We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Multiple Kernel-Based Multimedia Fusion for Automated Event Detection from Tweets

Suhuai Luo, Samar M. Alqhtani and Jiaming Li

Additional information is available at the end of the chapter

Abstract

A method for detecting hot events such as wildfires is proposed. It uses visual and textual information to improve detection. Starting with picking up tweets having texts and images, it preprocesses the data to eliminate unwanted data, transforms unstructured data into structured data, then extracts features. Text features include term frequency-inverse document frequency. Image features include histogram of oriented gradients, gray-level co-occurrence matrix, color histogram, and scale-invariant feature transform. Next, it inputs the features to the multiple kernel learning (MKL) for fusion to automatically combine both feature types to achieve the best performance. Finally, it does event detection. The method was tested on Brisbane hailstorm 2014 and California wildfires 2017. It was compared with methods that used text only or images only. With the Brisbane hailstorm data, the proposed method achieved the best performance, with a fusion accuracy of 0.93, comparing to 0.89 with text only, and 0.85 with images only. With the California wildfires data, a similar performance was recorded. It has demonstrated that event detection in Twitter is enhanced and improved by combination of multiple features. It has delivered an accurate and effective event detection method for spreading awareness and organizing responses, leading to better disaster management.

Keywords: data fusion, data mining, event detection, kernel method, multiple kernel learning, text features, image features

1. Introduction

Social media platforms such as Facebook, Twitter, and Instagram allow their users to easily connect and share information. The unprecedented data generated by millions of users from all around the world make social media ideal places of finding what is happening in the wider world beyond direct personal experience. As a microblog site, Twitter enables its users to post

instantly what is happening in their location in 140-character messages, or tweets. Twitter is an information system that provides a real-time reflection of its users. As a consequence, Twitter serves as a rich source for exploring what is attracting users' attention and what is happening around the world. For example, for news and communications in time of a disaster, social media users use Twitter to tweet and post text, images, and video through their smartphones and tablets. As a result, Twitter becomes a good source for detection of events such as disasters [1].

An event is the basis on which people form and recall memories. Events are a natural way to refer to any observable occurrence that groups persons, places, times, and activities together. They are useful because they help us make sense of the world around us, helping to recollect real-world experiences, explaining phenomena that we observe, or assisting us in predicting future events. Social events are the events that are attended by people and are represented by multimedia content shared online. Instances of such events are concerts, disasters, sports events, public celebrations, or protests. Twitter platform forms a rich site for news, events, and information mining. It allows the posting of images and videos to accompany tweets produced by users of the site. As a result, the site contains multimedia content which can be mined using complicated algorithms. However, due to the huge burst in information, event detection in Twitter is a complicated task that requires a lot of skill and expertise in data mining. Here, an event detection is a data mining task aiming to identify the event in a media collection. To enhance the process of event detection, an automatic algorithm needs be developed to mine multimedia information.

Many approaches have been proposed for event detection [2–4]. For event detection using Twitter data, there are different ways to detect event, including using part of speech technique [5], hidden Markov model (HMM) [6], and term frequency and inverse document frequency (TF-IDF), and part-of-speech (POS) tagging and parsing. Alqhtani et al. [7] introduced a data fusion approach in multimedia data for earthquake detection in Twitter by using kernel fusion. It had achieved a high detection accuracy of 0.94, comparing to accuracy of 0.89 with texts only, and accuracy of 0.83 with images only. Sakaki et al. [8] showed that mining of relevant tweets can be used to detect earthquake events and predict the earthquake center in real time by using TF-IDF. In the process of event detection, the method utilized TF-IDF to eliminate redundant information or keywords. It provided a way of real-time interaction for earthquakes in Twitter. It developed a classifier based on several features including keywords, the number of words and the context, location and time of the words. It used a probabilistic spatiotemporal model to detect the location of the earthquake happened in Japan. Yardi and Boyd [9] used keyword search to present the role of stream news in spreading local information from Twitter for two accidents including a shooting and a building collapse. Ozdikis et al. [10] discussed an event detection method for various topics in Twitter using semantic similarities between hashtags based on clustering. Zhang et al. [11] proposed an event detection from online microblogging stream. It combined the normalized term frequency and user's social relation to weight words. Although many approaches have been proposed for event detection using Twitter data, most of them used no images but only textual analysis of tweet texts. With the cases of using images, restrictions had been applied. For example, Nguyen et al. [12] used textual features and image features for event detection. However, they focused on the principle

that no one user could be in multiple events at the given time, demanding that the image was separated by user at the beginning.

