Categorization of Event Clusters from Twitter Using Term Weighting Schemes

Surender Singh Samant
Birla Institute of Technology & Science Pilani - Hyderabad Campus, Telangana, India - 500078
Graphic Era (Deemed to be University), Dehradun, Uttarakhand, India - 248002
E-mail: surender.samant@hyderabad.bits-pilani.ac.in, surender.samant@geu.ac.in

N.L. Bhanu Murthy and Aruna Malapati
Birla Institute of Technology & Science Pilani - Hyderabad Campus, Telangana, India - 500078
E-mail: {bhanu, arunam}@hyderabad.bits-pilani.ac.in

Keywords: text categorization, text classification, term weighting

Received: February 14, 2020

A real-world event is commonly represented on Twitter as a collection of repetitive and noisy text messages posted by different users. Term weighting is a popular pre-processing step for text classification, especially when the size of the dataset is limited. In this paper, we propose a new term weighting scheme and a modification to an existing one and compare them with many state-of-the-art methods using three popular classifiers. We create a labelled Twitter dataset of events for exhaustive cross-validation experiments and use another Twitter event dataset for cross-corpus tests. The proposed schemes are among the best performers in many experiments, with the proposed modification significantly improving the performance of the original scheme. We create two majority voting based classifiers that further enhance the F1-scores of the best individual schemes.

Povzetek: V prispevku je opisana kategorizacija gruč dogodkov na Twitterju.

1 Introduction

Twitter is a popular microblogging platform with millions of active users [1] posting (publishing) messages (tweets) every day [18]. In microblogging, there is a limit to the maximum allowed length of a message (e.g. Twitter restricts the length to 280 characters). Since a large number of users access Twitter using mobile devices, real-world news is often shared first on Twitter. In this paper, we consider an event as any newsworthy real-world occurrence discussed on Twitter. For this reason, we use the terms *event* and *news* interchangeably. There can be a large number of tweets discussing an event. The set of event-related tweets has very high-dimensional vocabulary (features), is repetitive and noisy. We refer to a collection of related tweets (in English) discussing an event as an event cluster, an event or a document.

The number of real-world events that are detected online during a fixed time duration is generally limited. For example, Kalyanam et al. [7] were able to detect about 5000 real-world events in a year (including duplicates). Since event datasets are not huge, advanced neural network techniques are not applicable, and we need to use traditional methods for classification.

Term-weighting schemes have traditionally been one of the most popular pre-processing methods for text categorization. These schemes are applicable even when the dataset is not very big. There are two types of term-weighting: unsupervised term weighting (UTW) and supervised term weighting (STW). UTW schemes such as *tf*idf* do not consider the category of a term's containing-document, while STW schemes depend on the category information. A classifier is trained on the labelled dataset consisting of documents with weighted words and the corresponding document category labels. Fig. 1 gives an overview of event categorization using term weighting schemes. Note that event category is a conceptual grouping that contains a similar type of events (*e.g.* sports category). Human annotators assign a category label to each event (we discuss the process in Section 4).

We ask the following research questions in this paper. Can the existing term weighting schemes categorize noisy and repetitive Twitter event clusters effectively? Can we create a new term weighting scheme or improve existing ones? Would the proposed method and modification be effective in general text categorization? Can we improve event categorization by creating voting classifiers using term weighting schemes?

To this end, we make the following contributions:

- We propose a new term weighting scheme and a modification to an existing scheme for event categorization. We perform cross-validation and cross-corpus classification using two different datasets. We also evaluate the proposed schemes on multiple balanced and imbalanced sub-datasets.

- We show that the proposed term weighting schemes

[1] www.twitter.com

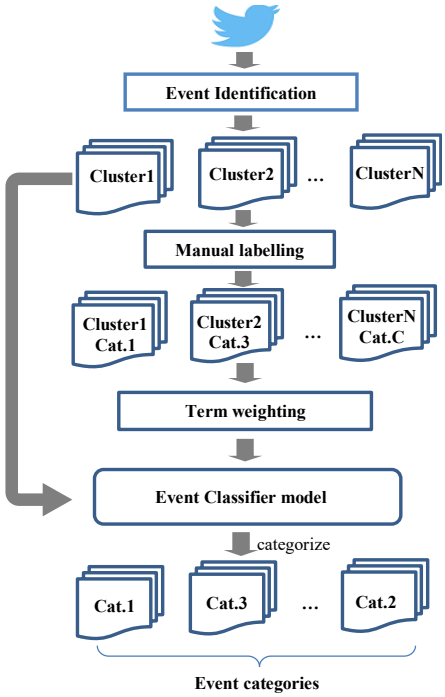

Figure 1: Overview of the process of Twitter event categorization.

and an existing scheme statistically make significantly different predictions. Consequently, we propose a voting-based classifier using these schemes that achieves the highest F1-scores.

We have organized the rest of this paper as follows. We discuss the related work and existing term weighting schemes in Section 2. We propose a term weighting scheme in Section 3.1 and modification to an existing scheme in Section 3.2. In Section 4, we discuss the datasets, experimental setup and the evaluation metrics. We present experimental results and analysis in Section 5 followed by conclusion in Section 6.

2 Related work

We now discuss a few state-of-the-art term weighting schemes that have been previously used by researchers for text categorization.

