Class Specific TF-IDF Boosting for Short-text Classification

Application to Short-texts Generated During Disasters

Samujjwal Ghosh
IIT Hyderabad
Hyderabad, Telangana, IN
cs16resch01001@iith.ac.in

Maunendra Sankar Desarkar
IIT Hyderabad
Hyderabad, Telangana, IN
maunendra@iith.ac.in

ABSTRACT

Proper formulation of features plays an important role in short-text classification tasks as the amount of text available is very little. In literature, Term Frequency - Inverse Document Frequency (TF-IDF) is commonly used to create feature vectors for such tasks. However, TF-IDF formulation does not utilize the class information available in supervised learning. For classification problems, if it is possible to identify terms that can strongly distinguish among classes, then more weight can be given to those terms during feature construction phase. This may result in improved classifier performance with the incorporation of extra class label related information. We propose a supervised feature construction method to classify tweets, based on the actionable information that might be present, posted during different disaster scenarios. Improved classifier performance for such classification tasks can be helpful in the rescue and relief operations. We used three benchmark datasets containing tweets posted during Nepal and Italy earthquakes in 2015 and 2016 respectively. Experimental results show that the proposed method obtains better classification performance on these benchmark datasets.

KEYWORDS

Information Retrieval, Short-text Classification, Feature Engineering, Entropy-based Feature Generation

ACM Reference Format:
Samujjwal Ghosh and Maunendra Sankar Desarkar. 2018. Class Specific TF-IDF Boosting for Short-text Classification: Application to Short-texts Generated During Disasters. In *The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3184558.3191621

1 INTRODUCTION

Short texts like tweet contain very limited contextual information due to their length restriction. Because of this, classifying short texts using machine learning techniques is a challenging task. Proper formulation of feature vector plays an important role in such a scenario. In disaster mitigation related literature, where textual feature is used, most common technique used for feature representation is Term Frequency - Inverse Document Frequency (TF-IDF). In TF-IDF feature representation, each document or short-text is represented as a vector where the fields correspond to the terms in the vocabulary. The value stored in a field is the TF-IDF score of the corresponding term. The TF-IDF score is the product of the Term Frequency (TF) of the term in that document and the Inverse Document Frequency (IDF) of that term in the corpus. Mathematically, TF-IDF can be denoted by,

$$TF.IDF = tf_d^i \times \log \frac{N}{df^i}$$

where tf_d^i is the number of times term i occurred in document d, N is the total number of documents in the corpus and df^i is the number of documents in which term i occurred. TF captures the importance of the term in the document and is computed as an increasing function of the term's frequency. On the other hand, IDF tries to measure how informative a term is in the corpus. The assumption commonly made here is, if a term is frequent in a corpus then it is not much informative, whereas terms that are rare are more informative and hence important. IDF is modeled as a decreasing function of the term's document frequency. This feature construction strategy using TF-IDF is often used to classify text documents - both short [24] and long [25].

For supervised classification problems, labeled training data is assumed to be available. The TF-IDF based approaches do not consider these class labels during feature construction. However, from the labeled data it might be possible to identify terms that are discriminative and hence strong indicators for certain classes. We want to add the term's importance (distinguishing power) among different classes as an extra information in the feature construction process. A term is considered discriminative if it occurs sufficiently large number of times in a particular class but rarely occurs in other classes. Let t_1 and t_2 be two terms that occurred k times in a corpus. However, t_1 appeared uniformly in documents across all classes, but t_2 occurred in class c_i sufficiently more times than it occurred in other classes combined (i.e. $\forall c_j \in C; c_j \neq c_i$). IDF score for both the terms t_1 and t_2 will be same. However, it is evident that the term t_2 has more discriminating power as its presence in a future document is strongly indicative of the document's belongingness to that particular class (i.e. in our example class c_i).

We wish to capture this distinguishing power of different terms and use this information during feature construction phase. In this work, we propose techniques of boosting TF-IDF scores to better represent the term distribution among classes. Classifiers can then

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution. In case of republication, reuse, etc., the following attribution should be used: "Published in WWW2018 Proceedings © 2018 International World Wide Web Conference Committee, published under Creative Commons CC BY 4.0 License."

WWW'18 Compaion, April 23–27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5640-4/18/04..
https://doi.org/10.1145/3184558.3191621

exploit this extra information to make better decisions. These general techniques can be applied in different applications where bag-of-word based TF-IDF features are used. We applied our proposed approach to classify disaster related tweets to understand its impact and usefulness over using traditional TF-IDF.

During disasters, people have been found to post lots of messages in the micro-blogging sites such as Twitter, Weibo etc. Some of these posts may actually contain information regarding damages to the infrastructure, requirement of resources such as water, medicines, etc. Proper annotation of such posts with kind of information they contain can help in the rescue and relief operations, thereby mitigating the miseries of people affected by the disaster.

