Using Relevancer to Detect Relevant Tweets: The Nepal Earthquake Case

Ali Hürriyetoğlu
Centre for Language Studies
Radboud University
P.O. Box 9103, NL-6500 HD,
Nijmegen, the Netherlands
a.hurriyetoglu@let.ru.nl

Antal van den Bosch
Centre for Language Studies
Radboud University
P.O. Box 9103, NL-6500 HD,
Nijmegen, the Netherlands
a.vandenbosch@let.ru.nl

Nelleke Oostdijk
Centre for Language Studies
Radboud University
P.O. Box 9103, NL-6500 HD,
Nijmegen, the Netherlands
n.oostdijk@let.ru.nl

1. INTRODUCTION

In this working note we describe our submission to the FIRE 2016 Microblog track *Information Extraction from Microblogs Posted during Disasters* [1]. The task in this track was to extract all relevant tweets pertaining to seven given topics from a set of tweets. The tweet set was collected using key terms related to the Nepal Earthquake[1].

Our submission is based on a semi-automatic approach in which we used Relevancer, a complete analysis pipeline designed for analyzing a tweet collection. The main analysis steps supported by Relevancer are (1) preprocessing the tweets, (2) clustering them, (3) manually labeling the coherent clusters, and (4) creating a classifier that can be used for classifying tweets that are not placed in any coherent cluster, and for classifying new (i.e. previously unseen) tweets using the labels defined in step (3).

The data and the system are described in more detail in Sections 2 and 3, respectively.

2. DATA

At the time of download (August 3, 2016), 49,660 tweet IDs were available out of the 50,068 tweet IDs provided for this task. The missing tweets had been deleted by the people who originally posted them. We used only the English tweets, 48,679 tweets in all, based on the language tag provided by the Twitter API. Tweets in this data set were already deduplicated by the task organisation team as much as possible.

The final tweet collection contains tweets that were posted between April 25, 2015 and May 10, 2015. The daily distribution of the tweets is visualized in Figure 1.

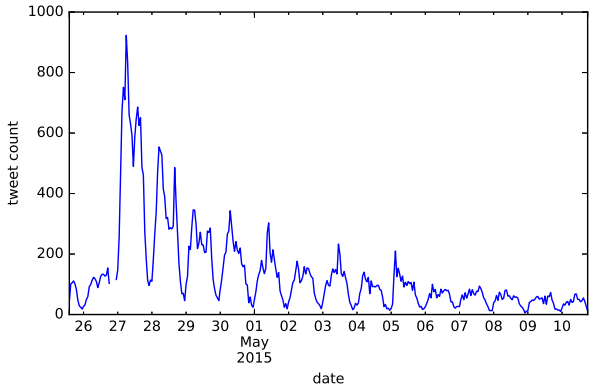

Figure 1: Temporal distribution of the tweets

3. SYSTEM OVERVIEW

The typical analysis steps of the Relevancer were applied to the data provided for this task. The current focus of the Relevancer tool is the text and the date of posting of a tweet. Relevancer aims at discovering and distinguishing between the different topically coherent information threads in a tweet collection[3, 2]. Tweets are clustered such that each cluster represents an information thread and the clusters can be used to train a classifier.

Each step of the analysis process is described in some detail in the following subsections[2].

3.1 Normalisation

Normalisation starts with converting user names and URLs that occur in the tweet text to the dummy values 'usrusrusr' and 'urlurlurl' respectively.

After inspection of the data, we decided to normalise a number of phenomena. First, we removed certain automatically generated parts at the beginning and at the end of a tweet text. We determined these manually, e.g. '*live updates:*', '*I posted 10 photos on Facebook in the album*' and '*via usrusrusr*'. After that, words that end in '...' were removed as well. These words are mostly incomplete due to the length restriction of a tweet text, and are usually at the end of tweets generated from within another application. Also, we eliminated any consecutive duplication of a token. Duplication of tokens mostly occurs with the dummy forms for user names and urls, and event-related key words and entities. For instance, two of three consecutive tokens at the beginning of the tweet *#nepal: nepal: nepal earthquake: main language groups (10 may 2015) urlurlurl #crisismanagement* were removed in this last step of normalization. This last step facilitates the process of identifying the actual content of the tweet text.

