Twitter Trend Topic Categorization

Nihal Ezgi Yuceturk
Computer Science
Distributed Information System Laboratory, EPFL

Abstract

Social media is deeply, inevitably and unimaginably effecting our world both in social and technical aspects. Social media platforms like Twitter is reaching millions of people daily; gathering their ideas, altering their views and connecting them. In this work, we strove to figure out what people talk about in Twitter in every day. We collected and analysed a big dataset of tweets along with daily trending topics, clustered them and categorised the trending topics into conventional media categories by using LDA. As result, we created and published a large, clean dataset of tweets matched with their trend topics, general categorizes and keywords to be used in summarization or description purposes in the future,

Introduction

Social media nowadays a borderless information channel of the world. Since it came into life it connected people at an unprecedented level and helped to spread the information and knowledge of mankind in more than every other event happened throughout history. Today, people are almost following every news regardless of domestic or international, every fashion trends, every celebrity and even scientific discoveries or stock markets via one multiple social media channels.

Among these channels, Twitter is one of the most widely used and effective ones. Reasons behind Twitter's popularity is still an active but highly speculated research topic which requires both technical and sociological aspects. Nonetheless, it is known that Twitter has an enormous, de facto power of directing people's idea and creating a popular opinion. One way of achieving this to gather people with same or similar ideas together, A method of many which Twitter uses is to provide trending topic information. How Twitter detects and decides to trend topic information as one or multiple precise keywords though is a black box operation for the public. Which of tweets is the interest of a particular trending topic is also unknown.

Our research was initially started as a quest to shed a light on these black box operations using a very big, handcrafted Twitter dataset. The original idea was to construct a clustering method for tweets in which each cluster will represent a wider, a more general trending topic, namely a Category. However, recently Twitter has started providing category information for its trending topics. Although those categories automatically provided are not perfectly sensible for the human it offers broad information of what people think as important mostly per that day. We did not, however, stop the project since the end result is already provided but continued to investigate since Category provided for trending topic are also a black box. Instead, we used those Categories as ground truth for our model.

In this project, we built a Trend Topic Categorization system using Latent Dirichlet Allocation (LDA) method, which is a popular NLP based clustering algorithm. We collected data, created a processing model, created a categorization model and tested the quality of the overall system by a human judge. By categorizing trend topics and tweets we achieved:

1. collected and produced qualitative and quantitative (stats) on tweets
2. obtained keywords later to be used in trend topic creation
3. obtained obtain keywords for summarization

Even though there have been many projects done related to the analysis of Twitter data mostly on NLP based methods, our main contribution can be summarized as follows:

1. trend topic match rules with the tweets
2. publish full dataset with matches and match rules
3. keyword list per category
4. automatized process for very very big data

In the following sections, we summarized the related previous works done, our methodology in detail, our results and discussion and finally the future work.

Related Work

Knowing the public opinion, what people think and how would they react has such importance that it can even change the results of elections, the destiny of the countries. Twitter, as a platform, is one of the gold mines for such knowledge. Therefore, many research was conducted and many works are published.

Topic classification and detection initially were required for comparison reasons. One of the most popular research is on Twitter is its comparison of traditional media [15], [6]. These were interested in finding a connection point with the conventional media, namely the news categories. They used statistical models to assign a tweet to a topic of news media, for example, KL divergence of a set of words in the tweet text with the New York Times article.

With best of our knowledge, the oldest attempt of topic modelling in Twitter is by Hong et al, [5] where they also used a kind of LDA and Author topic modelling, not for direct classification task but to obtain the signature of the tweet They later used the signature for downstream classification task of 15 defined category and hand-labelled the data using linear classifiers. Creating a signature for tweets was also used in Portuguese tweets' topic detection [10], namely Fuzzy fingerprints where those fingers prints were classified by the KNN model with again on hand-labelled data. These kinds of approaches aim to represent text data in a compact form which is a bag of words or numbers format where it is easier to used in downstream tasks. Another example is TF-DF and network-based classification of tweets on predefined topics with hand-labelled data [7]. An ultimate version of representation is the usage of words embeddings [8], where it was used as an auxiliary task for summarization and description along with LDA and Gaussian LDA [13].