There have been many studies in the literature, regarding proper utilization of short-texts generated during disaster, to effectively plan rescue and relief operations [8], [10], [11], [13], [18], [20], [23]. Here we present a consolidated summary of different work related to disaster related tweet classification. Paper [8] presented a comparative study of disaster related tweet classification using various algorithms with TF-IDF features. Authors in [10] proposed a system which not only filters and categorizes English tweets, but also worked on multilingual tweets related to typhoon Lawin (international name: Haima) and Karen (international name: Sarika). This system was built by using TF-IDF features with Support Vector Machine (SVM) classifier. Although, [11] mainly concentrates on neural network based approaches to retrieve disaster related microblogs, they used TF-IDF Rocchio scores to expand their query before using them on neural networks. Domain adaptation approach, which learns classifiers from unlabelled target data, is taken in [13] in which authors utilized information available from a past disaster to filter tweets related to a new disaster. Domain adaptation was achieved by using self-training technique on modified version of weighted Naive Bayes classifier with TF-IDF features. [18] built a system which automatically detects any disaster happening by monitoring the Twitter stream. Authors used Naive Bayes and SVM as their classifier with TF-IDF based feature vectors. In [20], authors proposed an automated text classification system which filters only disaster related short-texts. The proposed method works by selecting prominent TF-IDF features using Chi-square technique. A study between matching based and learning based approaches to filter relevant tweets generated during disaster were done by the authors of [23]. They employed various techniques like geo-tag information, word2vec [17] embedding along with TF-IDF scores. As we can see from these discussions use of TF-IDF is very common in the area of disaster related tweet classification.

In this work we focus on the task of classifying informative tweets posted during disasters. The example classes we consider are resource availability and requirement related, infrastructure damage related, etc. Even minor improvement in the classification performance can help the rescue organizations to look at specific messages and accordingly make decisions to channel the relief operations in appropriate manner. We used three disaster related tweet datasets to test the effectiveness of our proposed feature construction techniques. Our methods significantly outperformed the TF-IDF based classification method on these benchmark datasets.

The rest of the paper is organized as follows. In Section 2 we discuss related works in the field of TF-IDF score modification for classification task. We define the problem in Section 3. Section 4 discusses about proposed approach in detail. Discussion about our experimental setup along with dataset details are given in Section 5. Finally, we will present our experimental results in Section 6.

2 RELATED WORK

In this Section, we look at different work from the literature that deals with variants of TF-IDF modification for classification tasks. However, most techniques are based on feature-selection approach, rather than TF-IDF score modification, in which a subset of features are selected based on terms' discriminative power. This subset selection can be done using various methods like, Information Gain (IG) [2], Chi-square [19], Mutual Information [26], etc. However, these methods do not take advantage of term's frequency among classes. However, people have experimented with different TF-IDF modification techniques. Below we discuss few such approaches present in literature.

Authors in [3] incorporated bi-grams along with traditional unigrams based features to incorporate extra information. Although this approach does not alter the values of TF-IDF, they increase the number of unique features in the vocabulary. They showed that increasing the vocabulary size might increase classifier performance. Bi-Normal Separation (BNS) was used instead of IDF when generating features by the authors of [5]. BNS ranks terms based on their distinguishing power. Authors found scaling terms' importance by BNS without any feature selection improved their classifier accuracy. In [12] an entropy based approach was proposed called Entropy-based Category Coverage Difference (ECCD) in which they calculated the entropy of each term across classes to get the importance of terms for different class concentrations. To tackle class imbalance problem, [14] proposed a probability based term weighting scheme which improved classifier performance for classes in which number of data points are less compared to other classes. In another approach, semantically modified TF-IDF scores were used to categorize biomedical data by [15]. Better performance of SVM classifier using modified feature set was found. Delta TF-IDF was proposed by [16] which modifies the TF-IDF score to better understand sentiments of blogs. The Delta part was calculated by taking the difference in the TF-IDF score of positive and negative sentiments of training data. In [21], authors used inverse-class-frequency (ICF) similar to IDF which denotes how important a term is. ICF gives highest score to those terms which occur in few classes and lowest score to terms which occur in many classes. They showed using ICF instead of IDF gives better classifier performance. [22] shows the effect of using IG to select most prominent features instead of using all the features. They found that use of IG improved their classification accuracy. The work by [24] proposed low granular features for short text classification task focusing mainly on Chinese texts. Authors in [25] also proposed two entropy based approaches called tf.dc and tf.bdc which measure the Distributional Concentration (DC) among classes. In DC approach, entropy was calculated over classes rather than documents. In second approach Balanced Distributional Concentration (BDC) was proposed which takes class size into account to calculate DC. However, most of these approaches are tuned for long-text and does not optimize for short texts where context information is limited.