This chapter introduces a novel algorithm to detect a major event such as a wildfire through mining social media Twitter. In developing an efficient event detection algorithm, our considerations are: For Twitter users, it is much easier than ever before to post about natural disaster like wildfire, by posting different kinds of multimedia like pictures, rather than just typing a message. Using both image and text can improve disaster management than using image only or text only. Furthermore, Twitter has been used as a source for obtaining information about wildfires, specifically when landlines and mobile phone lines are damaged. Therefore, we propose to use visual information as well as textual information to improve the performance of automatic even detection. The algorithm starts with monitoring a Twitter stream to pick up tweets having texts and images. Secondly, it preprocesses the Twitter data to eliminate unwanted data and transform unstructured data into structured data. Thirdly, it extracts features from the text and image. Fourthly, a multiple kernel learning is applied to the features to fuse the multimedia data. Finally, a decision on event detection is made.

The chapter is organized as follows. After this section, Section 2 describes the proposed event detection method, which consists of Twitter data collection, data preprocessing, feature extraction, multiple kernel learning fusion, and event classification. Section 3 gives experiment design, results, and discussion. Section 4 presents conclusion.

2. The proposed algorithm

The proposed automatic event detection method includes five steps, including Twitter data collection, data preprocessing, features extraction, multimedia data fusion, and final event detection. The block diagram of the proposed method is shown in **Figure 1**. The following subsections explain the details of these five steps of the proposed algorithm.

2.1. Twitter data collection

Data about specific events have been obtained through the use of a Twitter application program or through Twitter partner sites. This study characterizes public responses on Twitter for different kinds of events such as storm, earthquake, wildfire, terror attacks, and other events. Two recent extreme events which happened in the last 4 years were used as case studies, including Brisbane hailstorm and California wildfire.
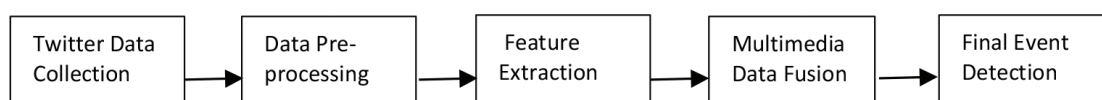
Twitter Data Collection → Data Pre-processing → Feature Extraction → Multimedia Data Fusion → Final Event Detection

Figure 1. The block diagram of the proposed method of event detection from Twitter.

2.1.1. Brisbane hailstorm 2014

The Brisbane hailstorm occurred in Brisbane, Australia on November 27, 2014. It was the worst hailstorm in a decade, causing injury to about 40 people and costing around 1.1 billion Australian dollars. The data about this hailstorm were collected between November 27, 2014 and November 28, 2014 and contained both texts and images. The dataset contained 280,000 tweets. **Figure 2** presents an example of the twitters (left column) and the word cloud for the data (right column). A word cloud is an image consisting of the words used in the data, where the size of each word indicates its occurrent frequency.

2.1.2. California wildfires 2017

The 2017 wildfire season in California started in April and extended to December. 1,381,405 acres were burned and the economic cost was over 13.028 billion American dollars. The data for this event were collected for 5 days in July 2017. It contained 600,000 tweets with some tweets consisting of both text and images.

2.2. Data preprocessing

The goal of data preprocessing is to discover important features from collected raw data. Preprocessing is a set of techniques used prior to analysis to remove imperfection, inconsistency, and redundancy. In this study, there was a high need to preprocess text data, because many tweets were not properly formatted or contained spelling errors. As a result, using a filter, cleaning is done before the text data are further handled. For image data in Twitter, we extracted the image's hyperlink and removed a tweet if its hyperlink was empty or did not work, since in this study, the tweet must contain both image and text. After preprocessing, the data will be ready for feature extraction.

Figure 2. An example of tweets on Brisbane hailstorm (left) and the word cloud for the event (right).