Some approaches gave a chance network-based and hierarchical models where the network designed by following-follower relationships and weights assigned by the popularity of the users [14], [11], [4]. Those models again counted on human-labelled data. Assignments were made based on an anchor tweet of a celebrity, or an expert person. A topic was assigned to the anchor manually then retweets or mentions were classified with the same topic.[14].

In many works mentioned here, hand-labelling was preferred by authors due to small data size and ambiguous, bad linguistic qualities of micro-blog texts, resisting automatic labelling. However, this method is neither efficient nor sufficiently large enough to assess the quality of the work. Semi-supervised [3] and graph-based summarization [12] have been becoming popular recently but the main problem of how to match a tweet with a trending topic is remaining as a challenge. A widely used approach, which we as well inspired form is decomposing the hashtag trend [1] and match it with a word(s) in the tweet's texts. Although, it is slow and has potential problems due to bad linguistic qualities of micro-blog texts it significantly reduces the human interference to the automated processing.

Methodology

We used the topic modelling pipeline for this project. The stages were as follows:

- Collect raw data, tweets and trend topics
- Couple trending topics with the tweets
- Construct trend document structure
- Construct and train LDA models
- Assign categories to the topics found by the best model

Data Scrapping Using Twitter API

Data collection started years ago using Twitter's Stream API with English language option. We are possessed of millions of raw tweets along with their user id, tweet id, creation time etc. from 27-09-2011 to 30-09-2019. Similarly, we had scraped the trending topic data of the day provided by Twitter API starting from 07-07-2013 to 08-11-2019. We have collected an overall 4.2 million of trend topic along with their duration information during this time period.

To ease of use and for proof of the concept we only consider one month of data for this project. We preferred choosing longer text (280 character text) of Twitter which is released after 01-10-2017 then chose July 2019 as a full month as a smaller representative sample. We obtained 99.3 million tweets and 14,200 trends for that month. We later matched each trend with one or more tweets, or vice versa by using a hand-crafted matching algorithm.

Trend Augmentation and Matching Rules

A single glance at the trend data provided us with the insight that Twitter is not using an efficient separation and grouping while reporting the trending topic data. For example, for a single day both **#2cupsstuffed** and **#2CupsStuffed** exist in our dataset. Semantically they refer exactly the same trend topic but due to syntax difference, Twitter reported them as two different topic. To avoid this problem we design a trending topic augmented set for each trending topic. This set is to hold upper and lower case version, hashtagged and no-hashtagged version as well as camel case (*CamelCaseExample*) split, upper, lower and rejoin versions of the same trend topic. This helped us to obtain multiple keywords to search in the tweet text to later match them with the trending topic. This also helped us to eliminate duplicate trends since we are able to catch them together if they differ only syntactically.

We used detailed augmentation method with trends with only one word (In fact, many trend topic is obtained as a single word this is due to hashtags or conventions). We used only upper and lower case versions in the augmented set for trend topic contains multiple words.

While creating the augmented set, we add each augmented version of the trend word with a rule to obtain them. For example, augmented set of hypothetical trend topic **#EzgiYuceturk** would have following augmented set = {

RuleMatch(word=#ezgi, rule=camel lower hashtag),
RuleMatch(word=#EZGI, rule=camel upper hashtag),
RuleMatch(word=ezgiyuceturk, rule=no hashtag lower),
RuleMatch(word=#EZGIYUCETURK, rule=simple
upper), . . . }

With this, we obtained at least 3 at most 15 versions of the same trending topic to match in the raw tweet text.

Trend-Tweet Matching

We did not clean and preprocessed the data before matching. The Twitter user generally tweet informally with using lots of idioms, acronyms and typos. We believed (also by looking at the trend topic data) that Twitter does not correct or alter acronyms while reporting the trending topic. Therefore, we thought it would be straightforward to search in the raw text.

