TRAFAN: Road Traffic Analysis By Using Social Media Web Pages

Akilesh B
IIT Hyderabad
cs13b1042@iith.ac.in

Nagendra Kumar
IIT Hyderabad
cs14resch11005@iith.ac.in

Bharath Reddy
IIT Hyderabad
cs13b1038@iith.ac.in

Manish Singh
IIT Hyderabad
msingh@iith.ac.in

Abstract

In this paper, we present TRAFAN, a system to analyze traffic activities of various cities. Traffic information is publicly available in various social networking sites. Our aim is to analyze the streaming transportation data moving through social networking sites, such as Facebook. We take the data from various Facebook traffic pages as input to our system and after processing this raw data, we allow users to perform three important functions: fast browsing through keyword search result, compare Facebook pages and get statistical summaries about the Facebook pages. By looking at the results, users will be able to conclude which traffic issues are the most chaotic in a city and they will be able to see all problems in that city related to a particular issue. Users will also be able to see which issue is more severe in a city in comparison to other cities. We perform our experiment on Facebook pages containing 21,000 posts and 0.5 million reactions. In addition, we present the results using information visualization techniques to improve the usability of our proposed TRAFAN interface.

Introduction

Road traffic analysis and prediction is a critical component of modern smart transportation systems. Popular Social Networking website, Facebook allows users to create page for businesses, brands and organizations to share their stories and connect with people. There are several government organizations who have pages in Facebook where they share the important traffic information in form of posts. These posts can be many types like general safety information (wear helmets, no high beam light), accidents, water logging, live traffic updates, reasons for traffic disruption, traffic diversions etc. This important information can be used by government to note down which are the major issues in cities. However, these posts are in disorganized and haphazard manner. There is lack of usability so user/organization cannot make any good decisions based on this information. User will not be able to see all the popular or recent posts which are related to a particular topic even if he is interested to see only those posts. If user wants to search popular posts related to a particular topic then the user needs to search through the entire page. User will not able to see which issue is trending in that page and user will not able to compare the intensity of an issue across the multiple cities.

Example 1 *Let's consider a user who wants to see all the popular/recent issues related to water logging in Delhi Traffic Police page*[1]*. For this information the user has to browse through the entire Delhi Traffic Police Facebook page, which has thousands of posts on all kinds of traffic related issues. It is very difficult for the user to search and browse through the posts that only deal with water logging. Moreover, if the user wants to compare the water logging problem across multiple cities, the user has to go to multiple Facebook pages that are maintained by each city and browse through the relevant posts. If the user wants to get a statistical summary about any specific traffic issue, the user has to go through all the posts on that topic manually and then himself infer the statistical summary. This is very difficult task as there are hundreds of posts on each topic.*

In this paper, we describe TRAFAN, a generic interactive interface. Our focus is to create a system by which users can get the information related to these traffic pages in a visual form. Using our system users can become aware of various accident prone zones, such as the frequent types and causes of accidents in each area presented in the form of novel pop-up summary. Government authorities, NGOs and local residents can use our dashboard to see popular traffic issues in each locality, such as dog menace, wrong parking, water logging, road works, etc. We use historical data analysis to identify the top problems that are being faced in each locality. TRAFAN allows users to explore large volume of data in a user-friendly, systematic and usable form (Singh, Nandi, and Jagadish 2012; Liu and Jagadish 2009; Wu et al. 2007; Jagadish et al. 2007).

To get the data from these Facebook pages we use

[1] https://www.facebook.com/dtptraffic/

Facebook Graph API[2]. We create a user interface to allow users explore the data. User can get all the posts related to particular topic within milliseconds. User can analyze how a particular issue is affecting the city over a period of time and compare the intensity of an issue across multiple cities.

Data Acquisition and Summarization

In this section, we describe about the dataset used for the experiment and its summarization.

Dataset and pre-processing

The first step of TRAFAN is to get the data from various Facebook traffic pages. To get the data from Traffic pages, we use Facebook graph API. We collect the 21,000 posts and 0.5 million reactions from the traffic pages. We remove the stop words (a, an, the) which do not contain important significance to be used in search queries. We also remove the names which are not in English dictionary. Furthermore, we perform stemming and lemmatization (Manning et al. 2014) on collected dataset. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For example, words like cars car's cars' will be replaced by word car and fish fishes fisher will be replaced by word fish. It increases the effectiveness of retrieval process.