2.3. Feature extraction

In event detection, a set of features is required. A feature vector is a set of features used to reduce the dimensionality of the data, especially in the case of large volume data. Feature extraction involves reducing the amount of resources required to describe a large set of data accurately. Two approaches to feature extraction were employed for different data sets: content-based and description-based. Content-based feature extraction is based on the content of an object, whereas description-based extraction relies on metadata such as keywords. In this study, content-based features were used for images, and description-based features were used for texts.

2.3.1. Textual features

In extracting textual features, two major processes are executed, including filtering and feature calculation. The filtering will derive the key information out of tweets. The feature calculation will represent the significance of a word within a given document using a measurement named term frequency-inverse document frequency (TF-IDF) [13].

The filtering consists of five major steps including: filtering tweets in such way that they are in English only; converting all words to lowercase; converting the string to a list of tokens based on whitespace; removing punctuation marks from the text; eliminating common words that do not tell anything about the dataset (such as the, and, for, etc.); and reducing each word to its stem by removing any prefixes or suffixes.

After the filtering, TF-IDF is calculated, which is a statistical measure that details the significance of a word within tweets based on how often the word occurs in an individual tweet compared with how often it occurs in other tweets [14]. The advantage of using the TF-IDF algorithm technique is that it allows the retrieval of information since the TF-IDF values increase proportionally with the number of times a certain keyword appears in a document, being offset by the frequency of the word in the database. The TF-IDF algorithm utilizes a combination of term frequency and inverse document frequency.

Suppose there is a vocabulary of k words, then each document is represented by a k-vector $V_d = (t_1, \ldots, t_i, \ldots, t_k)^T$ of weighted word frequencies with components t_i. TF-IDF is computed as follows:

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \ t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \tag{1}$$

where n_{id} is the number of occurrences of word i in document d, n_d is the total number of words in document d, n_i is the number of occurrences of term i in the database, and N is the total number of documents in the database. It can be seen that TF-IDF is a product of the word frequency $\left(\frac{n_{id}}{n_d} \right)$ and the inverse document frequency $\left(\log \frac{N}{n_i} \right)$. For a word i, the more it occurs in document d (i.e., the higher the n_{id} is), the bigger the t_i is, meaning the word i is more significant. Note here, the significance of the word i in document d is offset by the frequency

of the word in the whole database. This offsetting will result in different t_i for word i that are unevenly distributed among the documents.

2.3.2. Visual features

In calculating visual features, each image is represented with a visual-word vector consisting of visual words. A visual word is a cluster in an image that represents a specific pattern shared by keypoints in that cluster. A keypoint in an image is a section of the image that is highly distinctive, allowing its correct match in a large database of features to be found. A keypoint is detected based on various image features. In this study, four types of features are used to detect a keypoint, including histogram of oriented gradients (HOG) [15], gray-level co-occurrence matrix (GLCM) [16], color histogram (CH) [17], and scale-invariant feature transform (SIFT) [14].

HOG is a feature descriptor that is calculated by counting occurrences of gradient orientation in localized portions of an image. Operating on local cells, HOG is invariant to geometric and photometric transformations, but for object orientation.

GLCM is got by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image. It is used to describe texture such as a land surface. It can provide useful information about the texture of an object but not information about the shape or size.

CH is defined as the distribution of colors in an image. It represents the actual number of pixels of a certain color in each of a fixed list of color ranges. A major drawback of a color histogram is that it does not take into account the size and shape of object.

SIFT is an algorithm to detect and describe local features in images. It produces an image descriptor for image-based matching and recognition. It mainly detects interest points from a gray image, at which statistics of local gradient directions of image intensities are accumulated to give a summarizing description of the local image structures around each interest point. The descriptor is used for matching corresponding interest points between different images.

In calculating visual word, the four types of features are firstly calculated for an image. Then, keypoints are derived based on these features. Thirdly, K-means clustering algorithm is used to cluster the keypoints into a large number of clusters. Each cluster is then considered as a visual word that represents a specific pattern. In this way, the clustering process generates a visual-word vocabulary describing different patterns in the images. The number of clusters determines the size of the vocabulary.

2.4. Multimedia data fusion

Starting from an introduction of multimedia data fusion, this section discusses the principle of kernel-based data fusion, then presents the details of the proposed multiple kernel learning for data fusion, and finally gives the details of final event detection.