Another issue was related to the time course of the trending topic. For a topic to appear as a trending topic, we believe, it requires a certain number of related tweets. Many of those tweets are supposedly written before the time of the topic and some of them after. We use a candidate set of tweets to match with the trend topics of the day. We used the tweets of one day before, one day after and the same day of the trending topic appeared in the candidate set of tweets. For example, trending topics of 10-07-2019 were searched within the tweets of 09-07.2019, 10-07.2019 and 11-07-2019.

We run our matching algorithm to match each tweet with one or more words in the augmented set of each trending topic. The search was based on an exact match of two words. We replicated a tweet for each trending topic it was matched by the algorithm. We then add the Rule Match values, the version matched and rule of the version. The algorithm run and reported by day based and matched 4.7 million tweets with one or more trending topic. With ranging from 1.9% to 6.2% per day, it covered 4.77 % of the tweets of the initial set.

Preprocessing

Before supplying data for any supervised or unsupervised model, a decent preprocessing and cleaning requires. For this, we removed digits and none-ASCII characters from trends and altered them to the lowercase; and removed digits, URLs and none-ASCII characters and lowered the case of tweet text. As a standard procedure, we also used the Clean method of preprocessor API !!!cite, a method which is specifically built to be used on Twitter data before any NLP pipeline applied. We removed the empty tweets and trends and obtained 4 million tweet text over 6700 trending topics to be used in training.

Trend Documents

As known, LDA takes documents to cluster. We define trend documents as the joined text of the tweets that each trend has been matched before. We collected the statistics like how many tweets a trend matched vice versa, how many authors a trending topic has and etc which is reported in the result section. We also randomly selected 1/100000 of the tweets (400 trend-tweets for human judges) as a manual test set later to be used while assessing the quality of the models. Train and test sets then were stored for reproducibility of the results.

LDA Model Creation and Selection

We used the Gensim [9] corpora and Gensim model to build and run our models. We removed the stop words using NLTK [2] default stop words for English and extended it to include the Twitter jargon, eg. rt, tbh, etc. We tokenized the text and uses Porter Stemmer of [2] for stemming. We did not use lemmatization since the Twitter text is not necessarily following the correct language structure. In this case, lemmatization would be ineffective. We also used a basic stemmer due to very long processing time of advanced stemmers in our big dataset. We created the corpus and dictionary from the stemmed dataset of trend documents. Our corpus and dictionary were the sizes of 6700 and 230000 respectively. All of these data structures were also stored for reproducibility of results.

We run our models in iccil1-ds-20.epfl.ch cluster without using parallelism. Chunk size was chosen as 100 while passes over corpus for each update was chosen 5. We chose alpha of the LDA as auto meaning that we initially assigned equal probability distribution to the categories over the documents.

We used perplexity and coherence scores while evaluating the goodness of the models as standards. The scores of each model with respect to a different number of topics/category are reported in the results section. We chose the best model as the last model before the coherence score starts dropping while topic numbers increase. In this case, the best model is the model with 19 different categories, which also gave the second-best perplexity score of all models.

Topic Categories and Testing

LDA clusters/categorizes the documents according to latent distribution in the dataset. This distribution is calculated by using the distribution of the words in the dictionary over each document. It yields an assignment for each document to multiple categories with a score as well as the weight/importance of the words in that category.

Our model gave us the 19 category distribution for trending topics. We then by checking at the most important (biggest weight) 20 words in each category to assign a name by hand like *Entertainment, Sports, ... etc*. Assignments can be found in the results section. We later used our test dataset to assess the quality of our assignments. Same preprocessing and cleaning were applied to test set. It's stemmed as well but while building the test corpus to test the LDA model we used the training dictionary. Manual test results can be found in the discussion section.

Results

Statistics

Our algorithm matched approximately 4.7% of overall tweets. That's partly because we were looking for the exact keyword match in the augmented set of the trend.