3 PROBLEM DEFINITION

Our main goal in this work is to classify short-texts, given a set of short-texts and their classes. The problem can be formulated as:

> Let, $T = \{\boldsymbol{t}^1, \boldsymbol{t}^2, \cdots, \boldsymbol{t}^N\}$ be a set of N textual data points and $C = \{1, 2, \cdots, m\}$ be a set of m classes. Given a set of mappings of the form $\{\boldsymbol{t}^i, c_1^i, \cdots, c_k^i\}$ where data $\boldsymbol{t}^i \in T$ and classes $c_1^i, \cdots, c_k^i \in C$, our goal is to find all applicable classes for a new data $\boldsymbol{t}^{new}$.

Below we discuss our proposed methods of utilizing term-class relationship when constructing TF-IDF features.

4 PROPOSED METHOD

We want to classify unseen short texts, given a set of short-texts and their class labels as training data. Keeping this objective in mind, we first try to identify ways of measuring the term's relationship with different classes. Then we see how this information can be leveraged to assign new tweets to appropriate classes. For terms that are inherently specific to certain classes, we expect their distributions to be concentrated in those classes. On the other hand, terms that are generic may be roughly uniformly distributed over all the classes. One common way to identify presence or absence of such concentrations is through Entropy. We compute the entropy of a term t_i as

$$H(t^i) = -\sum_c p_c^i \times \log_2(p_c^i)$$

where p_c^i is the probability that if a term t_i is present in a document, then the document comes from class c. We estimate p_c^i as the ratio of number of times t_i is present in class c and the number of times it is present across all classes. Then, the formula for computing entropy of a term t_i can be written as:

$$H(t^i) = -\sum_{c=1}^{m} \frac{tc_c^i}{tc^i} \times \log_2\left(\frac{tc_c^i}{tc^i}\right). \tag{1}$$

where tc_c^i denotes the count of term t^i in class c and tc^i denotes the count across all classes, i.e. $tc^i = \sum_{k=1}^{m} tc_k^i$.

4.1 Normalized Entropy Boosting

Once we calculated the entropy of each term in the corpus, we want to get an estimate of how informative (concentrated) a term is for each class. We proposed an entropy based approach called Normalized Entropy Boosting. We calculated Normalized Entropy (NE) for term t^i by,

$$NE(t^i) = \frac{H_{max} - H(t^i)}{H_{max}} \tag{2}$$

where $H_{max} = \max_{t^i} H(t^i)$ and H_{max} denotes the maximum value of all the entropies. We modified the TF-IDF values by following equation,

$$TF.IDF_{NE}(t^i) = TF.IDF(t^i) \times NE(t^i) \tag{3}$$

Terms which are concentrated in few classes should have higher NE whereas terms that are almost uniformly distributed among classes should have lower NE. Although TF.IDF$_{NE}$ gives better precision than traditional TF-IDF, but Recall is very low as observed in Table 4. We propose another approach in Section 4.2 to improve performance over TF.IDF$_{NE}$.

4.2 Class Normalized Entropy Boosting

Here we propose our second approach which handles the low Recall problem mentioned in Section 4.1. This approach retains the actual TF-IDF score and boosts the class-specific term's importance as a side information. In this way, we would be able to retain significance of TF-IDF based scores and also be able to give extra boost to known important terms. The entropy measure described in above Section 4.1 suffers from class size imbalance, mainly for smaller classes. In this approach, we also factor in the class sizes when computing the Importance Weight (IW) of a term. We can calculate the IW of term t^i for class c as follows:

$$IW(t_c^i) = \frac{tc_c^i}{k_c}$$

where k_c denotes the number of terms present in class c. Now, we calculate the Class Normalized Entropy (CNE) by considering both entropy and importance according to the following equation.

$$TF.IDF_{CNE}(t_c^i) = \begin{cases} TF.IDF(t^i) + \dfrac{NE(t^i) \times IW(t_c^i)}{k}, & \text{if } TF.IDF(t^i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The denominator k works as a normalizing hyper-parameter. Effects of the additional boosting can be controlled by changing the value of k. More detailed effects of k is discussed in Section 5.3.

5 EXPERIMENTAL DETAILS

In our experiments, we used Support Vector Machine (SVM) [4] with linear kernel as our classifier. Many studies [8], [10], [18], [20] found SVM works best with TF-IDF feature vectors in disaster scenario. SVM with three different TF-IDF boosting approaches were used for experiments. $TF.IDF$ denotes simple TF-IDF values, TF.IDF$_{NE}$ denotes TF-IDF values with Normalized Entropy. Our other approach in Section 4.2 is denoted as TF.IDF$_{CNE}$. We also implemented the approach discussed in [12] called ECCD and denoted as TF.IDF$_{ECCD}$ in Table 4[1].