2.1 Unsupervised methods

These methods make an assumption that terms important to a document are frequently present in it. Another common assumption is that a term is important to a document if it is not present in many documents. Raw count (frequency) *tf* and its variations are often used as term weighting schemes. The *tf*idf* and its variations additionally consider inverse document frequency (*idf*) for weight assignment. The *idf* of a term is defined in inverse proportion to the fraction of documents in which it is present. Let N be the total number of documents in a corpus, and n be the number of documents that contain a term t, then (1) is used to calculate its *tf*idf*.

$$\text{w} = tf * idf, \text{ where idf} = log\Big(\frac{N}{n}\Big) \tag{1}$$

There are a few popular variants of *tf*. Binary weight can be 1 or 0 depending on whether a term is present or absent in a document:

$$\text{w} = \begin{cases} 1, \text{if } tf > 0 \\ 0, \text{otherwise} \end{cases} \tag{2}$$

Another variant is calculated by dividing *tf* by the length of the document. Yet another variation is the log-normalized *tf* in which the log of *tf* is calculated, as shown in (3).

$$\text{w} = log(1 + tf) \tag{3}$$

There are many variations of *idf* scheme. In smoothed *idf*, weight is calculated using (4).

$$\text{smooth-idf} = log\Big(1 + \frac{N}{n}\Big) \tag{4}$$

Probabilistic *idf* is calculated using (5)

$$\text{idf} = log\frac{(N-n)}{n} \tag{5}$$

These schemes have been used by many researchers for text categorization ([6], [22]).

2.2 Supervised methods

Assume there are N training documents in $|C|$ categories, the following notations are used in STW schemes:

- *tp*: number of positive category documents that contain *w*.
- *fp*: number of positive category documents that do not contain *w*.
- *tn*: number of negative category documents that do not contain *w*.
- *fn*: number of negative category documents that contain *w*.
- *cf*: number of categories that have *w* present in at least one document.

Debole and Sebastiani [4] compared unsupervised *tf*idf* method with supervised term weighting methods *tf*χ2*, *tf*ig* and *tf*gr* using (6), (7), and (8), respectively. They used news categorization dataset Reuters-21578 and found *tf*gr* to perform the best among the three schemes. The results were mostly inconclusive when compared to *tf*idf*.

$tf*\chi^2$ measures independence of a term from a category.

$$tf*\chi^2 = \frac{tf * N * (tp * tn - fp * fn)^2}{(tp + fp)(fn + tn)(tp + fn)(fp + tn)} \tag{6}$$

*tf*ig* measures the information a term contains about a category.

$$\begin{aligned} tf*ig = tf * \Big(&- \frac{tp+fp}{N}\log\frac{tp+fp}{N} - \frac{fn+tn}{N}\log\frac{fn+tn}{N} \\ &+ \frac{tp}{N}\log\frac{tp}{tp+fn} + \frac{fn}{N}\log\frac{fn}{tp+fn} \\ &+ \frac{fp}{N}\log\frac{fp}{fp+tn} + \frac{tn}{N}\log\frac{tn}{fp+tn}\Big) \end{aligned} \tag{7}$$

*tf*gr* is similar to *tf*ig* with a normalization factor added to give weights to a term on an equal basis across different categories [21].

$$tf{*}gr = \frac{tf * ig}{-\frac{tp+fp}{N}\log\frac{tp+fp}{N} - \frac{fn+tn}{N}\log\frac{fn+tn}{N}} \quad (8)$$

Lan et al. [8], the authors proposed a term weighting method based on the relevance frequency (*rf*) of a term. They argued that relevance of a term in a document should only be affected by *tp* and *fn*, while *tn* and *fp* should not have any role in term weighting. They used k-NN and SVM classifiers on Reuters-21578 and 20 Newsgroups datasets to compare the methods against other supervised and unsupervised methods. The weight of a term in *tf*rf* is calculated using (9).

$$tf{*}rf = tf * \log\Big(2 + \frac{tp}{Max(1, fn)}\Big) \quad (9)$$

An Odds Ratio based method *tf*OR* has been found to perform well by researchers [8, 15]. Odds-Ratio is used to measure the strength of association between a term and a category. It is calculated by using (10).

$$tf{*}OR = tf * \log\Big(\frac{tp * tn}{fp * fn}\Big) \quad (10)$$

Quan et al. [15] proposed *iqf*qf*icf* (inverse question frequency, question frequency, inverse category frequency) for the question categorization task. They argued that words in a question mostly have *tf* of 1, which is equivalent to using binary features (presence or absence of words). Hence, the scheme did not use *tf*. The performance of this scheme in news categorization and general document categorization was better than other schemes. Term weights in *iqf*qf*icf* are calculated using (11).

$$\begin{aligned} iqf{*}qf{*}icf &= \log\Big(\frac{N}{tp+fn}\Big) * \log(tp+1) * icf \\ &\text{where } icf = \log\Big(\frac{|C|}{cf} + 1\Big) \end{aligned} \quad (11)$$

Wu et al. [20] proposed a scheme called regularized-entropy that attempts to avoid overweighting and underweighting of terms. They reported that the scheme gives better results on multiple text categorization and sentiment analysis dataset as compared to schemes such as *tf*$*\chi 2$, *tf*ig*, and *tf*rf*. In this method, (12) is used to compute the weight of a term.

$$\begin{aligned} &g = b + (1-b) * (1-h) \text{ , where} \\ &b \in [0,1] \text{ tradeoff between over/under weighting} \\ &h = -p^{+} * \log p^{+} - p^{-} * \log p^{-} \text{ , where} \\ &p^{+} = -\frac{tp/(tp+fp)}{tp/(tp+fp) + tp/(fn+tn)}, \\ &p^{-} = -\frac{fn/(fn+tn)}{tp/(tp+fp) + tp/(fn+tn)} \end{aligned} \quad (12)$$

Apart from term weighting schemes, other types of term weighting methods have similarly been proposed in the literature. [9] and [13] proposed term-weighting schemes suited for imbalanced datasets. A graph-based term weighting scheme was proposed by Malliaros et al. [11], in which documents are represented as graphs that encode relationships between the different terms. Wang et al. [19] proposed entropy-based term weighting schemes that use a term's global distributional concentration in the categories to measure its discriminating power. Reed et al. [17] proposed a term weighting scheme called term frequency-inverse corpus frequency (*tf-icf* for clustering of document streams. They used this weighting scheme for unsupervised document clustering. An interesting non-conventional method was proposed by Escalante et al. [5]. In contrast to other schemes, their method uses genetic programming to learn effective term weighting schemes.

