section of the page loads, it is inefficient as the next load will only start after the previous load and the cursor scrolls further down.

Scrapy is faster and robust in handling errors but does not allow the crawling of dynamic sites. Scrapy can be provided with a Reddit URL extended by the '.json' link and get all internal links consisting of comments and replies. Instead of traversing the Reddit page using html and selectors, which would be time-consuming, a simple scraper has been built using the functionality of scrapy.

5.4 Filtering of extracted data

The filtering model is a simple algorithm that avoids excessive computation as it does not look outside the extracted data for filtering conditions. It works more along the lines of "sticking to the topic" by using the high-frequency terms and hashtags from the extracted data itself. A sample of filtered and unfiltered data for a Twitter search term "cancer" is attached below. One of the two variables to note here is the number of comments used to generate the high-frequency list - . Ideally, it should depend on the volume of extracted data, at least 2-5% of the total volume.

The other variable is the frequency number-used to classify a token as highly frequent. This would depend on the number of tokens used to create a high-frequency list. Approximately the first 80% of the tokens, when arranged in ascending order in terms of their occurrence frequency, need to be rejected. Samples of such filtering are seen in Figure 7 and Figure 8.

```
The extracted comments are:
  "looking another reason put cigarettes
according american cancer society smoking
increases risk cervical cancer also reduce
risk getting hpv vaccine scheduling regular
pap smear #papsmear #cervicalcancer
#savannahga"
  "circumcision greatest scam fell could
choose age knowledge would say"
  "give freaking break"
  "misleading claim take look list services
offered even ones sound like something
family doctor might arent ie pediatric care
doesnt include vaccinations prevention
doesnt include routine cancer screening like
pap tests 2"
```

Figure 7: Accepted and cleaned comments. This is a sample of accepted comments that were cleaned prior filtering for the searched keyword, "cancer".

Thus we see that this filtering algorithm achieves highly flexible strictness with respect to which comments are accepted and with no bias. Since this is a simple implementation of a modified bag of words model, it is computationally light in comparison with ML models that do the same.

6 Conclusion

The proposed crawler can fetch multi-turn dialogues from Twitter, YouTube. The crawler can scrape conversational data from Twitter, YouTube.

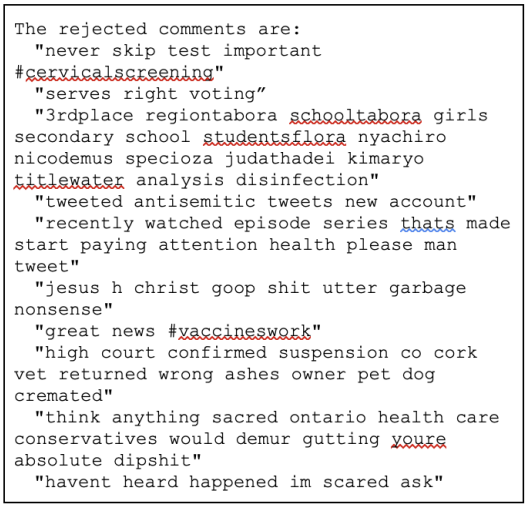

Figure 8: Cleaned but rejected comments. This is a sample of rejected comments that were cleaned before filtering for the searched keyword, "cancer".

Unstructured data retrieved from the crawler is converted to well-formatted data. Streaming API has been used to crawl Twitter and retrieve random sets of public tweets. Quoted Retweets or retweets with comments are filtered from the collected set of public tweets. The quoted retweet, with its contained parent tweet, is outputted to the JSON file. YouTube crawling is made easy without any limitations as like which the YouTube data v3 API has, and getting comments has never been this easy and fast before. Reddit crawler parses the JSON object from the response downloaded from the spider's request and can get the comments of the posts.

References

Wahyu Achsan, Harry Wibowo. 2014. A fast distributed focused- web crawling. *Procedia Engineering*, pages 492–499.

Sandip Chauhan Ayar Pranav. 2015. Efficient focused web crawling approach for search engine. *International Journal of Computer Science and Mobile Computing*, 4:545 – 551.

Hicks A Yuan J He Z Xie M Guo Y Prosperi M Salluom R Modave F Bian J, Yoshigoe K. 2016. Mining twitter to assess the public perception of the "internet of things". *PLoS One*, 11(7).

Salloum RG Guo Y Wang M Prosperi M Zhang H Du X Ramirez-Diaz LJ He Z Sun Y Bian J, Zhao Y. 2017. Using social media data to understand the impact of promotional information on laypeople's discussions: A case study of lynch syndrome. *J Med Internet Res*.

Satish Kumar Dhiraj Khurana. 2012. Web crawler: A review. *International Journal of Computer Science Management Studies*, 12(1).

Rutherford M Malin B Xie M Fellbaum C Yin Z Fabbri D Hanna J Bian J Hicks A, Hogan WR. 2015. Mining twitter as a first step toward assessing the adequacy of gender identification terms on intake forms. *AMIA AnnuSymp Proc*.

Vassiliki Angelis LefterisVakali Athena K. Paparrizos, IoannisKoutsonikola. 2010. Automatic extraction of structure, content and usage data statistics of websites. pages 301–302.

M. Singhal M. Bahrami and Z. Zhuang. 2015. A cloud-based web crawler architecture. *International Conference on Intelligence in Next Generation Networks)*, pages 216–223.

Varnica. Stockmeyer Mini Singh Ahuja, Dr. Jatinder Singh Bal. 2014. Web crawler: Extracting the web data. *International Journal of Computer Trends and Technology (IJCTT)*, 13(3):132–137.

Károly Nemeslaki, AndrásPocsarovszky. 2011. Web crawler research methodology.

Christopher Olston and Marc NajorkInfolab. 2010. Web crawling. *Stanford university, Foundations and TrendsR in Information Retrieval*, 4(3):175–246.

Felix K Akorli Pavalam S M, S V Kashmir Raja and Jawahar M. 2011. A survey of web crawler algorithms. *IJCSI International Journal of Computer Science Issues*, 8(1).

Jawahar M. Pavalam S. M., S. V. Kasmir Raja and Felix K. Akorli. 2012. Web crawler: Extracting the web data. *IJMLC*, 2(4):531–534.

Hui Shen Xiaoyu Li Wenjie Cao ZiqiangNiu Shuzi. Yanran, Li Su. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset.