Review of Crowdsourced Exploration of Mobile App Features: A Case Study of the Fort McMurray Wildfire

Andrew DiStasi
Rochester Institute of Technology
Rochester, NY
add5980@rit.edu

1. PAPER OVERVIEW

In this paper, the authors examine the effectiveness of crowd sourcing as a means of suggesting and ranking mobile application features, using the 2016 Fort McMurray Wildfire as a case study. To do so, they propose a method called MAPFEAT, which used machine learning to analyze tweets from individuals affected by the Fort McMurray wildfires to determine what features and information were most relevant to those individuals. MAPFEAT was trained using manual tagging of tweets and through volunteers on Amazon Mechanical Turk. It uses an 11-step process to classify inputted tweet data, query the app store with identified relevant terms, mine features from the app description, identify common features, map tweet topics to common features, and use crowdsourced data to validate those features' inclusion.

After creating MAPFEAT and using it to extract a set of desired features, pre-existing applications were examined to determine which, if any, contained a significant number of these features. This was accomplished by searching the Apple iTunes App Store with "fire" and "wildfire" as keywords and filtering out irrelevant applications. This created a list of 26 applications that were related to wildfire emergencies. Feature extraction was performed on the descriptions of these applications, creating a list of 28 unique application features across the 26 applications. This list was compared with the list of desired features generated by MAPFEAT to identify how completely these pre-existing applications satisfied user desires. The paper concludes with examinations of threats to the experiment's validity, related work, and relevance and applicability of the experiment.

2. HYPOTHESES & MOTIVATION

The authors address three research questions in their paper (going so far as to explicitly enumerate them and discuss each one). First, they attempt to identify how tweets can be mapped into mobile application features automatically. They list this question as relevant because it has been historically established that mining Twitter and other social media sources is an effective means of managing emergency situations. Therefore, the ability to automatically mine the needs of individuals to appropriately support them with software functionality will have a relevant impact on the development of related applications. Second, they attempt to examine how the features mined from MAPFEAT in the Fort McMurray case study compare to features provided by existing wildfire apps. The authors posit that this is relevant because it allows for evaluation of MAPFEAT and that it offers real world benchmarks for comparison. Finally, the authors ask How the features generated by MAPFEAT will be perceived by the general public (specifically in the Fort McMurray case study).

This also allows for evaluation of the MAPFEAT system and enables the authors to further identify the perceived importance of application features to members of the general public. The motivation behind this paper is relatively simple and, in my opinion, fairly obvious. Creating a system that can automatically identify and suggest functionality enhancements to emergency applications addresses a need that can potentially be the difference between life and death. While the ability to perform feature extraction of natural language data to improve application functionality is inherently valuable, doing this with applications that aid users in the event of a crisis or disaster has an apparent moral and social benefit in that it can facilitate the connection of impacted individuals to emergency services and the can allow for individuals to connect with each other.

3. EXPERIMENT METHODOLOGY

Before any work could be completed on MAPFEAT, the authors began by aggregating a set of 69,680 unique tweets regarding the Fort McMurray wildfire from a period of May 2nd through May 7th, 2016. This dataset was mined by using the Twitter Search API to obtain all publicly available tweets that included "#ymmfire," "#FortMacfire," and "#ymm." After aggregating the data, the authors randomly selected 2% of the tweets to manually analyze. This manually tagged data was used to explore the study's usefulness at a high level and to train a Naïve Bayes classifier to separate informative tweets from non-informative tweets.

As is common in any text mining, the data was preprocessed to prepare it for use in machine learning tasks. The authors made use of Pattern, an NLTK python package, to eliminate retweets, eliminate hashtags, emojis, and special characters, eliminate URLs, eliminate duplicate tweets, and lemmatize tweets to offer a reference to both the dictionary form of the word and the original context of the word. After this process was completed, an nine step MAPFEAT process to map tweets to application features using was implemented. The pre-processed tweet data was classified by the Naïve Bayes classifier mentioned in the previous paragraph to create a dataset of informative tweets. Topic modeling, using LDA and the Gensim python library, was used to extract common topics to generalize the needs expressed in the tweets. Each topic was comprised of a cluster of words that frequently appeared together frequently in the text corpus. For example, "gas," "traffic," and "map" were grouped together as one topic.

These topics were used in an automated three step process to generate a list of applications from which features will be mined. First, a set of search queries (limited to at least two-word queries) were generated from the combination of words in each topic. These queries were then sent to the Apple iTunes application store using the store's API. The descriptions of the top applications retrieved by each query were retrieved by each query. These app descriptions were then mined for features using a method that extracted "featurelets," defined as "a set of commonly occurring co-located words, identified using NLTK's N-gram CollocationFinder package," from each description. Featurelets were clustered to aggregate features that were

measured to be at least 60 % similar, creating a set of bigrams and trigrams that were treated as desired application features.

Next each search query was mapped to a set of features that were shared between a specified minimum number of applications retrieved from the query. This step was performed because a specific keyword retrieves a variety of applications, meaning that the commonality between their features likely holds value and indicates why they were retrieved by this query. Each of these feature groups were mapped to the original tweet topic and presented to five Amazon Mechanic Turk workers who were asked to select features relevant to the tweet topic. This allowed the authors to identify and exclude any mismatched features from the topic group. Finally, the original tweet topics were mapped to a feature set, defined as a set of all features that were received from every search query originating from that tweet topic.