2.4.1. About multimedia data fusion

Multimedia data fusion is the process in which different features of multimedia are brought together for the purpose of analyzing specific media data. Some common multimedia analyses that enable understanding of multimodal data include event detection, human tracking, audio-visual speaker detection, and semantic concept detection. The purpose of data fusion is to ensure that the algorithm of a process is improved. Through the use of a fusion strategy, the multimedia analysis can improve the accuracy of the output, resulting in more reliable decision-making.

There are many fusion methods such as linear fusion, linear weighted fusion, nonlinear fusion, and nonlinear weighted fusion. This study relates to a fusion strategy of combining both textual and visual modalities in the context of event detection. A new method of multimedia fusion has been proposed. It is based on multiple kernel learning (MKL). It has the advantage of incorporating with classifier learning and handling a big volume of data.

2.4.2. Kernel-based data fusion

Kernel methods are based on a kernel function, which is a similarity function that finds similarities over pairs of data points. The kernel function enables the kernel method to operate in a high-dimensional space by simply applying an inner product. The kernel method introduces nonlinearity into the decision parameters by simply mapping the original features of the original sources onto a higher dimensional space. For kernel function $\kappa(x, y)$ and mapping function $\phi : \mathcal{X} \to \mathcal{F}$, the model built by the kernel method can be expressed as an inner product in the following equation:

$$\kappa(x,y) = \langle \phi(x) \cdot \phi(y) \rangle \tag{2}$$

where $\kappa(x, y)$ is positive semidefinite and $\phi : \mathcal{X} \to \mathcal{F}$ maps each instance x, y into feature space $\mathcal{F}$, which is a Hilbert space. With the kernel method, a simple mining technique such as classification can be applied further to analyze the data.

Kernel methods can be described as a class of algorithms for pattern analysis, whose best member is the support vector machine [18]. There are many kernel methods including polynomial, fisher, radial basis functions (RBF), string, and graph kernels. Several commonly used kernel functions are:

$$\text{Linear function}: \kappa(x_i, x) = x_i \cdot x \tag{3}$$

$$\text{Polynomial function}: \kappa(x_i, x) = [(x_i \cdot x) + 1]^P \tag{4}$$

$$\text{Radial basis function (RBF)}: \kappa(x_i, x) = e^{-\|x_i - x\|^2 / 2\sigma^2} \tag{5}$$

where x_i and x are two samples represented as feature vectors, $\|x - x'\|$ is the distance between the two feature vectors, σ is a free parameter, and p is a constant.

Studies show that nonlinear kernels, for example, string kernel or RBF, have a significantly higher level of accuracy for multimedia data compared to linear classification models [19]. Kernel-based data fusion, denoted as kernel fusion, has been pioneered by Lanckriet et al. [20] as a statistical learning framework for genomic data fusion and has been applied widely in various applications. In particular, kernel representation resolves the heterogeneities of data sources by transforming different data structures into kernel matrices.

2.4.3. Multiple kernel learning for fusion

When dealing with multimedia input, Kernel-based data fusion can be applied so that it merges all the features from different sources into a concatenated vector before achieving classification. However, it is hard to combine features into one representation without facing the problem of dimensionality [21]. Multiple kernel learning (MKL) is one of the most popular fusion technologies (Lan et al.), which allows us to combine possibly heterogeneous data sources, making use of the reduction of heterogeneous data to the common framework of kernel matrices. The reduction of heterogeneous data is achieved by using a kernel for each type of feature rather than using one kernel for all the features. For a set of base kernels κ_l, the optimal kernel combination is calculated as:

$$\kappa_{optimal} = \sum_l \beta_l \kappa_l \tag{6}$$

where β_l is the weight for each base kernel κ_l.

Multiple kernel learning is flexible for multimodal data, since each set of data features is assigned a different notion of similarity, i.e., a different kernel. Instead of building a specialized kernel for the applications with multimodal data, it is possible to define a kernel for each of these data and linearly combine these kernels [22]. Multiple kernel learning presents the solution of the optimal combination of the kernels. In this study, semi-infinite programming [23] is used to achieve robustly and automatically optimizing the kernel weights. It solves the MKL in two steps: the first step is the initialization of the problem with a small number of linear constraints and the second step is to solve the parameters.

