Extract and Summarize Tweets

Tanvi Kamble (2018114004)
Chaitanya Agarwal (2018114003)
Apurvi Mansinghka (2019201093)
Jyoti Gambhir (2019201032)

Scope Document

1 Introduction

Twitter has become a mainstream platform for users to express their opinions on any topic and engage in debates. A large amount of valuable real-time information is posted on twitter. However this information is dispersed among millions of tweets containing varied sentiments, and opinions of the masses. User debates and interactions yield content based a specific topic which is referred as a 'trend'. Twitter trend extraction aims at clustering/grouping tweets with similar themes that are generated in a short period. To effectively utilize this content, it is essential to not only extract the information, but also to summarize the extracted data to move forward for the required analysis.
The most straightforward approach to solve this problem is to utilise 'hashtags'. It is a type of metadata tag used on social networks such as Twitter and Instagram and other micro-blogging services. It lets users apply dynamic, user-generated tagging that helps other users easily find messages with a specific theme or content. We attempt to get insights on any latest trend or social movements by extracting the tweets with relevant hashtags. The extracted tweets will then be put though a custom pipeline to summarize content and analyse varying sentiments over time.

1.1 Problem Statement

Given a set of hashtags, scrape twitter for tweets related to them and analyse (summarisation, sentiment analysis etc.) the scraped content.

2 Pipeline Design

2.1 Twitter Dev API Access

The first step is to register for a Twitter Dev Platform account to get access to the Developer APIs which allow scraping tweets.

2.2 Scraping Tweets

Multiple Python Libraries are available which provide utility to scrape or collect tweets. These libraries require an API Key from the previous step. Some of the libraries relevant to this project are as follows:

2.2.1 Tweepy

Tweepy is a Python library for accessing the Twitter API. There are several different types and levels of API access that Tweepy offers.There are limitations in using Tweepy for scraping tweets. The standard

API only allows you to retrieve tweets up to 7 days ago and is limited to scraping 18,000 tweets per a 15 minute window. Using Tweepy is great for someone who is trying to make use of Twitter's other functionality, making complex queries, or wants the most extensive information provided for each tweet.

2.2.2 Twitterscraper

A script that is used to scrape for Tweets using the Python package requests to retrieve the content and Beautifullsoup4 to parse the retrieved content. With Twitter's Search API you can only send 180 Requests every 15 minutes. With a maximum number of 100 tweets per Request this means you can mine for 4 x 180 x 100 = 72.000 tweets per hour. By using TwitterScraper you are not limited by this number but by your internet speed/bandwidth and the number of instances of TwitterScraper you are willing to start.

2.2.3 GetOldTweets3

GetOldTweets3 was created by Dmitry Mottl and is an improvement fork of Jefferson Henrqiue's GetOldTweets-python. This package allows to retrieve a larger amount of tweets and tweets older than a week. However, it does not provide the extent of information that other libraries.

2.3 Pre-Processing

After the tweets are obtained, they're put through the pre-processing pipeline which involves replacement of abbreviations with their expansions, link removal, contraction replacement, target hashtag removal, white-space and punctuation handling etc. This helps us filter out elements which do not contribute to sentiment analysis and summarization.

2.4 Event Extraction

We attempt to extract event information like temporal expressions, named entities, and event-referring phrases from the pre-processed tweets. This is done using algorithms like Conditional Random Fields (CRF) that have been proven to be efficient for named entity recognition and event tagging. There is no deep analysis like co-reference resolution done at this stage.

2.5 Summarization of Tweets

We attempt to create event graphs where in the nodes are NE and event phrases and the weighted edges are the number of tweets in which the two nodes co-occurred in. We use this graph to summarize our tweets using PageRank-like algorithms.

2.6 Age based Sentiment Analysis

Lastly we attempt to do Sentiment Analysis on the extracted tweets with respect to the age of the hashtag, and the event. This is done in order to analyse how the opinions and sentiments of people about a certain event change with time.

3 Milestones

1. Deliverable 1 : Project scope documentation - 27 Sept
2. Scraping and pre-processing tweets - 3 Oct.

3. Deliverable 2 : Report on initial analysis iterations - 15 Oct.

4. Deliverable 3 : End to end system - 25 Oct

5. Deliverable 4 : Optimized final system - 10 November

4 Challenges

We expect pre-processing to be challenging due to a high amount of noise in the tweets. Moreover, event extraction, a notoriously tough NLU task, might yield upsetting results.

References

[1] Goran Glavaš, Jan Šnajder, *Event graphs for information retrieval and multi-document summarization*, Expert Systems with Applications, Volume 41, Issue 15

[2] Wei Xu, Ralph Grishman, Adam Meyers, Alan Ritter, *A Preliminary Study of Tweet Summarization using Information Extraction*

[3] Zahra Majdabadi, Behnam Sabeti, Preni Golazizian, Seyed Arad Ashrafi Asli Omid Momenzadeh and Reza Fahmi, *Twitter Trend Extraction: A Graph-based Approach for Tweet and Hashtag Ranking, Utilizing No-Hashtag Tweets*, Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6213–6219 Marseille, 11–16 May 2020

[4] Roshni Chakraborty, Maitry Bhavsar, Sourav Dandapat, and Joydeep Chandra *A Network Based Stratification Approach for Summarizing relevant Comment Tweets of news articles*, Indian Institute of Technology, Patna