5.1 Datasets

Three disaster related tweets datasets were used for experimentation. "Forum for Information Retrieval Evaluation" 2016 (FIRE16) [6] and 2017 (FIRE17) [1] datasets which contain tweets posted during Nepal 2015 earthquake were selected. Class details of FIRE16 and FIRE17 are mentioned in Tables 1a and 1b respectively. We also tested with "Social Media for Emergency Relief and Preparedness" 2017 (SMERP17) [7] dataset containing tweets posted during August 2016 earthquake in Italy. Details of the SMERP17 dataset are given in Table 1c[2]. We also created a custom dataset by merging SMERP17 and FIRE16 datasets denoted as FIRE16+SMERP17 1d. As the number of classes varies between them, we mapped FIRE16 to 4 classes similar to SMERP17. The mapping between the two datasets is given in Table 3. We could not find any suitable mapping for class 5 of dataset FIRE16, so we removed all the tweets which occurs only in class 5. All of the above mentioned datasets were divided into train and test sets with 70% and 30% of the total available labeled data respectively. The class details were given

[1] The source code to reproduce the study can be found on **github**

[2] It is to be noted that data count may vary from the original dataset mentioned. Twitter does not allow direct tweet sharing, tweets were downloaded before the experiments and some tweets may not be retrieved if it is deleted or made private.

in TREC format. Description of each class contains four fields: the class ID, title (small title to denote the class), desc (short description of the class) and narr (detailed narrative of which text should be considered for this class). Example of class description in TREC format is given below for class 7 of FIRE16 dataset and class 1 of SMERP17 dataset.

```
<num> Number: FMT7
<title> WHAT INFRASTRUCTURE DAMAGE AND
RESTORATION WERE BEING REPORTED
<desc> Description: Identify the messages which
contain information related to infrastructure
damage or restoration.
<narr> A relevant message must mention the damage
or restoration of some specific infrastructure
resources, such as structures (e.g.,
dams, houses, mobile tower), communication
infrastructure (e.g., roads, runways, railway),
electricity, mobile or Internet connectivity,
etc. Generalized statements without reference
to infrastructure resources would not be
relevant.
```

```
<num> Number: SMERP-T1
<title> WHAT RESOURCES ARE AVAILABLE
<desc> Identify messages which describe the
availability of some resources.
<narr> 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 wi-fi, sms,
calling facility etc. will also be relevant.
In addition, any message or announcement about
donation of money will also be relevant. However,
generalized statements without reference to any
resource would not be relevant.
```

5.2 Preprocessing

Before working with the data we preprocessed and cleaned it by performing the below-mentioned steps in sequence.

(1) **Acronym Expansion:** Tweets are generally written with various acronyms. We used a modified version of the dictionary given in [9] by adding some extra terms ourselves. All the abbreviated words were replaced by the phrase/words given in the dictionary.

(2) **Removal of Emoticons and non-ASCII Characters:** Another prevalent problem with tweets is emoticons. We search for and removed all emoticon and non-ASCII characters by pattern matching.

(3) **Case Folding:** All tweet texts were converted to lower case after all the above mentioned processing was done.

(4) **Stop-words and Punctuation Removal:** After all the above mentioned steps were done we removed any word from the tweet which is present in the nltk stopwords[3].

(5) **Special Character Removal:** We removed characters like '#', '@' without removing the corresponding hashtags or user mentions. Also, we removed some other special words like "rt", "via" and "amp" which are not stop-words but contains no value whatsoever.

(6) **URLs and Phone numbers handling:** URLs' and phone numbers present in any tweet was replaced by keywords "urlurl" and "phonenumber" respectively.

Table 1: Class number, title and training and test data counts

(a) Class specific details of dataset FIRE16

Class	Title	Train	Test
1	Resources Available	401	175
2	Resources Required	210	81
3	Medical Resources Available	231	100
4	Medical Resources Required	75	36
5	Resources Specific Locations	135	53
6	Activities NGOs / Government	252	119
7	Infrastructure Damage Restoration	178	74
	Average tweets per class	211	91

(b) Class specific details of dataset FIRE17

Class	Title	Train	Test
1	Need related	461	207
2	Availability related	148	55
	Average tweets per class	304	131

(c) Class specific details of dataset SMERP17

Class	Title	Train	Test
1	Resources Available	228	82
2	Resources Required	152	62
3	Infrastructure Damage, Restoration, Casualties	1405	611
4	Rescue Activities NGOs / Government	255	105
	Average tweets per class	510	215

(d) Class specific details of dataset FIRE16 + SMERP17

Class	Title	Train	Test
1	Resources Available	733	367
2	Resources Required	402	156
3	Infrastructure Damage Restoration	1610	658
4	Activities NGOs / Government	494	237
	Average tweets per class	809	354

[3] https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/stopwords.zip

Below we show a tweet in original form and after preprocessing was done:

Table 2: Tweet before and after preprocessing

Before:	Doctors Italian Relief Corps of the Order of Malta are providing help in areas hit by violent earthquake in Italy https://t.co/DDszXXhKgn
After:	doctors italian relief corps order malta providing help areas hit violent earthquake italy urlurl

5.3 Parameter Tuning and Cross Validation

We have two different types of hyper-parameter. First parameter is the value of k in Equation (4) which decides the contribution of class specific boosting of our approach TF.IDF$_{CNE}$ and the regularization parameter for SVM classifier. We first tuned the boosting parameter k without tuning the regularization parameter[4]. We considered $[1, 2, 3, \cdots 10]$ as values of k to tune the boosting parameter and found $k = 2$ gives best result [5]. We fixed the value of k for all subsequent operations.

