Summarizing Microblogs for Emergency Relief and Preparedness

Kanav Mehra[1] and Vibhash Chandra[1]

Indian Institute of Engineering Science and Technology, Shibpur, India
kanav.mehra6@gmail.com

Abstract. In recent years, Online Social Media (OSM) has established itself as one of the most significant sources of situational information during any natural or man-made disaster. Real-time summarization of this rapidly posted huge volume of crowdsourced responses is a common requirement for emergency relief and preparedness when time is critical. Our semi-automatic method exploits a combination of SumBasic Summarizer and different classifiers to summarize the topic wise relevant microblogs (tweets) extracted through manually identified query term matching. The result of our participation in the SMERP 2017 Data Challenge Track shows that it is an effective approach in summarizing tweets in a disaster scenario and can be replicated across diverse domains.

Keywords: Microblog Retrieval, Summarization, Emergency Relief

1 Introduction

Recent disasters like Nepal Earthquake, Chennai Floods, and others have proven the key role of social media in providing situational information. Thus, voices of social media can no longer be ignored in emergency relief and preparedness. However, important information is typically obscured by a lot of personal opinions, emotions, and prayers for victims. More than often, tweets conveying similar information are written in a diverse style and the context of the posted content is time variant. This raises a challenge in developing efficient summarization techniques, to provide summarized, micro-level information from the huge pool of passively posted contents to aid decision makers. The SMERP 2017 Data Challenge Track is motivated by this scenario, and aspires to endorse the development of dynamic and adaptive summarization system to provide meaningful insights from microblogs posted during disasters.

A few approaches for online summarization of tweet streams have recently been proposed [1–3]. In present work, we demonstrate that topic-wise summarization of tweets during disaster events can be better accomplished by initially summarizing the tweets using some basic summarizer, then categorizing the relevant and non-relevant tweets, and finally summarizing again.

2 Data and Resources

The SMERP 2017 Data Challenge Track has provided a dataset of microblogs posted during the earthquake in Italy in August 2016 and a set of 4 topics reflecting the practical information need to be addressed in an emergency situation. Data is provided on

two levels. In Level 1, 52K tweets of the first day (24 hours) after the earthquake and in Level 2, 20K tweets collected during the second day (24 hours) after the earthquake is provided.The topics are in TREC format, each of which contains three parts: title, brief description, and a more detailed narrative on what type of tweets will be considered relevant to the topic. An example of the TREC format topic is given below:
<top>
<num> Number: SMERP-T1
<title> WHAT RESOURCES ARE AVAILABLE
<desc> Description:Identify the messages which describe the availability of some resources.
<narr> Narrative: A relevant message must mention the availability of some resource like food, drinking water, shelter, clothes, blankets, blood, human resources like volunteers, resources to build or support infrastructure, like tents, water filter, power supply, etc. Messages informing the availability of transport vehicles for assisting the resource distribution process would also be relevant. Also, messages indicating any services like free wifi, SMS, calling facility etc. will also be relevant. In addition, any message or announcement about the donation of money will also be relevant. However, generalized statements without reference to any resource would not be relevant.
< /top>

3 Summarization Methodology

Our run was semi-automatic that includes a system with manual intervention in query formulation stage and relevance judgment stage before training the classifier. Our method of summarization is extractive.

3.1 Method1

Preprocessing: We have preprocessed the set of tweets by removing a standard set of English stopword and case-folding. Subsequently, we focus on particular end-markers (e.g., !, ., ;) to split a tweet into multiple fragments. Further, we have removed the set of tweets (or, fragments) whose length fall below a certain threshold, say α, i.e., the set S= $\{t \in T$ s.t $len(\text{t}) \leq \alpha\}$ where T is the complete set of tweets and $len(t)$ is the tweet length in terms of words. We have experimented with a set of values of α = $\{1, 2, ..., 10\}$ and the best result is obtained for α = 4. Tweets having less than 4 words typically do not contain any significant information.

