Finding event location based on the user generated tweets

Amin Majdi

April 30, 2022

Abstract

In the last few years, finding the location of the users who are talking about a specific event become crucial for many companies, governments, and applications. Finding the location of the users in the tweeter is not an easy task, because almost all of the tweets do not have the 'Place' tag and the other information about the location of the users is very noisy [6]. In this Project, We developed a method to label the tweets about specific hashtags and successfully classified them based on the users' country who published these tweets. We predicted the countries in which a specific event is most popular. Our results revealed that the trained model can approximately predict the location of four users out of each five users correctly based on their tweets metadata.
Keywords: Social media; Twitter ; classification ; labeling ; real-time event detection ; location finding

1 Introduction

It has been a decade since social media has been one inseparable part of our day-to-day life. Detecting the event location and the groups connected to the events is very important to many companies whose services are matched. All the big companies focus on the data coming from social media to find the target population better and expand their businesses. one of the most valuable information about the customers is their location. In this project, we focus on predicting the location of the Twitter users based on the tweets' metadata. To reach this goal, we followed the incidents happening in the world and found the related hashtags. Our primary focus was on the Ukraine War, Easter, and Ramadan events. The metadata that we focus on consists of the place of the published tweet, the user's account location, the user's account name, the user's account screen name, the user's account description, the tweet's text, the language of the tweet, and the domain of the user' account website address. Solving this problem is very challenging due to the noisy nature of the tweets. Most of the tweets don't have the appropriate tags for the location of the tweets, and we need to address this by extracting other features in the data. The database used in this project is unlabeled, So at first, we aimed to label these data, then we used the dataset to train a model to predict the users' location based on the tweets.

2 Method and Data and Evaluation

1. Data collection
   To collect data, we used the tweepy library.This module. helped us to collect some real time tweets from twitter using some hashtags. In this task, we filtered all the retweets and

the tweets can be in any language. We focused on three popular topic that is related to an incident or event happening in the world right now which are the Russia-Ukraine-war, the Ramadan and the Easter.

(1) Russia-Ukraine War
To collect the tweets about this topic, we use ukrainerussianwar, ukrainewar, warukraine ukrainianarmy and militaryukraine hashtags. The total number of collected tweets is 10000. We collected these data-set at April 23Th.

(2) Easter
For this topic we only used Easter hashtag. When we collect this data-set, It was some time after the even, So we could only collect 2000 tweets.

(3) Ramadan
We wanted to see the influence of other languages on our model as well, So, for this topic we used the Ramadan and the translation of the Ramadan in Arabic as our hash-tags. As a result, we collected many Arabic tweets as well. Totally, the sizee of initial database for this topic is 10000 tweets.

2. Auxiliary database.
To label the dataset, we needed some ground Truth to compare our data. So, we added 5 other databases to this project.

(1) Cities and Countries [5].
this file contains 42906 well-known cities in the world. It also has the country names, two character country names ”iso2”, and some other information for each city.

(2) Country Codes [3].
This file contains the name of 252 countries around the world alongside with their two character names.

(3) Language code [4].
This database have the 34 different languages that is used in the twitter. For each of the languages, there is a two character abbreviation that is the key code for that language in twitter.

(4) Flags [1].
This file contains all the emojis for country flags that people use in the tweets.

(5) Country specific domain [2].
There is an specific domain for each country that can be known by the suffixes at the end of the domain address. These suffixes contain only two characters that is similar to the ”iso2” code for each country.

3. Data preparation
In this section we follow two main goal. The first goal is to label the data and the second one is create a file format that is compatible with the VopalWabbit library to classify the data.

(1) Metadata Introduction and name-spacing
Each tweet, contains many different and divers information that is called metadata. In this task, we focused on eight different metadata in each tweet to extract information about the location of the user that tweeted that tweet. we assigned a name-space to each of these metadata to make different features in the final classification task.

(1) Place
The ”place” field in the tweet shows the exact location that the tweet has been published from. We extracted the ”iso2” ’country_code’ that is inside the ”place” meta data.

(2) Location
In this section, users can specify any location for their account. There is so validity check for the correctness of the location the people specify. In other words, people can write any thing they want in this section, from their city or country name to some irrelevant notes like ”in your heart” or ”the world” as their ”location”.

(3) User’s Name
This tag specified name of the users in their twitter account.

(4) User’s screen Name
The screen Name tag is the name the twitter shows for that account on the main account and for each tweet that that account publishes.

(5) User’s screen Description
The twitter let users to write a description about themselves. Sometimes the description may contain some information about the location of the user.

(6) Tweet’s full text
This tag contain the main text of the tweet.

(7) tweet’s Language
The language tag shows the two character language code for the language in which the tweet is written.

(8) Website Domain
Sometimes, users specify a website i their account. ”user.url” gives this information. If the website address contains the country domain, it can be usefully for predicting the location of the user.

(2) Label Check Methods
In the main code for this project, we wrote some functions to find the appropriate labels for each tweet. they receives the tokens as their input and compare them with the databases and return the label if it is available based on the input tokens.

(1) Check City or Country
The function searches for cities and countries in the input tokens.

(2) Check Flag
This function finds the flag emojis in the tokens and returns the associated country.

(3) Check Domain suffix
This function uses the regular expression method to find the country domains i the website addresses if there is any. The output is the country label.