We used 5-fold cross validation on the training set to tune the regularization parameter of SVM. It was tuned with values [10^{-1}, 10^{0}, 10^{1}, $\cdots$, 10^{4}]. Regularization parameter in SVM indicates the amount of importance to be given to wrong classifications. Higher value signifies higher cost for miss classifications. However, there is a trade-off of incrementing, as it shrinks the margin between classes. As a result we will get a classifier with a small margin. It should be noted that in case of TF.IDF$_{CNE}$ regularization parameter was tuned after our boosting parameter tuning was done.

6 RESULTS

In this Section, we discuss our findings related to the effects of TF-IDF boosting for disaster related tweets.

Table 3: Class mappings between FIRE16 and SMERP17

FIRE16 Class	Class	SMERP17 Class	Class
Resources Available	1	Resources Available	1
Medical Resources Available	3		
Resources Required	2	Resources Required	2
Medical Resources Required	4		
Resources Specific Locations	5		-
Activities NGOs / Government	6	Rescue Activities NGOs / Government	4
Infrastructure damage restoration	7	Infrastructure Damage, Restoration, Casualties	3

6.1 Results for the complete collection - all classes considered together

Table 4 lists the F$_1$, Precision and Recall of our experiments on 4 datasets mentioned in Section 5.1. We can observe from Table 4 that incorporating class specific information in the TF-IDF formulation has significantly increased classifier accuracy over traditional TF-IDF in all datasets.

In the NE approach (TF.IDF$_{NE}$), we can clearly see that the Precision has increased over traditional TF.IDF but Recall went down bringing down the F$_1$ score. Although TF.IDF$_{NE}$ gives better precision than traditional TF-IDF, it fails to generalize where new data points do not contain any important terms from the existing training set vocabulary. As a result, NE value for those new terms will be very low and TF.IDF$_{NE}$ gives very low score to that data point. This happens because of the multiplicative nature of Equation (3). This is in-fact one of the limitations of TF.IDF$_{NE}$ boosting approach. As we are multiplying the boosted value with TF-IDF, if the boosting value is low it will bring down the overall score for TF.IDF$_{NE}$. We found this happens significantly more in smaller classes (small number of data points in the training set) because the vocabulary size for that class will be very limited. However, this technique might be useful in scenarios where Precision has higher priority than Recall.

Our second approach called TF.IDF$_{CNE}$ generalizes better than TF.IDF$_{NE}$ as seen in Table 4. This technique is able to handle unseen terms better and works well for smaller datasets where some of the actual important terms may not have sufficient statistics in the observed data as it incorporates the vocabulary size of each class. Our approach gives better result than TF.IDF$_{ECCD}$. However, we still see Recall is low in the first two datasets. This is happening because of the small number of data points. One of the obvious remedy of this problem is to have more data. FIRE16 and FIRE17 have 211 and 304 data point per class on average respectively as mentioned in Table 1a and 1b. More data will most likely include all possible important terms to the vocabulary. This behavior can be observed in case of SMERP17 and the FIRE16 + SMERP17 dataset as they on average have 510 (Table 1c) and 809 (Table 1d) data points per class respectively.