Query Generation and Extraction: The query is generated manually by selecting specific terms from the description and narrative of the topic, which are intuitively important and likely to be present in the tweets relevant to the topic. We have used the python nltk implementation of the lemmatizer to lemmatize the set of specific terms (query) generated previously. The relevance of a tweet for a particular topic is judged by the presence of at least one of the query term and hence extracted.

Initial Summarization: We have customized and used the SumBasic summarization algorithm [1], that works on the frequency based sentence selection component. It min-

[1] https://github.com/EthanMacdonald/SumBasic

imizes redundancy through a component to re-weight the word probabilities. For every sentence, SumBasic calculates average probability of occurrence of the words in the sentence and accordingly assigns a weight. It selects the best scoring sentence that inherently includes the highest probability word. For our current submission, we have manually selected a subset of terms from query, which contributed in extracting relevant tweet fragments in the previous phase. However, we have ignored a set of terms like 'pray', 'sad', which merely convey opinion or emotion. The customized SumBasic algorithm is used to summarize the set of relevant tweet fragments or sentences extracted in the previous phase. The score of a sentence is doubled each time when a specific term from the set of terms i.e., subset of the terms from query is present in the sentence. For example, terms like 'food', 'water', 'blood', 'tents' help in highlighting significant information regarding resouce need or availability and thus relevant sentences are selected by increasing the score of sentence containing these terms. Hence, the sentences with highest scores are considered to be most relevant to the topic The summarization algorithm needs as input the desired number of sentences in the summary, which in our case set as 10, 20, 50, 100 and 200.

Classification: We have used python nltk implementation of the Naive Bayes Classifier (NBclassifier)[2], i.e., based on supervised keyword extraction. Through this extraction method, we have learnt features of sentences that make them good candidates for inclusion in the summary. Since we are dealing with tweets in this task, we find it quite appropriate to consider different terms present in a relevant tweet or non-relevant tweet as a feature of that tweet.
We have selected around 20 tweets from the summaries of length 10 and 20 obtained by SumBasic summarization algorithm and classified them into two sets of tweets, namely, relevant and non-relevant. We have created a feature set according to the above mentioned approach by tokenizing each tweet present in the two sets into words. Further the features set is split into training and test data in 70:30 ratio.
We have trained our NBclassifier using the training dataset created earlier. Using the NBClassifier we have generated a list of words. If the presence of a word from the list of most informative features helps the NBClassifier in determining if the tweet is relevant, it is considered as a 'keyword' and if its absence is the deterministic factor, then it is considered as a stopword. In this way we have two list of words, namely keywords and stopwords.

Final Summarization: In this section summaries obtained from Initial Summarization phase are combined. Duplicate tweet fragments or sentences are removed, if measured cosine similarity between the tweet fragments present in the summaries is greater than the threshold value of 0.7. Thereafter, the relevance score of tweet fragments is calculated by using the two list of words obtained after Testing phase. If a sentence contains a keyword, its score was doubled and if it contains a stopword, score was divided by 1.8. The reason for choosing this random value was that, even if a tweet contained a stopword, we did not want to nullify the presence of keyword in the tweet by ignoring the tweet completely due to the presence of a stopword. The final summarized docu-

[2] `https://pypi.python.org/pypi/nltk`

ment contains the sentences with the highest relevance scores, such that the length of the whole document is in the range of 300 words.

3.2 Method2

Preprocessing, Query Generation, Extraction and Initial Summarazation phases are done in the similar way as described in *Method 1*. From the Classification phase few changes were made in the methodology as mentioned below.

Classification: Rather than just training an NBClassifier as in the previous method, we have trained and tested an ensemble classifier, in the same way as done in the Classification phase of method 1. Our Ensemble Classifier contains these classifiers - Naive Bayes Classifier, Multinomial Naive Bayes Classifier, Bernoulli Naive Bayes Classifier, Logistic Regression, Linear Support Vector Classification, Stochastic Gradient Descent Classifier, implementations of which are obtained from sklearn package present in python [3].