The final step of the MAPFEAT process involves a crowdsourced evaluation that validates the accuracy and value of the results. Crowdsourcing was sued to evaluate the validity of the semantic relationship between each tweet topic and it's resulting features, created in the previous step. This step was automated through use of the Amazon Mechanical Turk API, submitting extracted features along with their corresponding tweet topic, asking users to select all features relevant to the topic. If the majority ranked a feature as not relevant to a topic, that feature was eliminated from the mapping set. Given the extensive amount of automation and validation present throughout the experiment, I don't see any real flaws or limitations in the manner the authors approached the experiment. The only potential issue I can identify is in the data collection process, as the Twitter API offers limited amounts of tweet data per query and only allows tweets that are less than two weeks old to be returned. However, the authors directly mention this shortcoming in a later section, and took steps to automate querying to allow for the inclusion of as many results as possible in the dataset.

4. DATA GATHERING & ANALYSIS

As previously mentioned, the data utilized was a collection of 69, 680 unique tweets accessed from the Twitter Search API that were determined to be about the Fort MacMurray wildfire (identified by the inclusion of "#ymmfire", "#FortMacfire", or "#ymm"). The data was analyzed at several stages throughout the course of the experiment. Initially, 2% of the dataset was manually analyzed by the authors and used to train a Naïve Bayes classifier that could distinguish informative tweets from non-informative tweets. Non-informative tweets were those defined as lacking explicit potential to be mapped into a software feature (because they convey no need or requirement). An example given in the paper is "All my thoughts are with all the people who are affected by the devastation of the fire. #ymmfire #FortMacFire." A Naïve Bayes classifier was used because it has been proven to perform well with smaller datasets and has been suggested by multiple other studies as the ideal classifier for tweet classification.

Through the MAPFEAT process discussed in the previous section, this tweet data is mapped to application features and is evaluated in terms of validity and accuracy through crowdsourcing. The authors considered the crowd to be representative of the general public and surveyed them to evaluate features proposed through MAPFEAT. Members were asked to imagine their hometown was being destroyed in a wildfire and that they had their smart phone to help them with this situation. They were presented with a series of features (generated through MAPFEAT), and for each feature was asked how important they felt that feature would be as part of an emergency wildfire application. Choices included "Essential," "worthwhile," "unimportant," "unwise," and "I don't understand." To improve data quality, incomplete and low quality results were filtered out. Low quality results were defined as results where the respondent took less than 20 seconds answering a question or selected the same answer for over 90% of the questions. This dataset was discretized into one of the four groups based on which answer was selected most commonly. This data was used in the evaluation of the results for the features mined by MAPFEAT. I find this approach reasonable for a number of reasons. First, steps were taken multiple times throughout the application to ensure data quality. Additionally, the data was evaluated with a dataset comprised of crowdsourced results, which works to remove the possibility that the results are impacted by the bias of authors. Finally, the authors implemented processes to continually refine the data throughout the process.

5. RESULTS

To analyze the effectiveness of the result set, the features extracted through the research process with MAPFEAT were compared to features found in pre-existing applications designed for use in wildfire emergencies. There are three scenarios possible in these comparisons. First, that the feature mined by MAPFEAT exists in the current wildfire application. Second, that the feature mined by MAPFEAT does not exist in the current wildfire applications. In this case, MAPFEAT can offer suggestions on features needed by the general public to developers. Finally, the feature exists in current wildfire apps but was not mined by MAPFEAT. The authors interpret this scenario as a lack of justification for the inclusion of that feature or a result of an incompleteness in the data. If it is not the result of incompleteness, developers would be able to use these results to inspect the usefulness of application features.

The features from pre-existing applications were drawn from a list of applications resulting in queries of the iTunes search API using the keywords "fire" and "wildfire." These queries returned 86 applications, which was filtered down to 26 relevant applications. Between these 26 applications, 28 unique application features were found. The MAPFEAT results were then compared with this list of features. MAPFEAT extracted 163 features related to the tweets about the Fort McMurray wildfire. Of those 163, 139 features were not found within the existing applications (meaning that there are 139 features matching the needs of the general public that are not provided). Of the 28 identified features, MAPFEAT found 87.5% (24) available in existing applications. MAPFEAT mined 90% of all features that were present in multiple applications. Another round of crowdsourcing evaluation was performed to judge crowd perception of the value of features mined by MAPFEAT. 500 master workers using Amazon Mechanical Turk rated the importance of having each specific feature mined by MAPFEAT using the four categories discussed in the data gathering & analysis section. Of the 163 features MAPFEAT mined, 28% were classified as essential, 56.1% were classified as worthwhile, 14.3% were classified as unimportant, and 1.4% were classified as unwise. These results are promising, as 84% of the features detected by MAPFEAT were classified as either worthwhile or essential and of the four existing features that were missed by MAPFEAT, three ranked either unwise or unimportant. These results indicate that MAPFEAT is highly effective at identify mainly useful features and at identifying a wide variety of useful features.

6. LIMITATIONS

The authors directly address several threats to validity that I feel comprehensively address all potential limitations or threats. First, they identify that there may a disparity between the perceptions of the crowdworkers and those who have experienced a wildfire, but they argue that the context of the questions is related closely enough to general fire safety that the workers should be able to effectively understand. Additionally, individuals directly impacted by the Fort McMurray wildfire were used to gather response data. A second issue posed is the reliability of the crowd responses, but the authors argue that 500 responses is enough to exclude randomness and that the methods used to ensure data reliability (excluding low effort responses) are sufficient. Finally, the issues of internal validity surrounding search results (hashtags used in twitter search, validity of the application store search, and assumptions made about parameters during the MAPFEAT process) are acknowledged, but the authors state that multiple checks and validations were used to minimize the impact of these limitations.