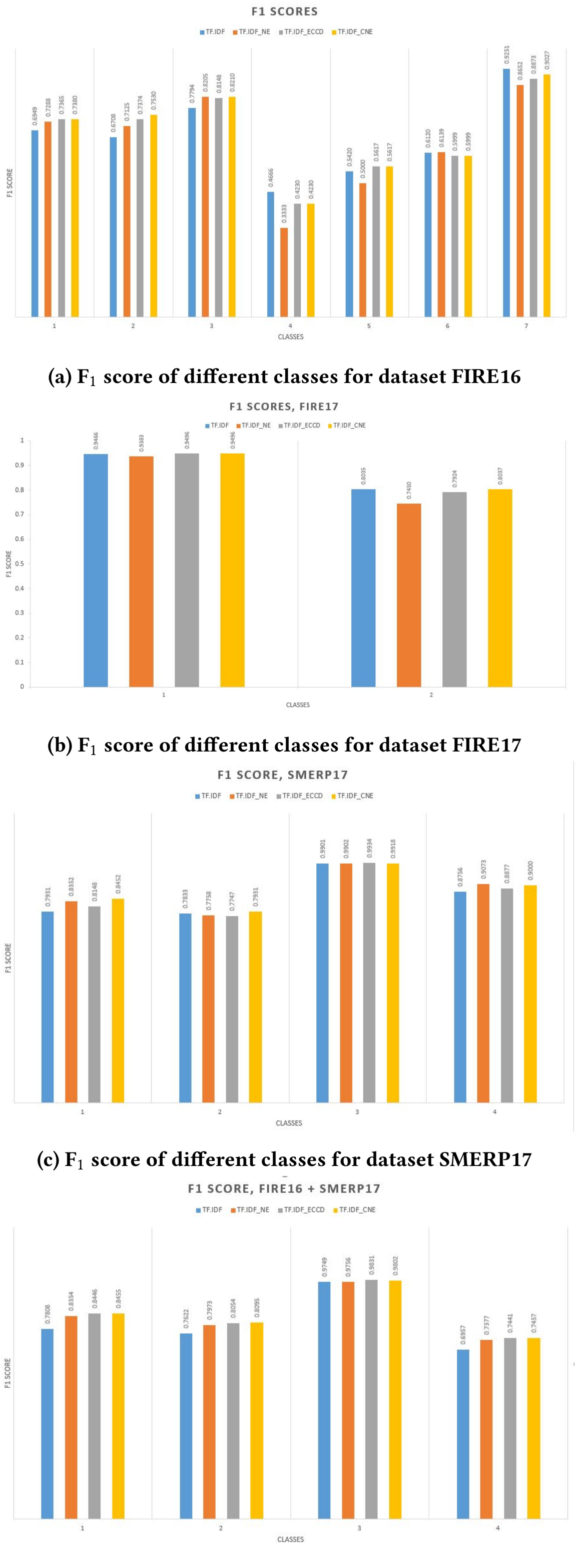
6.2 Results for individual classes

In this Section we look more deeply into individual class label performances of our proposed approaches. Consolidated performances of all the approaches mentioned in Section 4.2 is provided in Table 5, 6, 7 and 8. We see better performance of our proposed approaches than traditional TF-IDF when available training data was large. Our approach TF.IDF$_{CNE}$ performed poorly than traditional TF.IDF for only for class 4 of FIRE16 dataset as observed in Table 5d. It should also be noted that class 4 of FIRE16 has only 75 training data among all the classes across all datasets as seen on Table 1. Figure 1a, 1b, 1c, 1d display the F$_1$ scores of FIRE16, FIRE17, SMERP17 and FIRE16 + SMERP17 datasets respectively.

7 CONCLUSION

In this paper we studied the usefulness of class specific TF-IDF score boosting. It is evident that incorporating class details by the

[4] Regularization parameter was set to 1 during boosting parameter k was tuned.
[5] Results in Table 4 for TF.IDF$_{CNE}$ is after regularization parameter was tuned.



(a) F$_1$ score of different classes for dataset FIRE16

(b) F$_1$ score of different classes for dataset FIRE17

(c) F$_1$ score of different classes for dataset SMERP17

(d) F$_1$ score of different classes for dataset FIRE16 + SMERP17

Figure 1: F$_1$ scores of different classes

Table 4: Consolidated results of TF-IDF boosting approaches

Dataset	Feature	F$_1$ score	Precision	Recall
FIRE16	TF.IDF	0.6701	0.6526	**0.6936**
	TF.IDF$_{NE}$	0.6535	0.7396	0.6081
	TF.IDF$_{ECCD}$	0.6801	0.7633	0.6284
	TF.IDF$_{CNE}$	**0.6856**	**0.7643**	0.6380
FIRE17	TF.IDF	0.8751	0.8703	**0.8801**
	TF.IDF$_{NE}$	0.8417	0.8647	0.8237
	TF.IDF$_{ECCD}$	0.8710	0.8831	0.8600
	TF.IDF$_{CNE}$	**0.8767**	**0.8849**	0.8692
SMERP17	TF.IDF	0.8605	0.8629	0.8604
	TF.IDF$_{NE}$	0.8771	0.8851	**0.8711**
	TF.IDF$_{ECCD}$	0.8677	**0.9082**	0.8333
	TF.IDF$_{CNE}$	**0.8825**	0.8994	0.8680
FIRE16 + SMERP17	TF.IDF	0.8034	0.8276	0.7831
	TF.IDF$_{NE}$	0.8365	0.8386	**0.8350**
	TF.IDF$_{ECCD}$	0.8443	**0.8686**	0.8223
	TF.IDF$_{CNE}$	**0.8452**	0.8663	0.8260

means of entropy and term frequencies can improve classifier accuracy over a purely TF-IDF scoring scheme. We showed our approach to work on 4 different multi-label disaster related datasets of short-texts. However, we also found that our approach works better if the classes are sufficiently large. In our future work we want to handle boosting such a way so that it can handle imbalanced class sizes. Another improvement can be explored if extra dimensional info can be incorporated for better perform.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work is supported by the uder Grant No.:EE/2016-17/034/MLA/MZAK/0235.