	Raw Tweet Number	Matched Tweet Number	Percentage %
SUM	99,368,601	4,742,358	4.773
AVG	3,205,439	152,979	4.773

Table 1: Average Tweet Match

For our algorithms, a tweet mostly is assigned with one trend although some of them had keywords that had matched up to 54 trending topics. These tweets, we believe, are spam and included generic keywords that matches with multiple trending topic.

#of Trend a	mean	std	min	25%	50%	75%	max
Tweet Matches	1.696	1.172	1	1	1	2	54

Table 2: #Trend A Tweet Assigned With

Also for trend perspective, there are some trending topics attracted a huge community. In this case, **#MTVHottest** matched with the maximum number of tweets of all.

#of Tweets a	mean	std	min	25%	50%	75%	max
Trend Matches	561.39	2489.98	1	8	65	314.25	111234

Table 3: #Tweet A Trend Assigned With

Model

Results of multiple model run. We chose Model19 (model with 19 category) due to highest coherence score. Although perplexity score is not the highest, according to the figure category distributions it is good enough to serve our purpose.

Topic Number	Perplexity Score	Coherence Score
7	-8.086	0.500
10	-8.374	0.502
13	-9.017	0.574
16	-9.861	0.591
19	-10.559	0.598
22	-10.935	0.579
25	-11.298	0.579

Table 4: Scores of Models

Distribution of the categories and distribution of the most used words can be seen in Figure 4.

Trend by Time

We also checked how the category of trending topics is changing by time. These graphs also show how balanced the LDA model was. For this, we first show a number of trending topic each category posses by the time interval in Figure 5. We also graph how many people, namely the author tweeted in that category Figure 6. Both of these graphs were to show the popularity of the categories we found. We may say these are basically why and how people use the social media platform, Twitter.

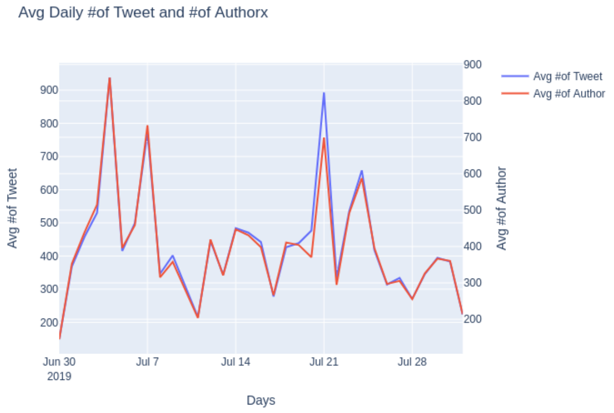

Figure 1: Daily # Tweets vs # Authors

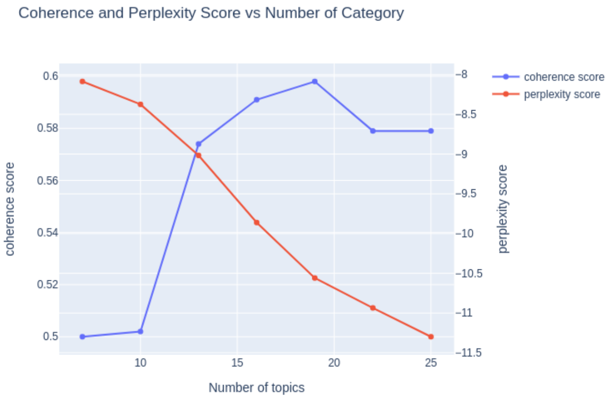

Figure 2: Scores vs Category of Models

Discussion

Text data despite its straightforward understanding is one of the hardest data type applicable to numerical based methods, namely any Machine Learning application. Thus, methods which inherently represents the words in the numerical domain has advantages over others while working with the text data. That's why LDA with its compact representation of the large text was our best choice for categorizing trending topics. The other models such as linear, non-linear classifiers would require decent and consistent representations of the text in the first place. On the other hand, any supervised method would require a large number of labelled data to perform in reasonable quality. For such big data we had, it was almost impossible to label them by a human. Hence LDA, an unsupervised method, was a better choice for our purpose.