In event detection, the MKL framework defines a new kernel function as a linear combination of l base kernels:

$$\kappa(x_i, x) = \sum_l \beta_l \kappa_l(x_i, x) \tag{7}$$

where each base kernel κ_l is selected for one specific feature, the nonnegative coefficient β_l represents the weight of the lth base kernel in the combination, and $\sum_{l=1} \beta_l = 1$.

A kernel is utilized for each of the features followed by a combination of multiple features as indicated in Eq. (7). To select the spread parameter σ for each kernel, a cross-validation is performed with grid search for the range 0.001–0.01. Such selection is suitable for our data, resulting in the best classification accuracy without need for long time processing. The cross-validation is a model evaluation method that is applied during the training phase to find

unknown parameters. To find the best kernel for image features and text features, cross-validation is applied. The best kernel means the best σ of the RBF kernel. The final kernel is the weighted sum of each feature kernel, with each feature kernel having its optimal σ.

MKL is coupled with classifier learning, such as support vector machine (SVM) [24] in our method, enhancing mutually interpretability of results. Support vector machine is formalized as solving an optimization problem. In the process, it finds the best hyperplane separating relevant and irrelevant vectors by maximizing the size of the margin between the two sets. By using a kernel, it can find the maximum-margin hyperplane in a transformed space.

For a given set of n training examples, $\left\{\left(x_i, y_j\right)_{i=1}^n, x_i \in \mathbb{R}^d \text{ and } y_i \in \{+1, -1\}\right\}$, where x_i is a training example and y_i is the corresponding class label. The nonlinear support vector machine maps a training example x_i in the input space to a higher dimensional space $\phi(x_i)$ using a nonlinear mapping function ϕ. It constructs an optimal hyperplane, defined by Eq. (8), to separate the two classes.

$$w^T \phi(x) + b = 0 \tag{8}$$

where $b \in R$, w is a normal vector. The hyperplane constructed in kernel feature space is a maximum-margin hyperplane, one which maximizes the margin between the two datasets. This is achieved by solving the primal SVM problem:

$$\min\left(\frac{1}{2}\|w\|^2 + C\sum_i \xi_i^2\right) \quad \text{subject to}$$
$$y_i\left(w^T\phi(x_i) + b\right) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n \tag{9}$$
$$\xi_i \geq 0 \quad i = 1, 2, \ldots, n$$

where ξ_i are nonnegative slack variables and C is a regularization parameter that determines the trade-off between the margin and the error in training data. The minimizing operation is against parameters w, b, and ξ_i. The corresponding SVM dual problem for the primal problem described in Eq. (9) is its Lagrangian defined as:

$$\max\left(\sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \alpha_i\alpha_j y_i y_j\left(k(x_i, x_j) + \frac{1}{C}\boxtimes_{ij}\right)\right) \quad \text{subject to}$$
$$\sum_{i=1}^n y_i\alpha_i = 0 \tag{10}$$
$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n$$

where δ_{ij} is the Kronecker δ defined to be 1 if $i = j$ and 0.

The dual problem is a keypoint for deriving SVM algorithms and studying their convergence properties. The function $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel function and α_j are the Lagrange coefficients. The Karush-Kuhn-Tucker (KKT) conditions are necessary conditions for the solution to the optimal parameters when there are one or more inequality constraints. Here, the KKT conditions for Eq. (10) are also sufficient for optimality since Eq. (10) meets the following three

conditions: the object function is concave, the inequality constraint is a continuously differentiable convex function, and the equality constraint is an affine function. According to the KKT conditions, the optimal parameters α^*, w^*, and b^* must satisfy:

$$\alpha_i^* \left[y_i \left(\sum_{j=1}^{n} \alpha_j^* y_j k(x_i, x_j) + b^* \right) - 1 + \xi_i \right] = 0, \quad i = 1, 2, \ldots, n \tag{11}$$

In classification, only a small subset of the Lagrange multipliers α_i^* tend to be nonzero usually. The training examples with nonzero α_i^* are defined as support vectors. They construct the optimal separating hyperplane as:

$$\mathbf{w}^{*T} \phi(x) + b^* = \sum_{j=1}^{n} \alpha_j^* y_j k(x, x_j) + b^* = 0 \tag{12}$$