Category size	*tp*	*tpr*
200	100	500
50	25	500

Table 1: Example of *tp* bias towards important terms in bigger categories.

In this paper, we use *log* to mean log_2. We use cosine normalization in which a term t_i in document D is converted into its cosine normalized form using (13). It is done to prevent the terms in bigger events from overwhelming the terms in smaller events.

$$t_{cosine} = \frac{t_i}{\sqrt{\sum_{i \in D} t_i^2}} \quad (13)$$

3 Proposed schemes

In this section, we propose a term weighting scheme specific to imbalanced datasets, and two improvements to existing schemes.

3.1 Proposed method

The first observation is that the many existing term weighting schemes do not consider the imbalance of categories in a dataset into account. As a result, *tp* suffers from a bias towards the words in bigger categories with more events. The range of values for *tp* is much smaller for categories with fewer events (smaller categories) than bigger categories. If we use *tp* directly in computing weights, it is likely to give higher weights to terms (words) in the bigger categories.

Table 1 shows an example where the smaller category contains 50 events and the bigger includes 200. Let us call a term *important* to a class if it is present in half of the category documents. In the example, the term relevant to the smaller category has *tp* of 25, while the term relevant to the bigger group has *tp* of 100. All terms important in more prominent categories have higher *tp* than equally important words in smaller classes. As *tp* is a component of many term weighting schemes, this may lead to a scenario where events belonging to the smaller categories get wrongly classified as a bigger category.

From this observation, we need to assign weights inversely proportional to the size of the category. If N is the

total number of events, we introduce *tp* ratio (*tpr*) computed as (14).

$$tpr = tp * \frac{N}{tp + fp} \tag{14}$$

In Table 1, with N as 1000, the terms equally important to their respective categories now have the same *tpr* value 500. The *tpr* component has removed the bias of *tp* towards the bigger categories.

The second observation is that there should be a penalty factor for a term if and only if it is present in negative category documents. The presence of a term in positive category documents should not be considered in computing the penalty factor. This is in contrast to *tf*idf* and its derivative schemes that penalize a term solely based upon the number of the containing documents, irrespective of the category. This observation leads to the penalty factor *ifn* (inverse fn) computed by (15) where *Cn* is the number of documents in the negative category. Note that in *ifn*, the penalty is proportional to the size of the negative category. This ensures that bigger category documents are not disproportionately penalized. In (15), we perform add-one smoothing to avoid zero division. Also, 1 is added before log calculation to avoid a term weight from becoming zero due to penalty factor.

$$ifn = \log\Big(\frac{Cn + 1}{fn + 1} + 1\Big) \tag{15}$$

Combining (14) and (15) with term frequency *tf*, we propose the term weighting scheme given by (16)

$$proposed = tf * \log(tpr + 1) * \log\Big(\frac{Cn + 1}{fn + 1} + 1\Big) \tag{16}$$

The first component in the proposed scheme assigns weight locally within a document. It assigns higher weights to the more frequent terms in a document. The remaining part of the equation are the category level global components. They assign higher weights to the terms that are present in the more positive category documents, but penalize terms that are present in the negative category documents.

3.2 Proposed modification to $\chi 2$

The $\chi 2$ based term weighting scheme results in a disproportionate increase in weight even for a small increase in *tp*. We can see this in Fig. 2 where we compare the term weights assigned by $\chi 2$ and the Odds-Ratio (OR) scheme. The weight assigned by $\chi 2$ varies much faster with an increase in *tp* resulting in overweighting of terms and ultimately affects the classifier accuracy.

Another problem with $\chi 2$ scheme is shown with an example in Table 2. In the example, the number of documents is 1000. The term *t2* is assigned a higher weight than term *t1* even though *t1* has a higher *tp* and lower *fn* as compared to *t2*. Ng. et al. [14] have noted that $\chi 2$ (see (6)) not only picks out the set of words indicative of category membership but also those words indicative of non-membership. They suggested using the square root of $\chi 2$ (correlation coefficient CC) as it gives more weight to words that are highly indicative of category membership. We observe that

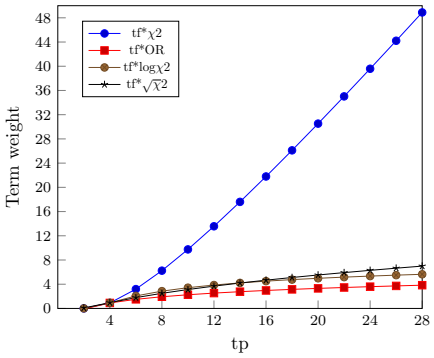

Figure 2: Term weights assigned by $\chi 2$ and others as *tp* is varied (assume that *N*=1000, *tn*=695, *fn*=5). Weights assigned by $log\chi 2$ and *OR* are comparable.

Term	*tp*	*fp*	*tn*	*fn*	*tf*χ2*
t1	20	280	695	5	30.52
t2	2	298	600	100	42.52

Table 2: Example of inappropriate weight assignment by $\chi 2$.

the log of $\chi 2$ gives better accuracy as it smoothens the weights more than CC (Fig. 2).

Table 3 summarizes the different term weighting schemes used in our research.