Table 5: Class specific results for FIRE16

(a) Class specific results for FIRE16 dataset with TF.IDF

Class	F_1-score	Precision	Recall
1	0.6949	0.7028	0.6871
2	0.6708	0.6543	0.6883
3	0.7794	0.7600	0.8000
4	0.4666	0.3888	0.5833
5	0.5420	0.5471	0.5370
6	0.6120	0.5966	0.6283
7	0.9251	0.9189	0.9315

(b) Class specific results for FIRE16 dataset with TF.IDF$_{NE}$

Class	F_1-score	Precision	Recall
1	0.7288	0.7206	0.7371
2	0.7125	0.7215	0.7037
3	0.8205	0.8421	0.8000
4	0.3333	0.6666	0.2222
5	0.5000	0.6285	0.4150
6	0.6139	0.6875	0.5546
7	0.8652	0.9104	0.8243

(c) Class specific results for FIRE16 dataset with TF.IDF$_{ECCD}$

Class	F_1-score	Precision	Recall
1	0.7365	0.7303	0.7428
2	0.7374	0.7468	0.7283
3	0.8242	0.8651	0.7700
4	0.4230	0.6875	0.3055
5	0.5617	0.6944	0.4716
6	0.5999	0.6923	0.5294
7	0.8873	0.9264	0.8513

(d) Class specific results for FIRE16 dataset with TF.IDF$_{CNE}$

Class	F_1-score	Precision	Recall
1	0.7380	0.7277	0.7485
2	0.7530	0.7530	0.7530
3	0.8210	0.8666	0.7800
4	0.4230	0.6875	0.3055
5	0.5617	0.6944	0.4716
6	0.5999	0.6923	0.5294
7	0.9027	0.9285	0.8783

Table 6: Class specific results for FIRE17

(a) Class specific results for FIRE17 dataset with TF.IDF

Class	F_1-score	Precision	Recall
1	0.9466	0.9512	0.9420
2	0.8035	0.7894	0.8181

(b) Class specific results for FIRE17 dataset with TF.IDF$_{NE}$

Class	F_1-score	Precision	Recall
1	0.9383	0.9209	0.9565
2	0.7450	0.8085	0.6909

(c) Class specific results for FIRE17 dataset with TF.IDF$_{ECCD}$

Class	F_1-score	Precision	Recall
1	0.9496	0.9428	0.9565
2	0.7924	0.8235	0.7636

(d) Class specific results for FIRE17 dataset with TF.IDF$_{CNE}$

Class	F_1-score	Precision	Recall
1	0.9496	0.9428	0.9428
2	0.8037	0.8269	0.7818

Table 7: Class specific results for SMERP17

(a) Class specific results for SMERP17 dataset with TF.IDF

Class	F$_1$-score	Precision	Recall
1	0.7931	0.7500	0.8414
2	0.7833	0.7966	0.7704
3	0.9901	0.9885	0.9918
4	0.8756	0.9166	0.8380

(b) Class specific results for SMERP17 dataset with TF.IDF$_{NE}$

Class	F$_1$-score	Precision	Recall
1	0.8352	0.8068	0.8658
2	0.7758	0.8181	0.7377
3	0.9902	0.9854	0.9950
4	0.9073	0.9300	0.8857

(c) Class specific results for SMERP17 dataset with TF.IDF$_{ECCD}$

Class	F$_1$-score	Precision	Recall
1	0.8148	0.8250	0.8048
2	0.7747	0.8600	0.7049
3	0.9934	0.9918	0.9950
4	0.8877	0.9570	0.8285

(d) Class specific results for SMERP17 dataset with TF.IDF$_{CNE}$

Class	F$_1$-score	Precision	Recall
1	0.8352	0.8255	0.8658
2	0.7931	0.8363	0.7540
3	0.9918	0.9886	0.9950
4	0.9000	0.9473	0.8571

Table 8: Class specific results for FIRE16 + SMERP17

(a) Class specific results for FIRE16 + SMERP17 dataset with TF.IDF

Class	F$_1$-score	Precision	Recall
1	0.7808	0.8118	0.7520
2	0.7622	0.8385	0.6987
3	0.9749	0.9771	0.9726
4	0.6957	0.6829	0.7089

(b) Class specific results for FIRE16 + SMERP17 dataset with TF.IDF$_{NE}$

Class	F$_1$-score	Precision	Recall
1	0.8354	0.8483	0.8228
2	0.7973	0.8133	0.7820
3	0.9756	0.9756	0.9756
4	0.7377	0.7171	0.7594

(c) Class specific results for FIRE16 + SMERP17 dataset with TF.IDF$_{ECCD}$

Class	F$_1$-score	Precision	Recall
1	0.8446	0.8768	0.8147
2	0.8054	0.8613	0.7564
3	0.9831	0.9907	0.9756
4	0.7441	0.7457	0.7426

(d) Class specific results for FIRE16 + SMERP17 dataset with TF.IDF$_{CNE}$

Class	F$_1$-score	Precision	Recall
1	0.8455	0.8724	0.8201
2	0.8095	0.8623	0.7628
3	0.9802	0.9817	0.9787
4	0.7457	0.7489	0.7426

REFERENCES

[1] Moumita Basu, Anurag Roy, Kripabandhu Ghosh, Somprakash Bandyopadhyay, and Saptarshi Ghosh. 2017. Microblog Retrieval in a Disaster Situation: A New Test Collection for Evaluation. In *Proceedings of the First International Workshop on Exploitation of Social Media for Emergency Relief and Preparedness co-located with European Conference on Information Retrieval, SMERP@ECIR 2017, Aberdeen, UK*. 22–31. http://ceur-ws.org/Vol-1832/SMERP_2017_peer_review_paper_3.pdf

[2] Iyad Batal and Milos Hauskrecht. 2009. Boosting KNN text classification accuracy by using supervised term weighting schemes. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2041–2044.