3.2 Clustering and labeling

The clustering step aims at finding topically coherent groups of tweets that we call information threads. These groups are

[1]https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake

[2]See http://relevancer.science.ru.nl and https://bitbucket.org/hurrial/relevancer for further details.

labeled as **relevant**, **irrelevant**, or **incoherent**. Coherent clusters were selected from the output of the clustering algorithm K-Means[3], with $k = 200$, i.e. a preset number of 200 clusters. Coherency of a cluster is calculated based on the distance between the tweets in a particular cluster and the cluster center. Tweets that are in incoherent clusters (as determined by the algorithm) were clustered again by relaxing the coherency restrictions until the algorithm reaches the requested number of coherent clusters. The second stop criterion for the algorithm is the limit of the coherency parameter relaxation.

The coherent clusters were extended with the tweets that are not in any coherent cluster. This step was performed by iterating all coherent clusters in descending order of the total length of the tweets in a cluster and adding tweets that have a cosine similarity higher than 0.85 with respect to the center of a cluster to that respective cluster. The total number of tweets that were transferred to the clusters this way was 847.

As Relevancer takes dates of posts as relevant information, the tool first searches for coherent clusters of tweets in each day separately. Then, in a second step it clusters all tweets from all days that previously were not placed in any coherent cluster. Applying the two steps sequentially enables Relevancer to detect local and global information threads as coherent clusters respectively.

For each cluster thus identified, a list of tweets is presented to an expert who then determines which are the relevant and irrelevant clusters[4]. Clusters that contain both relevant and irrelevant tweets are labeled as incoherent by the expert[5]. Relevant clusters are those which an expert considers to be relevant for the aim she wants to achieve. In the present context more specifically, clusters that are about a topic specified as relevant by the task organisation team should be labeled as relevant. Any other coherent cluster should be labeled as irrelevant.

3.3 Creating the classifier

The classifier was trained with the tweets labeled as relevant or irrelevant in the previous step. Tweets in the incoherent clusters were not used. The Naive Bayes method was used to train the classifier.

We used a small set of stop words. These are a small set of key words (nouns), viz. *nepal, earthquake, quake, kathmandu* and their hashtag versions[6], determiners *the, a, an*, conjunctions *and, or*, prepositions *to, of, from, with, in, on, for, at, by, about, under, above, after, before*, and the news related words *breaking* and *news* and their hashtag versions. The normalized forms of the user names and URLs *usrusrusr* and *urlurlurl* are included in the stop word list as well.

We optimized the smoothing prior parameter α to be 0.31 by cross-validation, comparing the classifier performance with equally separated 20 values of α between 0 and 2. Word unigrams and bigrams were used as features. The performance of the classifier on a 15% held-out data is provided below in Tables 1 and 2 [7].

	Irrelevant	Relevant
Irrelevant	**720**	34
Relevant	33	**257**

Table 1: Confusion matrix of the Naive Bayes classifier on test data. The rows and the columns represent the actual and the predicted labels of test tweets. The diagonal provides the correct number of predictions.

	precision	recall	F1	support
Irrelevant	.96	.95	.96	754
Relevant	.88	.89	.88	290
Avg/Total	.94	.94	.94	1,044

Table 2: Precision, recall, and F1-score of the classifier on the test collection. The recall is based on the test set.

The whole collection was classified with the trained Naive Bayes classifier. 11,300 tweets were predicted as relevant. We continued the analysis with these relevant tweets.

3.4 Clustering and labeling relevant tweets

Relevant tweets, as predicted by the automatic classifier, were clustered without filtering them based on the coherency criteria. In contrast to the first clustering step, the output of K-means was used as is, again with $k = 200$. These clusters were annotated using the seven topics as predetermined by the task. To the extent possible, incoherent clusters were labeled using the closest provided topic. Otherwise, the cluster was labeled as irrelevant.

The clusters that have a topic label contain 8,654 tweets. Since the remaining clusters, containing 2,646 tweets, were evaluated as irrelevant, they were not included in the submitted set.