4 Datasets and experimental setup

4.1 Twitter datasets

The only public Twitter dataset (of tweet ids) of sufficient size contains 506 event clusters [12]. As the event classification task needs a bigger dataset, we labelled a news dataset by Kalyanam et al. [7]. Also, we created many sub-datasets from the self-labelled dataset for a more extensive set of experiments. We now briefly discuss the Twitter datasets and the sub-datasets.

The dataset by Kalyanam et al. [7] contains collections of event clusters. There are more than 40 million tweets collected during the period from August 2013 to June 2014. It covers 5234 news events in chronological order. As the collection contains an enormous volume of tweet ids, we requested from Twitter the first 1000 tweets for each event cluster.

We manually labelled the dataset of 5234 event clusters into eight categories using the following process. We discarded duplicate event clusters, clusters containing less than ten tweets, and other clusters that did not contain real-world events. Two human annotators labelled the remaining event clusters. We selected 1461 event clusters (Events1461) on which there was an agreement between the two annotators. The annotators have a good agreement with Kohen's Kappa of 0.8. The categories are the same as used by [12] and partly resemble the categories used by

Method	Type	Brief description
tf	Unsupervised	log of frequency count
*tf*idf*	Unsupervised	popular method in IR
tfχ2*	Supervised	$\chi 2$
*tf*ig*	Supervised	information gain
*tf*gr*	Supervised	gain ratio
*tf*OR*	Supervised	Odds Ratio
*tf*rf*	Supervised	relevance frequency
*iqf*qf*icf*	Supervised	uses category frequency
proposed	Supervised	proposed scheme
*tf*logχ2*	Supervised	log *tf*χ2*
voting	Supervised	two voting-based schemes

Table 3: List of term weighting schemes.

Notation	Category Name; Examples
law	Law, politics, and scandals
spo	Sports; players, clubs, etc.
arm	Armed conflicts & attacks; terrorism
bus	Business & Finance; mergers
arts	Arts & entertainment; actors, movies
dis	Disasters; floods, hurricanes
sci	Science & technology; space, phone launch
misc	Miscellaneous; Pope's visit, Queen's birthday

Table 4: The eight categories of events.

online news aggregators such as Google News [2]. The eight categories are shown in Table 4.

For cross-corpus evaluation, we use the pre-labelled Events2012 dataset by [12]. It contains 506 events labelled into eight categories. After pre-processing, only 384 suitable events remain (Events384).

Fig. 3 shows the distribution of events in different categories in the two datasets.

4.1.1 Sub-datasets

The first sub-dataset contains equal number of events from each category. We randomly selected 90 events from each category of Events1461 to create the dataset. This sub-dataset is used to evaluate the performance of the proposed schemes on balanced datasets. A good score on the dataset would suggest that the proposed term-weighting schemes are overall good performers on any kind of dataset.

For more extensive experiments to test the robustness of the different schemes on balanced and imbalanced datasets, we created many sub-datasets out of Events1461 as follows. We created six sub-datasets from Events1461 with top 2, 3, 4, 5, 6, and 7 categories having the most events. We created the another sub-dataset by splitting the shuffled Events1461 in a stratified manner into seven increasing-sized sub-datasets. The first sub-dataset contains 30% of events from each category and each subsequent sub-dataset adds 10% additional event clusters. So, the biggest sub-dataset contains 90% of the events from each category.

[2]http://news.google.com

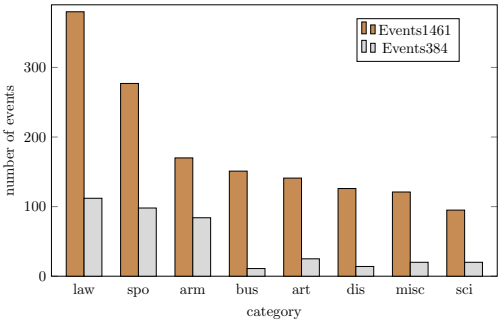

Figure 3: Number of events in each category of Events1461 and Events384 datasets.

4.1.2 Normalized event clusters

We use a centroid-based approach ([16], [1]) to extract the T most relevant tweets from each event. In this, we first compute the centroid of the cluster, followed by the extraction of tweets nearest to the centroid (one by one) in decreasing order of cosine similarity to the centroid. To avoid near-duplicates, we select a tweet only if it is less than 80% similar to already selected tweets. Finally, each event cluster contains an equal number of relevant non-duplicate tweets. We call them normalized clusters as the events contain an equal number of tweets. We used T from the set {10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200}.

4.2 Experimental setup

We briefly discuss the pre-processing steps, classifiers and policy used in the experiments.

4.2.1 Preprocessing

For all experiments, discard all the information except the text of the tweet. The removed information includes timestamp, tweet identity and user identity. We remove stopwords and perform stemming using Porter stemmer. We discard a word present in less than two documents (event clusters) as non-informative. Also, we discard a word as noise if it is present in more than half the documents.

We use three different classifiers: Support Vector Machine (SVM), Logistic Regression (LR), and K-Nearest Neighbours (k-NN) as these have been used widely by researchers ([8], [15], [2]).

We use the MAX global policy as it has given the best performance in earlier studies ([4]). In MAX, all categories share a common weight vector TW. Each term weight in the vector is the maximum of the term's weights among all classes.

5 Experiments and results

We now present the experimental results.

5.1 Cross validation experiments using Events1461

First, we discuss the results of the experiments on the self-labelled Events1461 dataset and its sub-datasets.

We performed 10-fold cross-validation (10-CV) experiments on the full Events1461 dataset. Using grid-search, we selected the hyperparameters for the SVM and logistic regression classifiers. For SVM, the linear kernel performs the best with C=10, and for logistic regression, solver='newton-cg' and C=1000 gives the best scores. For k-NN classifier, we used different values of k, as shown in Fig. 4. The value of k=7 gives the best scores for most of the term weighting schemes.