Data Summarization

When search results are comprised of items that have a large amount of information, it is not possible to show the complete information of each item in the main result panel. For example, if the result consists of text documents or high-dimensional structured data, then instead of showing the full information, it is considered more desirable to present only a short summary of each result item (Manning, Raghavan, and Schütze 2008; Tombros and Sanderson 1998; Huang, Liu, and Chen 2008), so that users can use those summaries to determine if a particular search item is relevant to them.

There are two broad approaches for creating snippets: *static* and *dynamic*. A static approach returns the same snippet regardless of the query. On the other hand, a dynamic approach returns a customized snippet, which is based on user's information need as deduced from the query. Although static solutions are very efficient because one can create those summaries once at the time of inserting (or indexing) the data, and then retrieve and present it at the time of displaying the search results. But this static approach is not desirable because it does not give any insight to the user of why a particular result is relevant to user's need. An informative snippet can reduce user's pain to manually go through details of each search result. However, creating a dynamic summary requires more run-time computation. Moreover, it is hard to guarantee a good quality dynamic summary. So, we use both dynamic and static snippet to summarize our search results.

[2]https://developers.facebook.com/docs/graph-api

Methodology

Many users get bored reading the long posts including messages and subsequent comments. Our task is to summarize the post in an effective manner to give a gist of the post so that it can convey the message in an effective manner.

Now, we combine the message and following comments into a single field. For summarizing the posts, we try to use the standard TF-IDF algorithm and our proposed algorithm. We compare the results of both algorithms.

Post Summarization using TF-IDF Algorithm

TF-IDF (Manning, Raghavan, and Schütze 2008) is a numerical statistic, often used in information retrieval and text mining, which reflects how important a word is to a document in a given collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. TF-IDF weight is composed by two terms: the first computes the normalized Term Frequency (TF), computed as number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus, divided by the number of documents where the specific term appears. TF-IDF weight is the product of these two quantities.

After combining the message and following comments into a single text, we use TF-IDF algorithm to get top six keywords for summarizing the post.

Example 2 *Consider the following Facebook post which contains message and comments.*

Message*: Heavy goods vehicle (12 tyres) caught in drunken drive in the limits of Tr. PS Tirmulghiri and the driver was convicted for 3 days imprisonment.*

Comments *(separated by #): Three days not sufficient for that person # Good job but 3 days is not fair # Please arrest wrong side driver of bike 3 days r nothing atleast 15 days # Only 1 u thing u did a great job its not enough u hav to work more In 3 days teach them importants of life I requiesting u sir # what a joke@Hyderabad traffic police: Sir please do something to stop heavy trucks going on bridges especially during night # Road king is now jailed king 3 days # Good work but the punishment shall be : Seizure of vehicle and suspension of Driving Licence # He was so dangerous he could kill many I demand a revision against the order of the court # Is it a joke, 3 days, The court gives more than that to a driver of 2 wheeler and 4 wheeler a 12 wheel heavy vehicle driver only gets 3 days*

 # Sorry Thats not sufficient why dint the state represented by concerned police file revision against the order for harsher punishment #

Keywords: {`days, sufficient, job, highways, suspension, revision`}

The above are the top-six keywords, which are generated by using TF-IDF algorithm. These combinations of words are also called as summarized post. Traditional TF-IDF algorithm does not seem to work well. We cannot make much sense out of summarized post.

Post Summarization using Word Priority

Algorithm 1 Post Summarization using Word Priority

Input: K: the number of summary keywords
M: a Facebook post message
C: Set of all the comments to message M
Output: S: top-K keywords that summarize message M
Method:

1: Store each unique word of the message M in a dictionary D with the word as the key and the number of times it occurs in M as the corresponding value.

2: Take a particular word (key) from the dictionary D and scan through the comments C, if there is match then increment corresponding value by one for every match.
3: Get the total number of times every unique word from the post has appeared in the post and following comments.
4: Sort the dictionary D based on value.
5: return $S \leftarrow$ top-K words. =0

We illustrate our algorithm using Example 2, which we used to illustrate TF-IDF algorithm and we compare the result of both the algorithms.

Dictionary D = {(3, 7), (driver, 2), (days, 7), (imprisonment, 1), (heavy, 3), (goods, 1), (vehicle, 2), (drunken, 1)}

Keywords = {3, days, heavy, vehicle, driver, imprisonment}

The above are the top-six keywords that are generated by our proposed algorithm. If we compare these keywords from the keywords we got using TF-IDF algorithm, then we found that these keywords make much better sense. By looking at these keywords, we can get a feel what this post is about. These combinations of words are also called summarized post. We do not get better results by using TF-IDF algorithm because algorithm treats message and comments equally and shows some unimportant keywords which are not part of the message. For example, if there is any dispute in comments and one word repeats many times, then TF-IDF results that keyword even though it is not an important keyword. In our algorithm, we summarize the posts by using the keywords which are there in the message. We take support from the comments also. Thus, we generate only important keywords.

System Evaluation and Results

To evaluate our system, we use mean average precision (MAP) and normalized discounted cumulative gain (NDCG) (Liu et al. 2007). These are the information retrieval techniques used to evaluate the ranking of webpages. We apply these techniques to evaluate our search results. As we have unlabeled dataset, so we ask the students of our lab to manually label the search results. We choose 20 important traffic issues as search query and generate the search results using TRAFAN. We use top 30 search results of each query for evaluation and we get 0.9 MAP and 0.81 NDCG which indicates that our system generates good results.

Now, we discuss about the results generated by the TRAFAN. It show the information to the user in an efficient and concise manner. It shows the traffic information of various cities at one place in a well-organized manner. It has the following three application programs:

Keyword Search

All the cities have their own individual traffic related Facebook page. TRAFAN provides an integrated means to search traffic issues across these different Facebook pages. To support fast browsing through the search results, we create informative summary (i.e., snippet) for each post in the results, so that users can look at the summary information. We provide options to choose a city and word. We display all the posts which are related to that particular word in the selected city. We also provide a way to search all the popular and recent posts related to any query (word). This feature help users to find the posts which receive lot of users' attention (audience reactions) on an issue. By default, we show the posts based on their creation time as shown in Figure 1.

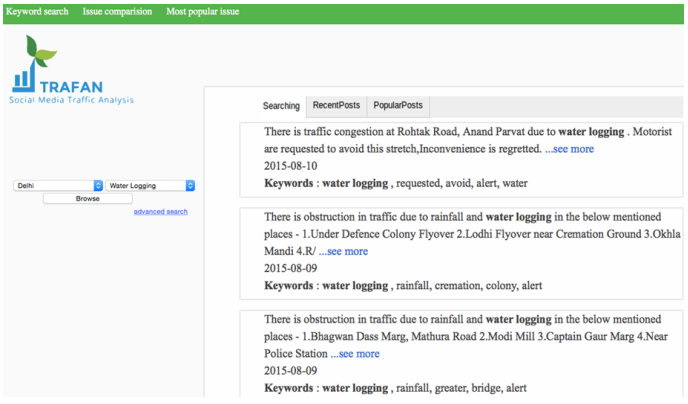

Figure 1: Keyword search

We present snippet of the post so user can get a gist

of the post by looking into the snippet. We do not show the whole post as many of them are lengthy. We show both static and dynamic snippets. To show static snippet, we present top-six most relevant keywords by looking at these keywords user can understand about the post. We show dynamic snippet of post also for that we show query keyword and one or two line before and after that keyword. If user still does not understand, we also provide see more option on clicking that option, user is redirected to the actual Facebook post.

Popular Issue

There are some topics (issues) on Facebook traffic pages that get more number of reactions. There are high chances that these are the critical issues i.e., breakdown, accidents, water logging etc., so they need to get more attention. Sometimes user/organization also wants to see popular issues of the city. So we present these issues in graphical format, as shown in Figure 2. To analyze most popular issues, user can select the city and type of graph (line, curved line, area, curved area). Based on user input, we plot the graphs of most shared posts, most commented posts and most liked posts over a period of time. When user hovers over the peak of graph, a tool tip appears which shows the corresponding post's message.

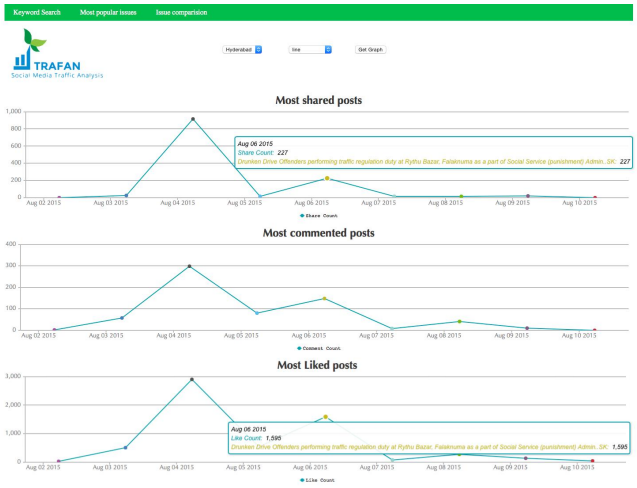

Figure 2: Most Popular Issues

Issue Comparison

TRAFAN allows users to get a statistical summary on an issue across cities as shown in Figure 3. User/organization can know which issue is severe in the city as compared to other cities. As this feature allows the organizations to compare the intensity of an issue in multiple cities, they can take an appropriate action to fix the issue. TRAFAN provide an option to select cities and word (issue). It compares the number of posts related to the issue across all the selected cities and present the result in form of pie chart.

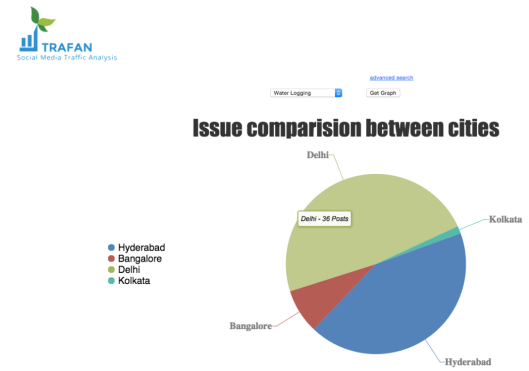

Figure 3: Intensity of an Issue Across Cities

Conclusion and Future Work

TRAFAN is a traffic analyzer which helps to analyze the traffic problems. It can be used by government organization to keep track of major issues in metro cities. It can also be used by individuals to know the various traffic issues of their localities. Our system is not only about the traffic, it analyzes various problems of the city like water, dog menace, wrong parking, accidents, road works and many more. Our future work is to use big data solutions to make it more usable, efficient and to extract the data from other publicly available sources like twitter.

References

[Huang, Liu, and Chen 2008] Huang, Y.; Liu, Z.; and Chen, Y. 2008. Query biased snippet generation in xml search. In *SIGMOD*, 315–326. ACM.

[Jagadish et al. 2007] Jagadish, H.; Chapman, A.; Elkiss, A.; Jayapandian, M.; Li, Y.; Nandi, A.; and Yu, C. 2007. Making database systems usable. *SIGMOD*.

[Liu and Jagadish 2009] Liu, B., and Jagadish, H. 2009. Using trees to depict a forest. *VLDB*.

[Liu et al. 2007] Liu, T.-Y.; Xu, J.; Qin, T.; Xiong, W.; and Li, H. 2007. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR 2007 workshop on Letor*, 3–10.

[Manning et al. 2014] Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, 55–60.

[Manning, Raghavan, and Schütze 2008] Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

[Singh, Nandi, and Jagadish 2012] Singh, M.; Nandi, A.; and Jagadish, H. 2012. Skimmer: rapid scrolling of relational query results. In *SIGMOD*. ACM.

[Tombros and Sanderson 1998] Tombros, A., and Sanderson, M. 1998. Advantages of query biased summaries in information retrieval. In *SIGIR*. ACM.

[Wu et al. 2007] Wu, T.; Li, X.; Xin, D.; Han, J.; Lee, J.; and Redder, R. 2007. DataScope: viewing database contents in Google Maps' way. *VLDB*.