Measuring Fact Check-ability In Microblogs*

Anurag Banerjee[1]

IIT (BHU), Varanasi
anu.bane.geu@gmail.com

Abstract. This paper presents the methodology employed by the author for IRMiDis Track in FIRE 2018. The task comprised of two subtasks, first, classifying whether a tweet is fact check-able or not and second, to identify the news article that would support or verify the tweet. Two runs were submitted for the first subtask only.

Keywords: Information Retrieval · Microblog · Disaster · Fact check-ability.

1 Introduction

Micro-blogging platforms, such as Twitter, have become in recent years, an important medium for information disbursement. The advantage of such platforms is that the source is omni-present, the people all around the world. It doesn't depend on any organisation to release the information. Due to this fact and its widespread availability, such platforms can be leveraged in times of emergency to gather information critical to relief/rescue operations.

The downside of this platform is that people also use it to spread rumours or false and misleading information. If rescue operatives are depending on a system that retrieves information from this platform, it becomes imperative to ensure that the information is genuine and free from bias.

To this end, this year's IRMiDis Track in FIRE 2018 [1] had set-up an experiment to detect the fact check-ability of tweets and to also find the relevant supporting news article to act as evidence.

1.1 Subtask 1 - Identifying fact check-able tweets

This subtask aimed at separating the given 50,000 tweets into two classes, if viewed as a classification problem or as an Information Retrieval problem that would generate a ranked list, with higher ranked tweets being more fact checkable. As an example:

- Fact Checkable tweet: #Nepal #Earthquake day four. Slowly in the capital valley Internet and electricity beeing restored . A relief for at least some ones

- Fact Uncheckable tweet: We humans need to come up with a strong solution to create earthquake proof zone's

* *Supported by IR-Lab, Dept. of Computer Science and Engineering, IIT (BHU), Varanasi*

1.2 Subtask 2 - Identification of supporting news article

This subtask aimed at identifying the news article(s) that would act as evidence for the tweet deemed as fact checkable. The dataset contained 6000 news articles for the same event.

2 Data

The dataset consisted of 50,000 tweets (microblog postings from Twitter) posted during the Nepal Earthquake of 2015 and 6,000 news articles published during the same time. A separate but small set of sample fact checkable tweets was also provided for training. The tweets were all in English. More training data was generated from within this dataset as part of the methodology applied.

3 Methodology

The task was treated as a PU (positive-unlabelled) classification problem (as described in [2] and [3]), since only a very small set of positive examples (around 80) was available for training. The steps followed are as under.

1. Given the JSONL file for 50,000 tweets and the small positive set,
   (a) New files were generated after removing punctuation, hashtags and URLs. Each line of the new file corresponds to a tweet as below:
   tweet id : cleaned tweet text

   (b) Using more tweet data from unrelated dataset and the gensim library for python, a doc2vec model was trained to represent each tweet as a 50 dimension vector

   (c) Manual intervention - manually observed all the provided tweets and randomly added few apparently fact checkable tweets from JSONL file to given small positive set

2. Calculated mean vector for given sample fact checkable tweet (SFCT) dataset - also calculated the max. euclidean distance any of these vectors had with the mean (max_euc) and the max. angle in radian any of these vectors had with the mean (max_angl).

3. Using thresholds max_euc and max_angl, did an ad-hoc classification of the JSONL dataset; out of 50,000 tweets - around 1400 tweets were declared as reliably negative (had distance and angle greater than the thresholds from the SFCT mean.)

4. Used crystallization (involved: original SFCT as seed and adding new tweets to the set in first iteration, then using new new SFCT as new seed in next iteration and so on) of the SFCT dataset using an angle threshold of 0.05166 over 7 iterations to grow the SFCT dataset - this will act as reliably positive (around 1500 tweets).

5. Using positive and negative examples generated at 4 and 3 above, an SVM Classifier was trained; accuracy of around 57.16% was obtained. (Accuracy of the classifier was measured on the original SFCT dataset.)

6. Using the above classifier, labels were predicted for the entire JSONL dataset. Two files were generated, one for fact checkable (*fc*) and another for fact uncheckable (*fuc*).

7. The *fc* file was put through the crystallization process for 5 iterations to grow its size, since a very small number of tweets were classified as positive by the classifier. The *fuc* file was simply populated as the complement of *fc* file w.r.t. the entire JSONL dataset.

8. For generating the final ranked result (positive tweets ranked higher than negative):
   (a) Both files were sorted separately on score and appended in final result as: first positive then negative
   (b) Score was the angle in radian; each tweet in *fc* and *fuc* files make some angles with all the tweets in the SFCT dataset; the least angle amongst these was taken as the score for the concerned tweet.

Contribution: In this semi-automatic method, the step that was introduced and apparently not generally used is the crystallization step at 4 and again used at 7. Since, the available set was extremely small for good training, this step was used to increase the size of training data.

4 Results

This submission was a semi-automatic approach as discussed above. The results for this methodology may be viewed in Table 1 for Run ID iitbhu_irlab_irmidis_ab_2. The 'NDCG Overall' was considered as the final measure of performance during evaluation.

Table 1. Results for Semi-Automatic approaches.

S. No.	Run ID	Run Type	Prec @100	Recall @1000	MAP @1000	MAP @100	NDCG @100	NDCG Overall
1	DAIICT-Hildesheim-mod3	SA	0.4000	0.2002	0.0129	0.1471	0.4021	0.7492
2	MIDAS_SEMI_AUTO-tweetid	SA	0.9600	0.1148	0.0740	0.1345	0.6007	0.6899
3	iitbhu_irlab_irmidis_ab_2	SA	0.3900	0.0447	0.0144	0.0401	0.3272	0.6200

References

1. Basu, M., Ghosh, S., Ghosh, K.: Overview of the FIRE 2018 track: Information Retrieval from Microblogs during Disasters (IRMiDis). In: Proceedings of FIRE 2018 - Forum for Information Retrieval Evaluation (December 2018)
2. Li, X.L., Liu, B.: Learning from positive and unlabeled examples with different data distributions. In: Machine Learning: ECML 2005. pp. 218–229. Springer Berlin Heidelberg (2005)
3. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Proceedings of the Third IEEE International Conference on Data Mining. pp. 179–. ICDM '03, IEEE Computer Society (2003)