In SVM framework, the task of multiple kernel learning is considered as a way of optimizing the kernel weights at the same time of training SVM. For multiple kernels, Eq. (12) can be converted into the following equation to derive the dual form for MKL.

$$\max \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \sum_{l=1}^{m} \beta_l k_l(x_i, x_j) \right) \text{ subject to}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{13}$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n$$

$$\beta_l \geq 0, \quad \sum_{l=1}^{m} \beta_l = 1, \quad l = 1, 2, \ldots, m$$

In Eq. (13), both the base kernel weights β_l and the Lagrange coefficients α_j need to be optimized. A two-step procedure is considered to decompose the problem into two optimization problems.

In the first step, through grid search and cross-validation, the best weights β_l are derived by minimizing the 2-norm soft margin error function using linear programming. The weights for text features and image features are changed according to the type of data. For example, for wildfire data, the weight for text features was chosen as 0.70, and the weight for image features was chosen as 0.30. In the second step, the Lagrange coefficients α_j are obtained by maximizing Eq. (13) using quadratic programming. The interior point method is used to solve quadratic programming in the proposed method, which achieves optimization by traversing the convex interior of the feasible region.

2.5. Final event detection

As described above, the training process of multimedia data fusion builds the system by deriving parameters α_j, b, x_i, β_l, and k_l. For a test input x, the decision function for MKL, i.e., the event detection function $F(x)$, is a convex combination of basis kernels, computed as:

$$F(x) = \text{sign}\left(\sum_i \sum_l \beta_l(k_l(x_i, x).\alpha_i + b)\right) \tag{14}$$

where x_i are support vectors, α_i denote Lagrange multipliers corresponding to support vectors, and b is a bias which intercepts the hyperplane that separates the two groups in the normalized data space.

Depending on the sign of Eq. (14), the Twitter data are divided into two groups. The first group contains twitters of a positive class, meaning the event has happened. The second group contains twitters of a negative class, meaning the event has not happened. Both classes are based on image and text features which are extracted from the same tweet.

3. Experiment design, result, and discussion

3.1. Experiment design

Experiments have been done to build the event detection method and test its performance on real twitters. The algorithm is implemented in Matlab. In the experiments, the tweets that contain both text and image are collected from the Twitter streams. The data collection is for two events: Brisbane hailstorm and California wildfire.

The data are separated into two sets, including training and testing. Training data are divided into two groups: the event has happened or the event has not happened, which are manually labeled. Each group has the same number of tweets. The same process is applied to the testing data. The numbers of samples for the two sets are the same. The reasons to have the same number of samples are: the greater the size of the training set and testing sets, the better the algorithm is trained and tested, and the total number of samples is big enough to split the data into two equal sets. For each tweet set to be used for detecting whether an event has happened or not, its features are extracted for fusing operation.

In order to validate the performance of the proposed MKL event detection using both text and image, two other methods are also built and tested. Both the other two methods are based on single kernel learning, with one method taking text only as input and the other taking image only as input.

3.2. Performance evaluation parameters

In order to measure the performance of the proposed method and those of other comparing methods more objectively and comprehensively, four performance parameters are used, including accuracy (A), precision, recall, and F-score [25]. They are defined below.

The accuracy for the event detection method is defined as

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

where *TP*, *TN*, *FP*, and *FN* represent true positive, true negative, false positive, and false negative, respectively. In classifying an event such as a wildfire, a true positive (*TP*) is considered to be when a wildfire happened and a tweet from the wildfire data is classified as wildfire. If a tweet from the wildfire data is classified as not wildfire, this is a false negative (*FN*). In contrast, when a tweet from the data about a nonwildfire event is classified as wildfire, that is a false positive (*FP*). If a tweet from the data about a nonwildfire event is classified as not wildfire, that is a true negative (*TN*). For other events such as hailstorm, the classification is applied in the same way.

Precision is a term that refers to the fraction of correctly retrieved tweets. It is a function of true positives and false positives. It is defined as:

$$precision = \frac{TP}{TP + FP} \tag{16}$$

The term recall refers to the fraction of relevant tweets that were retrieved. It is a function of correctly classified examples, i.e., true positives, and the false negatives true positive rate. It is defined as:

$$recall = \frac{TP}{TP + FN} \tag{17}$$

F-score is introduced as the harmonic mean of precision and recall, in this way combining and balancing precision and recall. It is defined as:

$$F - score = 2 * \frac{precision * recall}{precision + recall} \tag{18}$$

F-score measures how well a learning algorithm applies to a class. It is based on the weighted average of precision and recall.