Table 5 displays the 10-CV scores of the term weighting schemes on the full Events1461 dataset. Looking at the macroF1 scores, *tf*OR*, the proposed scheme, *tf*log*χ2 and *tf*rf* have scored better than other schemes. We call these better performing schemes *strong* term weighting schemes. The proposed modification has significantly improved the performance of the original scheme. As SVM has given the best scores in this experiment, we report results based on SVM for the remaining 10-CV experiments.

An interesting observation is that *tf* macroF1 scores are better than *tf*idf*. Apart from signifying the importance of *tf* in event categorization, it also highlights the limitation of *tf*idf* in this context. The *idf* component adversely affects the score as it penalizes the terms present in other documents without considering the category information.

5.1.1 Cross validation on sub-datasets

Table 6 shows the results of the term-weighting schemes using the balanced Twitter dataset. The proposed term-weighting schemes have given the best overall score on the sub-dataset. Among the existing schemes, *tf*OR* and *tf*rf* have given achieved F1-scores. These results show that the proposed scheme performs well on balanced datasets.

The second set of experiments tests the performance of STW schemes on the other two sub-datasets (described in Section 4.1.1). Fig. 5 shows the 10-CV scores of different STW methods for various subset sizes. Fig. 6 shows the 10-CV scores in sub-datasets containing different number of categories. To remove clutter from the figures, we have not shown scores of *tf*, *tf*ig* and *tf*gr* as these schemes have low scores. Among the *weaker* schemes, we show scores of the *tf*idf* and *tf**χ2 scheme for comparison. In these figures, the horizontal axis signifies the percentage of events taken from Events1461.

Both the microF1 and macroF1 scores improve with the size of the subsets. It is as expected since the term weighting schemes can perform better weight assignment with the increase in the number of event clusters. The proposed scheme *tf*log*χ2 has good scores. As expected, the scores of the term weighting schemes monotonically increase as the number of categories decrease.

	SVM		k-NN		LR	
	Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
tf	83.19	80.40	81.16	79.25	82.14	79.12
tf*idf	83.27	80.28	80.36	77.69	81.49	78.01
tf*χ2	81.20	79.03	77.08	74.23	81.70	79.16
tf*ig	81.01	78.10	76.54	72.11	81.32	78.43
tf*gr	80.97	78.20	77.94	75.05	81.75	79.33
tf*OR	**84.20**	**81.68**	81.88	79.38	**83.66**	**80.86**
tf*rf	83.75	80.98	**83.49**	**81.45**	82.63	79.14
tf*logχ2	83.96	81.27	81.51	79.02	83.19	80.36
iqf*qf*icf	83.92	80.93	82.58	80.26	82.04	78.66
proposed	84.13	81.55	82.72	80.65	82.86	79.65

Table 5: 10-fold cross-validation scores of the term weighting schemes using the three classifiers on the full Events1461 dataset.

	SVM		k-NN		LR	
	Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
tf	72.22	70.88	69.88	67.12	71.32	69.96
tf*idf	72.89	71.06	70.80	69.16	72.14	70.11
tf*χ2	71.80	70.78	70.12	68.98	72.28	70.12
tf*ig	71.21	69.88	70.42	68.36	71.34	68.42
tf*gr	71.77	70.02	70.44	69.08	72.54	70.03
tf*OR	75.12	73.38	73.98	71.86	74.66	**72.66**
tf*rf	75.08	72.88	74.20	**72.99**	74.78	72.43
tf*logχ2	74.88	72.24	73.25	71.24	74.09	72.59
iqf*qf*icf	74.81	72.83	73.95	71.86	73.14	71.52
proposed	**75.17**	**73.55**	**74.23**	72.88	**74.86**	72.56

Table 6: 10-fold cross-validation scores of the term weighting schemes using the three classifiers on the balanced sub-dataset of Events1461.

5.1.2 Cross-validation on normalized event clusters

In this experiment, we used the normalized event clusters with different number of tweets (see section 4.1.2).

Fig. 7 displays the results. The proposed scheme has given the best scores in many normalized clusters. The *tf*log*χ2 and *tf*OR* schemes have also performed well. Specifically, the proposed scheme and *tf*log*χ2 have the best macroF1 scores for most of the normalized clusters. Among the normalized subsets of events, none has better scores than the full dataset. Hence, we use the full dataset for the remaining experiments.

We used the normalized event clusters to compare the scores of the raw term count, binary count (1/0 for presence/absence of a term) and log_2(raw count) as *tf* on the proposed scheme. Fig. 8 shows the results using the SVM classifier. The log_2(raw count) as *tf* has given the best scores, while raw count as *tf* has the worst. We have observed similar results for all the schemes that use *tf*. In fact, the baseline *tf*idf* using log_2(raw count) as *tf* outperforms many STW schemes using raw count as *tf*. Hence, we have used log_2(raw count) as *tf* in this work.

