IIT BHU at FIRE 2017 IRMiDis Track - Fully Automatic Approaches to Information Retrieval

Harshit Mehrotra
Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi 221005
harshit.mehrotra.cse15@iitbhu.ac.in

Ribhav Soni
Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi 221005
ribhav.soni.cse13@iitbhu.ac.in

Sukomal Pal
Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi 221005
spal.cse@iitbhu.ac.in

ABSTRACT

This paper presents the work of the team of IIT (BHU) Varanasi for the IRMiDis track in FIRE 2017. The task involved classifying tweets posted during a disaster into those expressing need and availability of various types of resources, given some tweets from the Nepal 2015 earthquake. We submitted two runs, both of which were fully automatic.

KEYWORDS

Information retrieval, microblogs, disaster, Lucene ,query generation

1 INTRODUCTION

With the increasing impact of social media, websites like Twitter which provide microblogging services have become increasingly popular. Apart from acting as a window to the outside world, these also serve as an important means to communicate and collect information, especially in times of emergency/disaster. The IRMiDis track in FIRE 2017 [5] posed a challenge to work on such data collection and analysis purposes. Specifically, the task was to develop IR methodologies to classify tweets as:

- Need-tweets: Indicating the need or requirement of some specific resource such as food, water, medical aid, shelter, to name a few. Tweets pointing to scarcity or non-availability of some resources also qualify for this category.
- Availability-tweets: Informing about the potential/actual availability of resources. The former may be speaking about resources being transported, or food packets being delivered.

A tweet may be both a need-tweet and an availability-tweet.

We submitted two runs, both of which were fully automatic i.e. no retrieval step involved manual intervention. Details of the runs are given in the subsequent sections.

2 DATA

The data contained around 70,000 microblogs (tweets) from Twitter that were posted during Nepal earthquake 2015, some of which were code-mixed, i.e., contained different languages and/or scripts. Around 20,000 of these were provided for development/training purpose and the remaining 50,000 for testing and evaluation.

3 OUR METHODOLOGY - RUN 1

The run is fully automatic in both query generation and searching. It makes use of Apache Lucene, a open source Java based text search engine library [1]. The run can be divided into the following steps:

(1) Cleaning and Tokenization:

The tweets in the training data are first cleaned to remove hashtags, numbers, addresses (of the type @...) and URLs. These objects are not deterministic of the category (Nepal-Need/Nepal-Avail) a tweet falls in. Many hashtags (like #earthquake, #nepal, #NepalEarthquake) can appear in tweets of any category. Following this, the cleaned tweets are tokenized using the Standard Analyzer, which indexes documents after converting each token to lowercase, and removing stopwords and punctuations, if any. [4] The frequency of each token in the training set is then recorded.

(2) Query Generation:

The token set of each category is modified to its set difference with the other token set. Then the queries for the 2 categories are generated as follows:

- Nepal-Avail: Disjunction of tokens with frequency more than or equal to 3 given weight of their respective frequencies divided by 3.
- Nepal-Need: Disjunction of tokens with frequency more than or equal to 2 given weight of their respective frequencies divided by 2.

The threshold frequencies are set in accordance with the number of tweets of each category present in the training set.

(3) Searching and Scoring:
The test set is also pre-processed and indexed like the training set in step 1. The test index is then searched for the queries generated in step 2. The scores are computed by Lucene. This scoring uses a combination of the Vector Space Model (VSM) of Information Retrieval and the Boolean model to determine how relevant a given document is to a user's query. [3]

(4) Categorization:
The scores returned by Lucene are normalized to (0,1) and tweets having scores >=0.1 and >=0.2 are considered appropriate for the categories Nepal-Avail and Nepal-Need respectively. It is seen in our experiment that since tokens for Nepal-Avail are selected for a greater threshold frequency, the corresponding search query gives suitable tweets even on a lower score, hence the above difference.

4 OUR METHODOLOGY - RUN 2

This run is also a fully automatic one, i.e., no retrieval step required manual intervention.

- The task was treated as a classification task, and SVM algorithm was applied, as implemented in the scikit-learn machine learning library [6].
- The preprocessing included removal of tokens like "RT", URLs, and tokens starting with "@" or "#".
- Besides the provided code-mixed training data for this task, the gold standard from the FIRE Microblog Track 2016 was also used.
- Undersampling was employed, i.e., only as many non-relevant tweets were given as input to the machine learning classifier as relevant tweets (since relevant tweets were much less as compared to irrelevant ones).
- For dealing with code-mixed tweets, Google Translate [2] was used to convert tweets in other languages to English. Specifically, if the language field of the tweet metadata was "hi" (which denotes Hindi) or "ne" (for Nepali), the tweet was translated from its original language to English. For tweets in any other non-English language, they were assumed to be in Nepali (the most common non-English language) and were translated to English.
- A threshold of 0.2 in the predicted score by the SVM classifier was set to classify a tweet as relevant.

Table 1: Results of our runs

Submission Detail		Availability-Tweets Evaluation			Need-Tweets Evaluation			Average MAP
S. No.	Run ID	Precision@100	Recall@100	MAP	Precision@100	Recall@100	MAP	MAP
1	iitbhu_fmt17_task1_2	0.79	0.5082	0.3786	0.79	0.7237	0.4986	0.4386
2	iitbhu_fmt17_task1_1	0.54	0.0867	0.057	0.58	0.2272	0.1241	0.0906

5 RESULTS

The results of our runs based on several metrics are given in Table 1.

Our run with Run ID iitbhu_fmt17_task1_2 was the best-performing run in this task.

REFERENCES

[1] [n. d.]. Apache Lucene Core. ([n. d.]). https://lucene.apache.org/core/.
[2] [n. d.]. Google Translate. ([n. d.]). https://translate.google.com/.
[3] [n. d.]. Lucene Scoring. ([n. d.]). https://lucene.apache.org/core/3_6_0/scoring.html.
[4] [n. d.]. Lucene Standard Analyzer. ([n. d.]). https://www.tutorialspoint.com/lucene/lucene_standardanalyzer.htm.
[5] M. Basu, S. Ghosh, K. Ghosh, and M. Choudhury. 2017. Overview of the FIRE 2017 track: Information Retrieval from Microblogs during Disasters (IRMiDis). In *Working notes of FIRE 2017 - Forum for Information Retrieval Evaluation (CEUR Workshop Proceedings)*. CEUR-WS.org.
[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.