3.3. Result and discussion

In order to validate the performance of the proposed event detection based on multiple kernel learning, two other single kernel-based methods are also built and tested. Both of the other two methods take single media as input, i.e., text or image. The performance metrics of the proposed method and that of the other two methods for two events are given in **Table 1**.

From the table, it can be seen that for both the Brisbane hailstorm event and California wildfire event, the proposed method consistently achieved a better performance in all the four metrics than the methods using text only or image only. For example, the proposed method achieved an accuracy of 0.93 for Brisbane hailstorm, whereas the method of using text only achieved 0.89 and the method of using image only achieved 0.85. For California wildfire, the accuracy of the proposed method is 0.92, better than that of 0.90 and 0.86 of the other two methods. Comparing to the other two single kernel-based methods, it can also be seen that the proposed method has improved about 5%, 6%, 5%, and 6%, respectively, in accuracy, precision, recall,

Event	Data	Accuracy	Precision	Recall	F-score
Brisbane hailstorm	Text only	0.89463	0.90662	0.90171	0.90416
	Image only	0.85981	0.82759	0.90566	0.86486
	The proposed method	0.93434	0.93578	0.94444	0.94009
California wildfire	Text only	0.90981	0.91533	0.91116	0.91324
	Image only	0.86406	0.88971	0.84912	0.86894
	The proposed method	0.92736	0.9311	0.93721	0.93414

Table 1. Event detection performance of the proposed method in comparison with the performance of two methods that use text only or image only.

and *F*-score. The experiment results have proven that event detection from multimedia data in Twitter is enhanced and improved by using a combination of multiple features for both images and text.

4. Conclusion

In this chapter, a method for detecting hot events, in particular disasters such as hailstorm and wildfires, is proposed. The approach uses visual information as well as textual information to improve the performance of detection. It starts with monitoring a Twitter stream to pick up tweets having texts and images, and storing them in a database. After that, Twitter data is preprocessed to eliminate unwanted data and transform unstructured data into structured data. Then, features in both texts and images are extracted for event detection. For feature extraction from the text, the term frequency-inverse document frequency technique is used. For images, the features extracted are: histogram of oriented gradients descriptors for object detection, gray-level co-occurrence matrix for texture description, color histogram, and scale-invariant features transform. In the next step, text features and image features are input to the multiple kernel learning (MKL) for fusion. MKL can automatically combine both feature types in order to achieve the best performance. The proposed method was tested on two datasets from two events, including Brisbane hailstorm 2014 and California wildfires 2017. The method is compared with a method that used text only and another method that used images only. With the Brisbane hailstorm data, the proposed method achieved the best performance, with a fusion accuracy of 0.93, compared to 0.89 with text only, and 0.85 with images only. With the California wildfires data, the proposed method achieved the best performance, with a fusion accuracy of 0.92, compared to 0.90 with text only, and 0.86 with images only. It has demonstrated that event detection from multimedia data in Twitter is enhanced and improved by our approach of using a combination of multiple features for both images and text. The proposed method also improves computational efficiency when handling big volumes of data, and gives better performance than other fusion approaches. It has delivered an accurate and effective detection method for detecting events, which can be used for spreading awareness and organizing responses.

The research presents a breakthrough in terms of risk management strategies, one that can improve public health preparedness and lead to better disaster management actions.