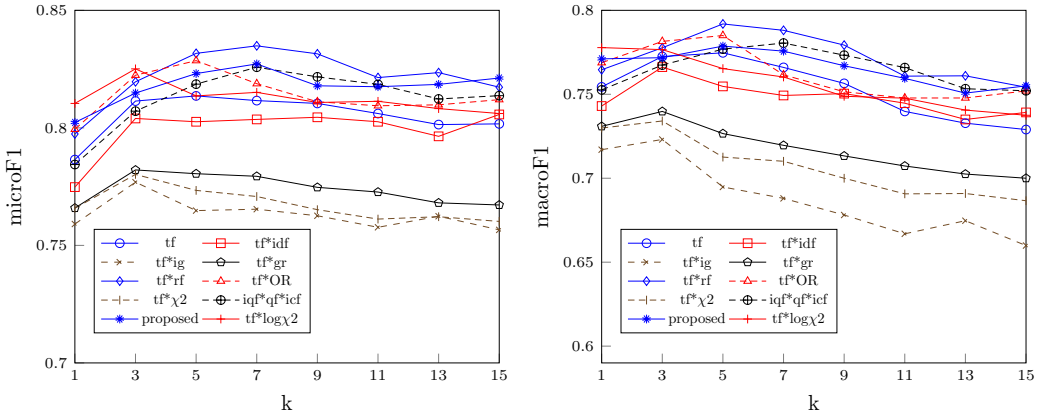

Figure 4: MicroF1 and macroF1 scores for different values of k in k-NN.

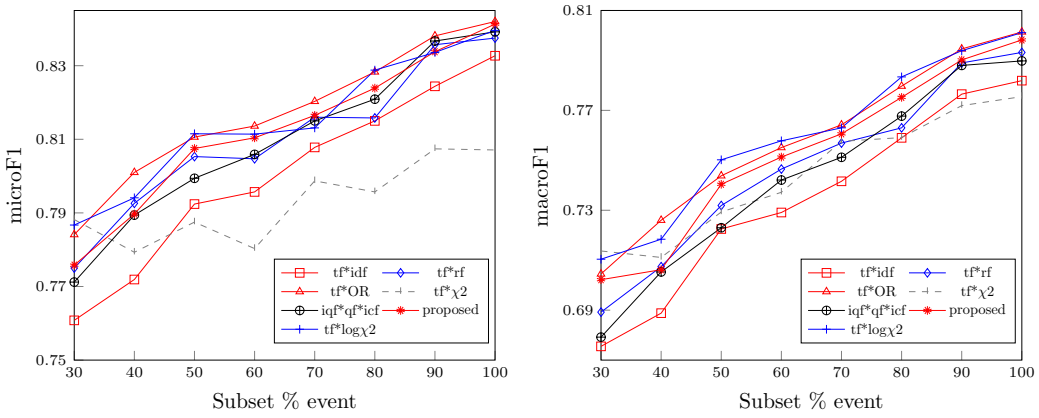

Figure 5: 10-CV scores of the term weighting schemes with subsets of different sizes. The x-axis represents the percentage of events used for each category.

5.2 Cross-corpus event classification

We performed this experiment to test the generalization capability of the term weighting schemes. We use Events1461 dataset for training and Events384 for testing. The two datasets contain the same categories of events from two non-overlapping periods, resulting in low statistical dependence of text. Table 7 shows the results using the SVM, k-NN and logistic regression classifiers. We use the same set of hyperparameters for the classifiers, as discussed in section 5.1.

The results are shown in Table 7. The *tf*OR* and *tf*log*χ2 schemes have the best macroF1 scores, followed by the proposed scheme. Among the classifiers, k-NN has the best categorization scores. Overall, the cross-corpus categorization scores are good considering the fact that the datasets used for training and testing are from different times and labelled by two unrelated groups of annotators.

	SVM		**k-NN**		**LR**	
	Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
tf	79.47	72.28	82.96	75.11	79.33	71.37
tf*idf	78.07	72.75	79.89	75.88	72.85	61.57
tf*χ2	77.79	70.40	79.01	72.99	78.57	71.88
tf*ig	76.22	69.75	78.43	71.21	78.67	71.17
tf*gr	77.78	72.22	77.64	71.09	78.92	72.26
tf*OR	82.53	**79.29**	83.63	80.85	79.20	73.57
tf*rf	81.47	74.02	83.93	76.60	80.18	70.30
tf*logχ2	**82.64**	78.35	**84.72**	**80.97**	**80.77**	**74.36**
iqf*qf*icf	80.00	72.62	82.45	78.52	77.09	66.95
proposed	80.48	74.65	83.58	79.82	77.74	70.24

Table 7: Cross-corpus F1-scores where Events1461 is used for training and Events384 for testing.

5.3 General text categorization

We use three standard datasets to evaluate the performance of the term weighting schemes in general text categorization: 20 Newsgroups (20NG), Cade12, and WebKB. We remove the headers from the email documents in 20NG as they contain category information. The other two datasets are pre-processed versions available from the research by [3]. Cade12 contains twelve categories of documents of Brazilian web pages while WebKB contains four categories of webpages of computer science departments from different universities.

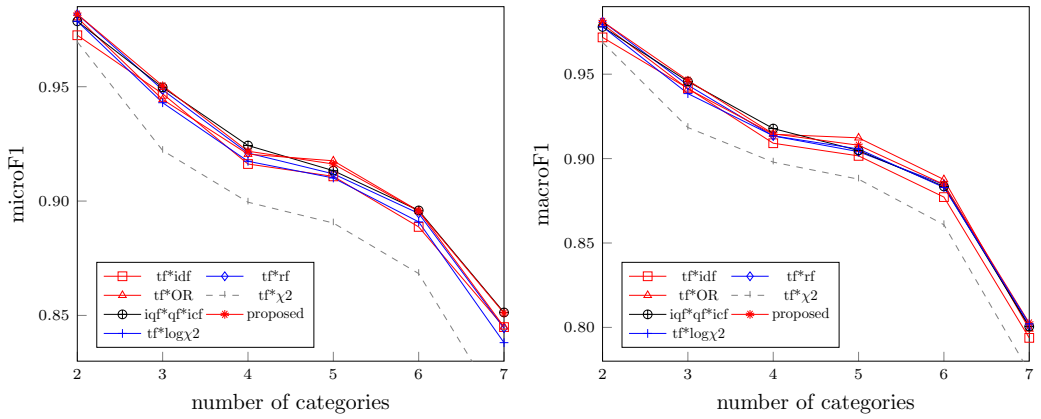

Figure 6: 10-CV scores of the term weighting schemes with different number of categories.