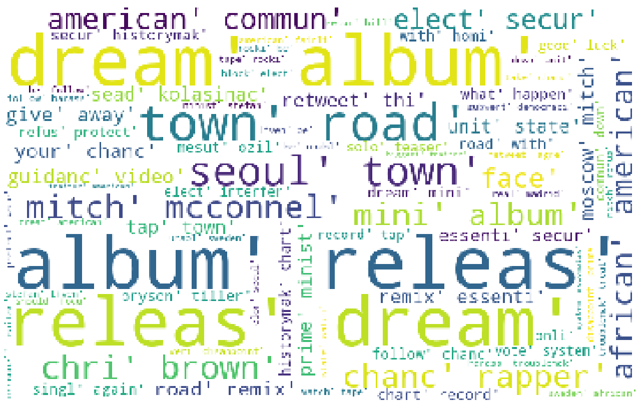

Figure 3: Music Category Keywords

In the result section and in figure 4, we numerically and visually showed how good our method is. For the sake of being complete, we also assessed our model with a human judge. In that, we prepared a test data which as human we know in which category they belong. This test data was never included in the training procedure.

The model found, for example, the trend topic **teamusa** is categorised as **Sports** with the text *rt bad defending of the player on the back shoulder for both teams make it v and too focused on the ball dont like the switched to for couldnt shut down the space that the mid bc a player down in mf*. Another example, **chance** is categorized under **Entertainment** with text *rt follow amp rt for a chance to win custom youngblood nintendo switch consoles for you and a friend*. For some trends and words, categorization was quite hard since they appear under almost every category as an example of **trump**. For those spam words, (too common) we made a list and ignored them while deciding the category of the clusters.

Conclusion and Future Work

This project was done and results presented here cover only one month of raw tweet date. Next step will be to run the same pipeline with more, if not all of, data and to examine if the performance improves or not. In general, for machine learning applications, more data accompany with better outcomes, however, Twitter trends both have time and geography localities meaning that topics changes very fast in time, eg in times of an election everything is related to the election whereas in regular times topics distribute more balanced. It may be interesting to check models covering full data and partial data and to compare the quality of results. We may also compare our result with other states of the art models like Twitter LDA or Gaussian Models [13]. Last but not least, we can always improve by better preprocessing and cleaning of the data