[3] Constantinos Boulis and Mari Ostendorf. 2005. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. In *Proc. of the International Workshop in Feature Selection in Data Mining*. Citeseer, 9–16.

[4] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[5] George Forman. 2008. BNS feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 263–270.

[6] Saptarshi Ghosh and Kripabandhu Ghosh. 2016. 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*. 56–61. http://ceur-ws.org/Vol-1737/T2-1.pdf

[7] Saptarshi Ghosh, Kripabandhu Ghosh, Debasis Ganguly, Tanmoy Chakraborty, Gareth J.F. Jones, and Marie-Francine Moens. 2017. ECIR 2017 Workshop on Exploitation of Social Media for Emergency Relief and Preparedness (SMERP 2017). *SIGIR Forum* 51, 1 (Aug. 2017), 36–41. https://doi.org/10.1145/3130332.3130338

[8] Samujjwal Ghosh, Srijith P. K., and Maunendra Sankar Desarkar. 2017. Using social media for classifying actionable insights in disaster scenario. *International Journal of Advances in Engineering Sciences* 9, 4 (Dec. 2017), 224–237. https://doi.org/10.1007/s12572-017-0197-2

[9] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. *CoRR* abs/1605.05894 (2016). arXiv:1605.05894 http://arxiv.org/abs/1605.05894

[10] Randy Joy and Magno Ventayen. 2017. Classification of Local Language Disaster Related Tweets in Micro Blogs. In *Asia Pacific Journal of Multidisciplinary Research*.

[11] Prannay Khosla, Moumita Basu, Kripabandhu Ghosh, and Saptarshi Ghosh. 2017. Microblog Retrieval for Post-Disaster Relief: Applying and Comparing Neural IR Models. *arXiv preprint arXiv:1707.06112* (2017).

[12] Christine Largeron, Christophe Moulin, and Mathias Géry. 2011. Entropy based feature selection for text categorization. In *Proceedings of the 2011 ACM Symposium on Applied Computing*. ACM, 924–928.

[13] Hongmin Li, Doina Caragea, Cornelia Caragea, and Nic Herndon. 2017. Disaster response aided by tweet classification with a domain adaptation approach. *Journal of Contingencies and Crisis Management* (2017).

[14] Ying Liu, Han Tong Loh, and Aixin Sun. 2009. Imbalanced text classification: A term weighting approach. *Expert systems with Applications* 36, 1 (2009), 690–701.

[15] Xinghua Lu, Bin Zheng, Atulya Velivelli, and ChengXiang Zhai. 2006. Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association* 13, 5 (2006), 526–535.

[16] Justin Martineau and Tim Finin. 2009. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. *Icwsm* 9 (2009), 106.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[18] Beverly Estephany Parilla-Ferrer, PL Fernandez, and JT Ballena. 2014. Automatic Classification of Disaster-Related Tweets. In *Proc. International conference on Innovative Engineering Technologies (ICIET)*. 62.

[19] Robin L Plackett. 1983. Karl Pearson and the chi-squared test. *International Statistical Review/Revue Internationale de Statistique* (1983), 59–72.

[20] J. R. Ragini and P. M. R. Anand. 2016. An empirical analysis and classification of crisis related tweets. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. 1–4. https://doi.org/10.1109/ICCIC.2016.7919608

[21] Fuji Ren and Mohammad Golam Sohrab. 2013. Class-indexing-based term weighting for automatic text classification. *Information Sciences* 236 (2013), 109–125.

[22] Yang Song, Ding Zhou, Jian Huang, Isaac G Councill, Hongyuan Zha, and C Lee Giles. 2006. Boosting the feature space: Text classification for unstructured data on the web. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 1064–1069.

[23] Hien To, Sumeet Agrawal, Seon Ho Kim, and Cyrus Shahabi. 2017. On Identifying Disaster-Related Tweets: Matching-based or Learning-based?. In *Multimedia Big Data (BigMM), 2017 IEEE Third International Conference on*. IEEE, 330–337.

[24] Hao Wang and Sanhong Deng. 2017. A paper-text perspective: Studies on the influence of feature granularity for Chinese short-text-classification in the Big Data era. *The Electronic Library* 35, 4 (2017), 689–708. https://doi.org/10.1108/EL-09-2016-0192

[25] Tao Wang, Yi Cai, Ho-fung Leung, Zhiwei Cai, and Huaqing Min. 2015. Entropy-based term weighting schemes for text categorization in VSM. In *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*. IEEE, 325–332.

[26] Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*, Vol. 97. 412–420.