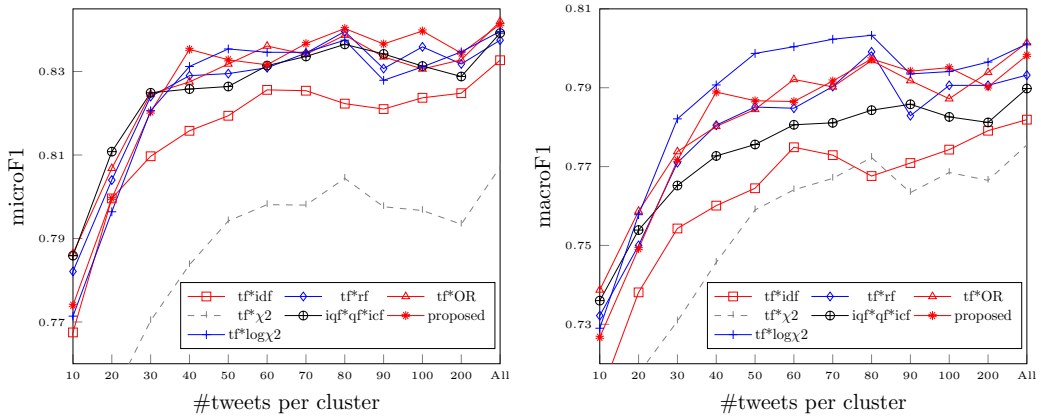

Figure 7: 10-fold cross-validation scores using SVM classifier on the normalized event clusters.

	20NG		Cade12		WebKB	
	Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
tf	75.62	74.91	56.00	49.54	90.86	89.60
tf*idf	76.29	75.67	55.92	49.41	90.01	88.40
tf*χ2	75.82	75.01	52.50	46.93	88.02	86.59
tf*ig	74.64	73.66	51.23	44.99	87.98	86.60
tf*gr	74.85	73.94	51.75	45.85	87.84	86.52
tf*OR	**79.54**	**78.90**	**57.30**	**51.35**	90.74	89.63
tf*rf	78.74	78.11	56.92	50.47	**90.85**	**89.73**
tf*logχ2	77.12	76.40	56.76	50.70	90.36	89.32
iqf*qf*icf	79.16	78.60	57.11	50.79	90.69	89.33
proposed	78.28	77.61	57.03	51.09	90.65	89.23

Table 8: F1-scores of term weighting schemes on the standard text categorization datasets.

We use the SVM classifier for this experiment. Table 8 displays the results. The *tf*OR* scheme has yet again proven to be a versatile scheme with good macroF1 scores. The proposed scheme has better scores in Cade12 and WebKB, but its scores in 20NG are lower than other *strong* schemes. The results achieved on general text categorization are different from event categorization.

	idf	χ2	ig	gr	OR	rf	logχ2	iqf	proposed
tf	y	y	y	y	y	y	y	y	y
tf*idf		y	y	y	y	y	y	y	y
tf*χ2			y	y	y	y	y	n	y
tf*ig				y	y	y	n	y	y
tf*gr					y	y	n	y	y
tf*OR						y	y	y	y
tf*rf							y	y	y
tf*logχ2								y	y
iqf*qf*icf									y

Table 9: Pairwise significance difference with p-value of 0.05. y represent a significant difference, while n represents no difference in predictions.

5.4 Voting-schemes based classifiers

We used Mcnemar's test with continuity correction to measure the statistically significant difference between predictions of term weighting schemes. This is a standard test that researchers have used to compare two classifiers ([8], [15]).

Since this test needs a dataset with many categories, with each having hundreds of examples, we use 20NG. Table 9 displays the results of this test. For most pairs of

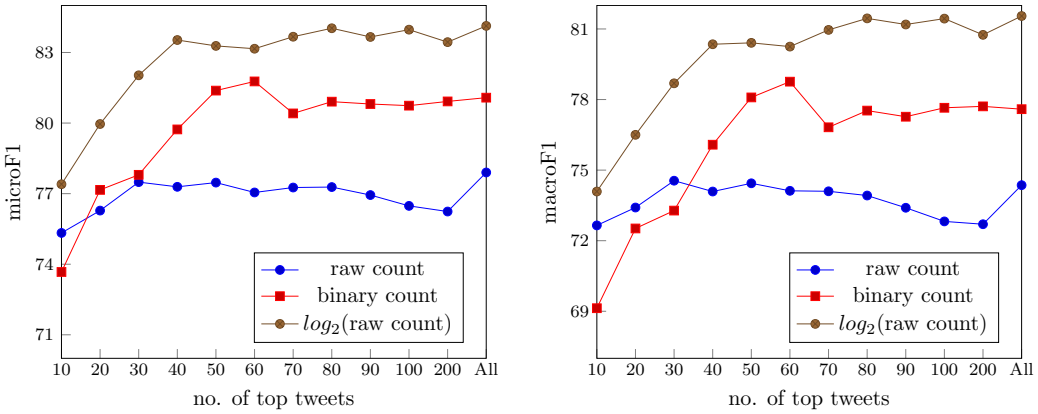

Figure 8: Comparison of raw count (tf), binary and log_2(tf) schemes using SVM.