Author details

Suhuai Luo[1]*, Samar M. Alqhtani[2] and Jiaming Li[3]

*Address all correspondence to: suhuai.luo@newcastle.edu.au

1 School of Electrical Engineering and Computing, The University of Newcastle, Callaghan NSW, Australia

2 College of Computer Science and Information Systems, Najran University, Najran, Saudi Arabia

3 Quantitative Imaging Research Team, CSIRO Data61, NSW, Australia

References

[1] Computing Community Consortium. Computing for Disasters: A Report from the Community Workshop, Visioning Workshop on Computing Disaster Management, Washington, DC; 2012

[2] Scherp A, Jain R, Kankanhalli M, Mezaris V. Modeling, detecting, and processing events in multimedia. In: Proceedings of the 18th ACM International Conference on Multimedia; 2010. pp. 1739-1740

[3] Petkos G, Papadopoulos S, Mezaris V, Troncy R, Cimiano P, Reuter T, et al. Social event detection at MediaEval: A three-year retrospect of tasks and results. In: Proceedings of the ICMR 2014 Workshop on Social Events in Web Multimedia (SEWM); 2014. pp. 27-34

[4] Xu Z, Liu Y, Yen N, Mei L, Luo X, Wei X, et al. Crowdsourcing based description of urban emergency events using social media big data. IEEE Transactions on Cloud Computing; 2016;**99**:1-1, DOI: 10.1109/TCC.2016.2517638

[5] Kirsch S. Sustainable mining. Dialectical Anthropology. 2010;**34**:87-93

[6] Ting IH. Social Network Mining, Analysis, and Research Trends: A Phenomenal Analysis. Boston, MA: Cengage Learning; 2012

[7] Alqhtani S, Luo S, Regan B. A multiple kernel learning based fusion for earthquake detection from multimedia Twitter data. In: Multimedia Tools and Applications - An International Journal. June 16, 2017. ISSN: 1380-7501 (Print) 1573-7721 (Online), DOI: 10.1007/s11042-017-4901-9

[8]   Sakaki T, Okazaki M, Matsuo Y. Earthquake shakes Twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web; 2010; pp. 851-860

[9]   Yardi S, Boyd D. Tweeting from the Town Square: Measuring geographic local networks. In: Proceedings of the ICWSM; 2010. pp. 194-201

[10] Ozdikis O, Senkul P, Oguztuzun H. Semantic expansion of hashtags for enhanced event detection in Twitter. In: Proceedings of the 1st International Workshop on Online Social Systems; 2012

[11] Zhang X, Chen X, Chen Y, Wang S, Li Z, Xia J. Event detection and popularity prediction in microblogging. Neurocomputing. 2015;**149**:1469-1480

[12] Nguyen T V, DaoMS, Mattivi R, Sansone E, De Natale F G, Boato G. Event Clustering and Classification from Social Media:Watershed-Based and Kernel Methods. Editors: M. Larson, et al. Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain, October 18-19, 2013, volume 1043 of CEUR Workshop Proceedings; 2013

[13] Wu H, Luk R, Wong K, Kwok K. Interpreting TF-IDF term weights as making relevance decisions. ACM Transactions on Information Systems. 2008;**26**(3):13-49

[14] Wang Z, Shawe-Taylor J. A kernel regression framework for SMT. Machine Translation. 2010;**24**(2):87-102

[15] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proceedings of Computer Vision and Pattern Recognition; 2005

[16] Mohanaiah P et al. Image texture feature extraction using GLCM approach. International Journal of Scientific and Research Publications. 2013;**3**(5):1-5. ISSN 2250-3153

[17] Zhang D, Islam MM, Lu G. A review on automatic image annotation techniques. Pattern Recognition. 2012;**45**(1):346-362

[18] Pelillo M. Similarity-Based Pattern Analysis and Recognition. Berlin, Germany: Springer Science & Business Media; 2013

[19] Borra S, Rocci R, Vichi M, Schader M. Advances in Classification and Data Analysis. Berlin, Germany: Springer Science & Business Media; 2012

[20] Lanckriet GR, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI. Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research. 2004;**5**:27-72

[21] Snoek CG, Worring M, Smeulders AW. Early versus late fusion in semantic video analysis. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM; 2005. pp. 399-402

[22] Lan Z, Bao L, Yu S, Liu W, Hauptmann AG. Multimedia classification and event detection using double fusion. Multimedia Tools and Applications. 2014;**71**(1):333-347

[23] Sonnenburg S, Rätsch G, Schäfer C, Schölkopf B. Large scale multiple kernel learning. Journal of Machine Learning Research. 2006;**7**:1531-1565

[24] Cristianini N, Shawer-Tatlor J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, United Kingdom: Cambridge University Press; 2005. ISBN 0521780195

[25] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing and Management. 2009;**45**(4):427-437