4. RESULTS

The result of our submission was recorded under the ID relevancer_ru_nl. The performance of our results was evaluated by the organisation committee at ranks 20, 1,000, and all, considering the tweets retrieved in the respective ranks. As announced by the organisation committee, our results are as follows: 0.3143 precision at rank 20, 0.1329 and 0.0319 recall and Mean Average Precision (MAP) at rank 1,000 respectively, and 0.0406 MAP considering all tweets in our submitted results.

We generated an additional calculation for our results based on the annotated tweets provided by task organizers. The overall precision and recall are 0.081 and 0.34 respectively. The performance for the topics FMT1 (available resources), FMT2 (required resources), FMT3 (available medical resources), FMT4 (required medical resources), FMT5

[3]We used scikit-learn v0.17.1 for all machine learning tasks in this study http://scikit-learn.org.

[4]The first author of this working note had the role of being the expert for this task. A real scenario would require a domain expert.

[5]Although the algorithmic approach determines the clusters that were returned as coherent, the expert may not agree with it.

[6]This set was based on our observation as we did not have access to the key words that were used to collect this data set.

[7]Since we optimize the classifier for this collection, the performance of the classifier on unseen data is not relevant here.

(resource availability at certain locations), FMT6 (NGO and governmental organization activities), and FMT7 (infrastructure damage and restoration reports) is provided in the Table 3.

	precision	recall	F1	percentage
FMT1	0.17	0.50	0.26	0.27
FMT2	0.35	0.09	0.15	0.14
FMT3	0.19	0.28	0.23	0.16
FMT4	0.06	0.06	0.06	0.05
FMT5	0.05	0.06	0.06	0.09
FMT6	0.05	0.74	0.09	0.18
FMT7	0.25	0.08	0.12	0.12

Table 3: Precision, recall, and F1-score of our submission and the percentage of the tweets in the annotated tweets per topic.

On the basis of these results, we conclude that the success of our method differs drastically across topics. In Table 3, we observe that there is a clear relation between the F1-score and the percentage of the tweets per topic in the manually annotated data. Consequently, we conclude that our method performs better in case the topic is presented well in the collection.

5. CONCLUSION

In this study we applied the methodology supported by the Relevancer system in order to identify relevant information by enabling human input in terms of cluster labels. This method has yielded an average performance in comparison to other participating systems.

We observed that clustering tweets for each day separately enabled the unsupervised clustering algorithm to identify specific coherent clusters in a shorter time than the time spent on clustering the whole set. Moreover, this setting provided an overview that realistically changes each day, for each day following the day of the earthquake.

Our approach is optimized to incorporate human input. In principle, an expert should be able to refine a tweet collection until she reaches a point where the time spent on a task is optimal and the performance is sufficient. However, with this particular task, an annotation manual was not available and the expert had to stop after one iteration without being sure to what extent certain information threads were actually relevant to the task at hand; for example, are (clusters of) tweets pertaining to providing or collecting funds for the disaster victims considered to be relevant or not.

It is important to note that the Relevancer system yields the results in random order, as it has no ranking mechanism that ranks posts for relative importance. We speculate that rank-based performance metrics are not optimally suited for evaluating it.

In our future work we will aim to increase the precision and diminish the performance differences across topics, possibly by downsampling or upsampling methods to tackle class imbalance.

6. ACKNOWLEDGEMENTS

This research was funded by the Dutch national research programme COMMIT.

7. REFERENCES

[1] S. Ghosh and K. Ghosh. Overview of the FIRE 2016 Microblog track: Information Extraction from Microblogs Posted during Disasters. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[2] A. Hürriyetoğlu, C. Gudehus, N. Oostdijk, and A. van den Bosch. Relevancer: Finding and labeling relevant information in tweet collections. In E. Spiro and Y.-Y. Ahn, editors, *Social Informatics*, volume 10046. Springer International Publishing, November 2016.

[3] A. Hürriyetoğlu, A. van den Bosch, and N. Oostdijk. Analysing role of key term inflections in knowledge discovery on twitter. In *International Workshop on Knowledge Discovery on the Web*, Cagliari, Italy, September 2016.