Final Summarization: For each tweet fragment present in the initial summary, we have classified it on the basis of voting done by all the classifiers in the Ensemble classifier. If the tweet is considered relevant by at least 4 of the classifiers, then the tweet is qualified to be in the final summary. We have classified the tweets in this way till either the word limit is exceeded or no more relevant tweets could be obtained from the dataset.

3.3 Methodology Enhancement in Level-2:

In the SMERP 2017 Data Challenge Track, organizers have released the data in two levels as illustrated in Section 2. In each level runs are submitted independently and the notion was to aid participants to learn from the feedback of Level-1 and enhance their method to perform bettter in Level-2. Thus, we have improved our methodology after receiving the initial evaluation result of your submission in Level-1 along with relevant portions of our summaries highlighted by the reviewers. We have used this feedback primarily to alter the basis of our term selection for manual query formation. In Level-1, for both the *Query Generation and Extraction* and *Initial Summarization* phase, as described earlier we have chosen a set of query terms and subset of query respectively by manually identifying precise terms from the description and narrative of the topic, which are intuitively important and likely to be relevant to the topic .However, in Level-2 terms are chosen by selecting specific words from the relevant portions of our Level-1 summary. Likewise we also used the relevant and non-relevant portions of our summary to train the NB classifier in the *Classification* phase at Level-2.

4 Experimental results

Results of our two runs are shown in Table 1. The performance of the submitted runs is evaluated in terms of Rouge scores. The methodology used for generating both the runs are similar in nature. However, the classifier used are different as described in Section 3.

The best result is obtained for Run1 that used Naive Bayes classifier, whereas the other run used ensemble classifier. It is also evident from the table that performance of

[3] `http://scikit-learn.org/stable/`

Run type	Team Id	Run Id	Recall ROUGE 1	Recall ROUGE 2	Recall ROUGE L	Recall ROUGE SU4
Semi-automatic (Level-1)	IIEST	Run1	0.5109	0.2824	0.4885	0.2329
Semi-automatic (Level-1)	IIEST	Run2	0.4589	0.2433	0.4375	0.1983
Semi-automatic (Level-2)	IIEST	Run1	0.5540	0.2436	0.5142	0.2864
Semi-automatic(Level-2)	IIEST	Run 2	0.5187	0.2512	0.4796	0.2505

Table 1. Evaluation Results of Our Runs

our system has significantly improved in Level-2. Since, after getting initial feedback of Level-1 submission, in Level-2 for both for extraction and basic summarization phases, we selected the terms from the relevant portions of our Level-1 run.

5 Concluding Discussion

In this paper, we presented a brief overview of our approach to summarize the topic wise relevant microblogs (tweets). We have observed that combination of a basic summarization algorithm with a classifier can generate fairly good performance. It is evident that query term generation method plays a vital role in defining the terms that represent the most relevant information contained in the dataset.Accordingly, our semi-automatic run obtained the overall first place.

As a future work, we would like to automate the query term generation method and explore the more sophisticated combination of summarization algorithm and classifier to improve the performance of our system.We also plan to extend our work to summarize the situational information location-wise and organization-wise to aid the emergency relief operation.

References

1. Khan, M.A.H., Bollegala, D., Liu, G., Sezaki, K.: Multi-Tweet Summarization of Real-Time Events. In: Socialcom (2013)
2. Shou, L., Wang, Z., Chen, K., Chen, G.: Sumblr: Continuous summarization of evolving tweet streams. In: Proc. ACM SIGIR (2013)
3. Zubiaga, A., Spina, D., Amigo, E., Gonzalo, J.: Towards Real-Time Summarization of Scheduled Events from Twitter Streams. In: Hypertext(Poster) (2012)