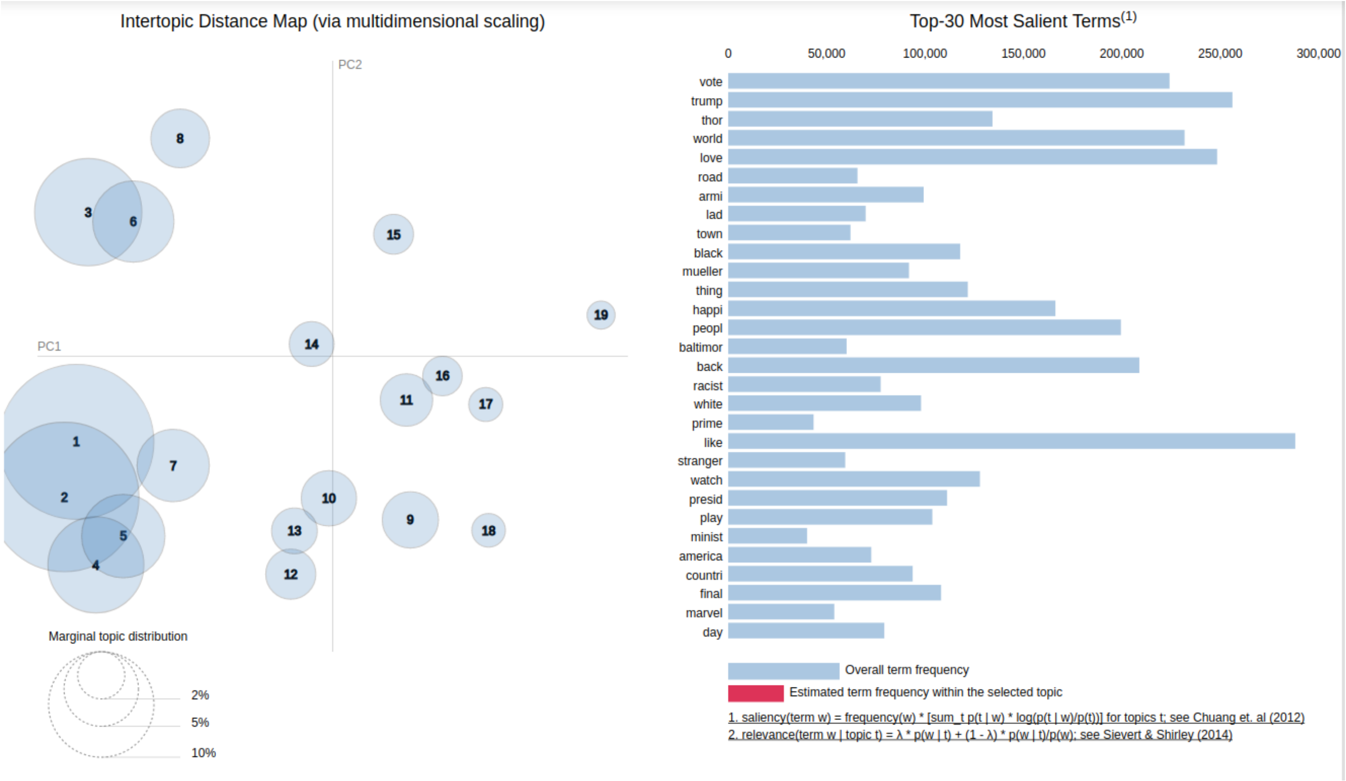

Figure 4: Category Clusters and Word Distribution

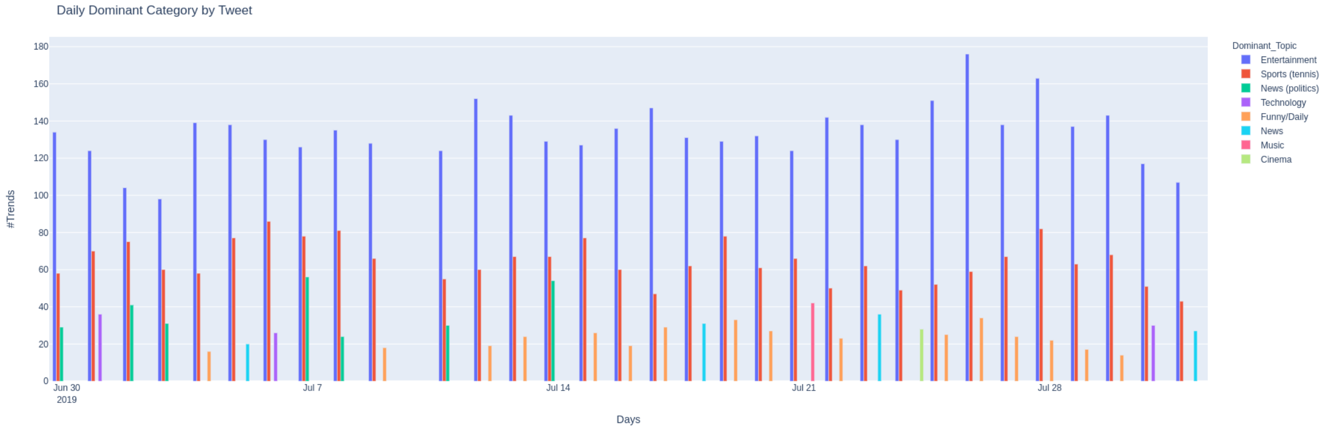

Figure 5: Number of Topics in Each Day in The Category

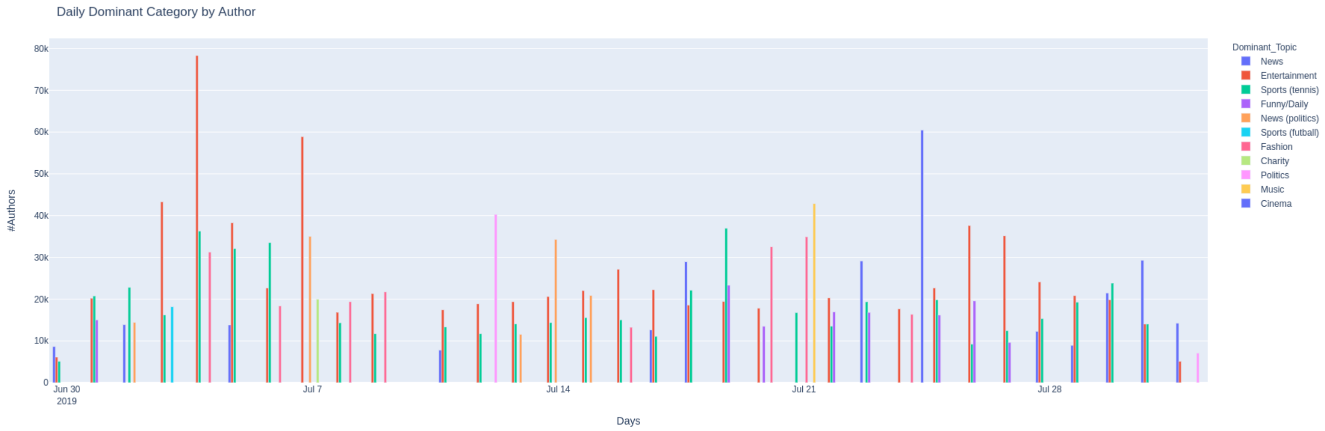

Figure 6: Number of Authors Tweeted in Each Day for The Category

References

[1] Billal Belainine, Alexsandro Fonseca, and Fatiha Sadat. “Named entity recognition and hashtag decomposition to improve the classification of tweets”. In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. 2016, pp. 102–111.

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. ” O’Reilly Media, Inc.”, 2009.

[3] Yan Chen et al. “Search engine reinforced semi-supervised classification and graph-based summarization of microblogs”. In: *Neurocomputing* 152 (2015), pp. 274–286.

[4] Jinan Fiaidhi et al. “Developing a hierarchical multi-label classifier for Twitter trending topics”. In: *International Journal of u-and e-Service, Science and Technology* 6.3 (2013), pp. 1–12.

[5] Liangjie Hong and Brian D Davison. “Empirical study of topic modeling in twitter”. In: *Proceedings of the first workshop on social media analytics*. 2010, pp. 80–88.

[6] Haewoon Kwak et al. “What is Twitter, a social network or a news media?” In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 591–600.

[7] Kathy Lee et al. “Twitter trending topic classification”. In: *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE. 2011, pp. 251–258.

[8] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[9] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. `http://is.muni.cz/publication/884893/en`. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[10] Hugo Rosa, João Paulo Carvalho, and Fernando Batista. “Detecting a tweet’s topic within a large number of Portuguese Twitter trends”. In: *3rd Symposium on Languages, Applications and Technologies*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2014.

[11] Koustav Rudra et al. “# FewThingsAboutIdioms: Understanding idioms and its users in the Twitter online social network”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2015, pp. 108–121.

[12] Ma Shiela C Sapul, Than Htike Aung, and Rachsuda Jiamthapthaksin. “Trending topic discovery of Twitter Tweets using clustering and topic modeling algorithms”. In: *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE. 2017, pp. 1–6.

[13] Guangxu Xun et al. “Topic discovery for short texts using word embeddings”. In: *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE. 2016, pp. 1299–1304.

[14] Muhammad Bilal Zafar et al. “On the wisdom of experts vs. crowds: Discovering trustworthy topical news in microblogs”. In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 2016, pp. 438–451.

[15] Wayne Xin Zhao et al. “Comparing twitter and traditional media using topic models”. In: *European conference on information retrieval*. Springer. 2011, pp. 338–349.