(3) Labeling
In this section of the code, we extracted the metadata from each tweet and used the label check methods which is defined in the previous section. There is a priority in the usage of the data for labeling and we only use the next data if we could not find the label based on the current metadata. the queue is as place, location, user’s name, user’s screen name, user’s screen description and the tweet’s text. The ”place” tag contains the exact label that we are interested in and there is no need for further process. For other tags, First we did the tokenization, then we passed the tokens to the labeling functions to find the related labels. Finally, we converted the labels to numbers. at first we had

252 labels with was assigned t all the countries in the world, but we decided to find top countries that had the most users talking about the searched topic and include the rest countries in "others". As a result, for the Ramadan and the war topics, we had 20 classes and for the Easter event, we had 10 classes.
Ultimately, we found the label for 1438 tweets in Ramadan topic, 2732 tweets in war topic and 746 tweets in the Easter topic. The Ramadan data-set is significantly smaller than the War data-set due to missing the features that are in the Arabic language.

(4) VW compatible file
the VopalWabbit's classifier's train and test files should be in a defined format. So we created a function to write the classes and all the name-spaces and the features in the format which is compatible with the VopalWabbit library. we shuffled and divided the data in half for train and test data.

4. Classification
After labeling the database, it was time to classify the data using VW multi-class classifier. For the Ramadan topic, War topic and the Easter topic, we used the following command to train the models respectively:
!vw -k -c -b 27 –oaa 20 -d file.tr -f file.model –passes 20 –holdout_after 650 -q pp -q pl -q pg -q lg
!vw -k -c -b 27 –oaa 20 -d file.tr -f file.model –passes 20 –holdout_after 1200 -q pp -q pl -q pg -q lg
!vw -k -c -b 27 –oaa 10 -d file.tr -f file.model –passes 20 –holdout_after 302 -q pp -q pl -q pg -q lg
We only added quadratic features which did increase our accuracy. We also used other methods like NN, ngrams, affixes and other loss functions but non of them were successful in reducing the loss.

3 Results and Insights

- Labeling Result
As the data was very noisy, we could label about %20 of the data in our data-set. Table 1 shows the number of the tweets that were labeled successfully in each topic.

	Ramadan	**Russia-Ukraine-War**	**Easter**
Original tweets	10000	10000	2000
Labeled tweets	1438	2732	746
Number of Classes	20	20	10

Table 1: The number of successfully labeled tweets

We also found the top countries which their people talk more about our interested topics. Here is the list of these countries:
When we analyzed the data we found out that the existence of some of the countries in this lists are not expected. As an example, we didn't expected Tonga be in the top 20 countries that are talking about the War. Further analyzes revealed that the tweets comes from Tonga are all originated from one account. The account seems to belong to a scammer who constantly tweets about the War.

Event	Top countries
Ramadan	saudi arabia, united states, pakistan, kuwait, egypt, emirates, indonesia, algeria, india, united kingdom, oman, nigeria, france, netherlands, yemen, turkey, norway, bahrain, sudan
Russia-Ukraine-War	united states, ukraine, russia, united kingdom, india, norway, japan, germany, czech republic, italy, netherlands, france, spain, philippines, finland, canada, poland, turkey, tonga
Easter	united states, ukraine, united kingdom, united states, ukraine, united kingdom, netherlands, timor-leste, ethiopia

Table 2: Top countries which their users were talking more about the specific topic

- Classification Result
  The vopalWabbit Classifier reports the best loss for the train and test process. Here is the results for each of the topics:

Event	Train loss	Test loss
Ramadan	0.25	0.24
Russia-Ukraine-War	0.12	0.17
Easter	0.26	0.18

Table 3: The reported loss for classifications

The loss for classification for the Ramadan event is higher than two other event. This happens because many of the tweets related to the Ramadan event are in the Arabic language and we only focused on extracting features from English tweets.
Also, The loss for the Easter event is higher than the loss for the War duo to the small number of training data for the 10 class classification.
Altogether, The results shows that our model is approximately predicting location of 4 out of 5 tweets which is grate.

4 Conclusion and Future work

We successfully extracted the features of the tweets and revealed that we can predict the location of the tweets based on it's metadata. We also reveled that we can find the location of the people who are talking about an specific topic. We reached to two main goal which was labeling and classifying the tweets based on the location of the users.
We can improve this work in many ways. A better tokenization which focus on the noun tokens can better find the cities and countries and reduce the run time. Also, tokenization can be generalized to other languages to better extract the location features.
furthermore, we can assign multiple labels to a tweet with different weights. In this project we couldn't use the language tag for labeling the data due to the ambiguity that it can create. there are a lot of countries that uses the same language and it is hard to label the data based on language itself. By multi-labeling using weights, we can use the language tag in labeling as well.

References

[1] Flags. https://emojipedia.org/flags/. Accessed: 2022-4-23.

[2] Internet country domains list. https://www.worldstandards.eu/other/tlds/. Accessed: 2022-4-23.

[3] List of all countries with their 2 digit codes (ISO 3166-1), howpublished = https://datahub.io/core/country-list#resource-data, note = Accessed: 2022-4-23.

[4] Supported languages and browsers. https://developer.twitter.com/en/docs/twitter-for-websites/supported-languages. Accessed: 2022-4-23.

[5] World cities database. https://simplemaps.com/data/world-cities. Accessed: 2022-4-23.

[6] Hamdy Mubarak and Sabit Hassan. Ul2c: Mapping user locations to countries on arabic twitter. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 145–153, 2021.