	SVM		k-NN	
	Micro	*Macro*	*Micro*	*Macro*
best-scores	82.64	79.29	84.72	80.97
Voting	83.28	80.41	84.97	81.25

Table 10: Result of cross-corpus test using voting classifiers.

term weighting schemes, it shows statistical significance. Specifically, our proposed scheme, *tf*log*χ2 and *tf*OR* disagree significantly in their predictions.

We create a voting classifier [10] from these schemes. We use SVM and k-NN classifiers. In a voting classifier, the category label of an event cluster is selected by majority decision among the three schemes (*i.e.* each cluster's category prediction is common to at least two of the schemes).

We repeat the cross-corpus test using the voting classifiers. Table 10 displays the F1-scores. The row labelled *best-scores* represents the best individual scores in cross-corpus test (from Table 7) by the respective classifiers. Both the voting classifiers have given achieved better scores than the individual schemes. We can say that the voting classifiers may be effective in event classification.

6 Conclusion and future work

In our experiments, among the existing term weigthing schemes, *OR* has the best performance, followed by *iqf*qf*icf* and *RF*. We observed that the proposed method has given good F1-scores and the proposed modification to χ2 improves the classification scores of the original scheme. We also observed that voting-based classifiers created from the term weighting schemes that significantly differ in their predictions have the best F1-score. We may suggest *OR* and the proposed schemes for improving the classifier performance on similar datasets. However, a limitation of the study is the use of a self-labelled twitter dataset and a few standard datasets. Hence, even though the results of this study are encouraging, further experiments need to be conducted in the future on more datasets.

References

[1] N. Alsaedi, P. Burnap, and O. F. Rana. Automatic summarization of real world events using twitter. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 511–514, 2016.

[2] W. Cao. Application of support vector machine algorithm based gesture recognition technology in human-computer interaction. *Informatica (Slovenia)*, 43(1), 2019. doi: 10.31449/inf.v43i1.2602.

[3] A. Cardoso-Cachopo. Improving methods for single-label text categorization, 2007. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

[4] F. Debole and F. Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, pages 784–788, New York, NY, USA, 2003. ACM. ISBN 1-58113-624-2. doi: 10.1145/952532.952688.

[5] H. J. Escalante, M. A. García-Limón, A. Morales-Reyes, M. Graff, M. Montes-y Gómez, E. F. Morales, and J. Martínez-Carranza. Term-weighting learning via genetic programming for text classification. *Know.-Based Syst.*, 83(C):176–189, July 2015. doi: 10.1016/j.knosys.2015.03.025.

[6] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML'98, pages 137–142, Berlin, Heidelberg, 1998. Springer-Verlag. ISBN 3-540-64417-2, 978-3-540-64417-0. doi: 10.1007/BFb0026683.

[7] J. Kalyanam, M. Quezada, B. Poblete, and G. Lanckriet. Prediction and characterization of high-activity events in social media triggered by real-world news. *PLOS ONE*, 11(12):1–13, 12 2016. doi: 10.1371/journal.pone.0166694.

[8] M. Lan, C. L. Tan, and H. Low. Proposing a new term weighting scheme for text categorization. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 763–768, 2006.

[9] Y. Liu, H. Loh, and A. Sun. Imbalanced text classification: A term weighting approach. *Expert Syst. Appl.*, 36:690–701, 2009.

[10] I. E. Livieris. A new ensemble self-labeled semi-supervised algorithm. *Informatica (Slovenia)*, 43(2), 2019. doi: 10.31449/inf.v43i2.2217.

[11] F. D. Malliaros and K. Skianis. Graph-based term weighting for text categorization. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1473–1479, Aug 2015. doi: 10.1145/2808797.2808872.

[12] A. J. McMinn, Y. Moshfeghi, and J. M. Jose. Building a large-scale corpus for evaluating event detection on twitter, 2013.

[13] B. Naderalvojoud, E. A. Sezer, and A. Ucan. Imbalanced text categorization based on positive and negative term weighting approach. In P. Král and V. Matousek, editors, *Text, Speech, and Dialogue - 18th International Conference, TSD 2015, Pilsen,Czech Republic, September 14-17, 2015, Proceedings*, volume 9302 of *Lecture Notes in Computer Science*, pages 325–333. Springer, 2015. doi: 10.1007/978-3-319-24033-6_37.

[14] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '97*, pages 67–73, 1997. doi: 10.1145/258525.258537.

[15] X. Quan, L. Wenyin, and B. Qiu. Term weighting schemes for question categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1009–1021, May 2011. doi: 10.1109/TPAMI.2010.154.

[16] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Inf. Process. Manage.*, 40(6):919–938, Nov. 2004. doi: 10.1016/j.ipm.2003.10.006.

[17] J. W. Reed, Y. Jiao, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *2006 5th International Conference on Machine Learning and Applications (ICMLA'06)*, pages 258–263, Dec 2006. doi: 10.1109/ICMLA.2006.50.

[18] P. Tiwari, H. M. Pandey, A. Khamparia, and S. Kumar. Twitter-based opinion mining for flight service utilizing machine learning. *Informatica (Slovenia)*, 43(3), 2019. doi: 10.31449/inf.v43i3.2615.

[19] T. Wang, Y. Cai, H. Leung, Z. Cai, and H. Min. Entropy-based term weighting schemes for text categorization in vsm. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 325–332, Nov 2015. doi: 10.1109/ICTAI.2015.57.

[20] H. Wu, X. Gu, and Y. Gu. Balancing between over-weighting and under-weighting in supervised term weighting. *Inf. Process. Manage.*, 53(2):547–557, Mar. 2017. doi: 10.1016/j.ipm.2016.10.003.

[21] F. Yang. Decision tree algorithm based university graduate employment trend prediction. *Informatica (Slovenia)*, 43(4), 2019. doi: 10.31449/inf.v43i4.3008.

[22] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3.