RESEARCH

Open Access

Real-time event detection in social media streams through semantic analysis of noisy terms

Taiwo Kolajo[1,2*], Olawande Daramola[1] and Ayodele A. Adebiyi[3]

*Correspondence: kolajot@cput.ac.za; taiwo.kolajo@fulokoja.edu.ng

[1] Department of Information Technology, Cape Peninsula University of Technology, Cape Town, South Africa
[2] Present Address: Department of Computer Science, Federal University Lokoja, Lokoja, Kogi State, Nigeria
[3] Department of Computer Science, Landmark University, Omu-Aran, Kwara State, Nigeria

Abstract

Interactions via social media platforms have made it possible for anyone, irrespective of physical location, to gain access to quick information on events taking place all over the globe. However, the semantic processing of social media data is complicated due to challenges such as language complexity, unstructured data, and ambiguity. In this paper, we proposed the Social Media Analysis Framework for Event Detection (SMAFED). SMAFED aims to facilitate improved semantic analysis of noisy terms in social media streams, improved representation/embedding of social media stream content, and improved summarization of event clusters in social media streams. For this, we employed key concepts such as integrated knowledge base, resolving ambiguity, semantic representation of social media streams, and Semantic Histogram-based Incremental Clustering based on semantic relatedness. Two evaluation experiments were conducted to validate the approach. First, we evaluated the impact of the data enrichment layer of SMAFED. We found that SMAFED outperformed other pre-processing frameworks with a lower loss function of 0.15 on the first dataset and 0.05 on the second dataset. Second, we determined the accuracy of SMAFED at detecting events from social media streams. The result of this second experiment showed that SMAFED outperformed existing event detection approaches with better Precision (0.922), Recall (0.793), and F-Measure (0.853) metric scores. The findings of the study present SMAFED as a more efficient approach to event detection in social media.

Keywords: Event detection, Event summarization, Semantic analysis, Social media stream, Word sense disambiguation

Introduction

Event detection is a computational operation that enables the automatic identification of significant incidents by analysing social media data. An event refers to a significant incident happening in a place and time [1].

Event detection in social media streams is a worthwhile research area because it provides the opportunity to know about current happenings and people's views at the click of a button. Gone are the days when persons can 'kill' the news that they do not want other persons to know about by suppressing it through the news agency or organization. This is no longer possible with the existence of various social media platforms. Once a piece of interesting news gets into social media, it travels faster. It reaches a wider audience who would continue to spread the news until a desirable action is taken. Thus, researching this area is important to objectively reveal current events and happenings, such as breaking news, instant outbreaks, infectious disease, and terror attacks [2].

We are in the era of social media, where there is abundant data and opportunities to be exploited. Social media enable us to take advantage of the social nature of human association, making it possible for individuals to express their feelings, become part of a virtual network and collaborate remotely [3].

A social media stream is made up of user-generated content. As such, ambiguity, which is part of the core problems of natural language text, also exists in the content of social media streams. Ambiguity occurs when a word can be expressed in at least two ways or senses in a determined context [4, 5]. Ambiguity may not present any difficulty to a human being (i.e., native speakers) because ambiguity can be resolved using the knowledge of the context and common sense. But disambiguating text efficiently with a computer application is still a problem [6].

Social media streams are characterized by short messages; utilization of progressively advancing sporadic, casual, abridged words, syntactic and spelling blunders; blended dialects; vagueness; and inappropriate sentence structure [7]. These characteristics make it difficult for methods that rely on them for computational purposes to perform adequately and effectively [7–12]. Likewise, many of the current methodologies for event detection have mostly paid attention to the use of trivial keywords or themes retrieval. Still, they have not fully considered the valuable semantics embedded in social media streams, which hinders the accuracy of event detection [13, 14]. Particularly of interest to this paper is the problem of ambiguity that stems from the use of slangs, abbreviations, and acronyms (SAB) in social media streams which makes the interpretation of such terms complicated during the event detection process. Most of the existing event detection methods have not considered the semantic analysis of noisy terms in the form of SAB and associated ambiguities in the design of their solutions.

Social media is a suitable medium for reporting serious events and emergencies. However, it presents many challenging issues that make it difficult to effectively uncover interesting and useful messages. Event summarization in the context of this paper can be referred to as finding a tweet representative that can suitably represent an event cluster. The noisy characteristics of social media content require new semantic innovations that will facilitate accurate analysis of social media streams. Thus, improving the precision of event detection techniques by resolving the noisy attributes of social media content is necessary.

This paper proposes a Social Media Analysis Framework for Event Detection (SMAFED) that resolves the noisy and ambiguous terms in social media streams to improve event detection accuracy. SMAFED was realized by integrating a local vocabulary consisting of slangs, acronyms, abbreviations; and incremental semantic clustering. This is to facilitate the understanding of implicit semantics embedded in social media streams to improve event detection. An evaluation was done by benchmarking SMAFED with existing approaches, including locality sensitive hashing [15], cluster summarisation [16], entity-based approach [17], and Repp framework [18]. The Precision, Recall, and F-measure metrics were used to assess the accuracy of event detection. To further demonstrate the plausibility of SMAFED in other research domains, the pre-processing and enrichment components of SMAFED were benchmarked with other pre-processing frameworks to extract sentiments from tweets using a generalized dataset, Twitter sentiment analysis training corpus, and a dataset of Nigerian origin called Naija-tweets.

The contributions of this paper are as follows:

1. Existing event detection methods have focused mostly on filtering out slangs, abbreviations, and acronyms (SAB); removing noisy terms including SAB, or ignoring them entirely during the pre-processing stage of social media streams. They did not perform semantic analysis of noisy terms like SAB to determine their contextual meanings and their impact on the accuracy of results. Semantic analysis of SAB terms was done for the first time in this study which yielded improved results.
2. In contrast to existing approaches, the proposed framework (SMAFED) introduces the data enrichment layer that enables the semantic analysis of SAB and ambiguity issues associated with their usage.
3. A dedicated algorithm for the disambiguation of slangs, acronyms, and abbreviations (SABDA) was proposed and used to disambiguate ambiguous SAB terms to better understand and interpret noisy terms in social media streams.
4. An integrated knowledge base (IKB) representing a local vocabulary of SAB terms was created to facilitate semantic analysis of noisy terms in social media streams. The IKB is a valuable and reusable resource that can support other computational operations on SAB.

The remaining part of this paper is organized as follows. Related work presents the related work, where an overview of the previous approaches that are relevant to event detection was presented. Methodology discussed the proposed Social Media Analysis Framework for Event Detection (SMAFED) for improved event detection in social media streams. Evaluation experiment presents the report of the evaluation of the SMAFED. In doing this, the impact of using the data enrichment layer to aid the results of SMAFED, and the performance of SMAFED when used for event detection from social media streams were discussed. The paper is concluded in Conclusion and further work with a summary and an overview of future work.

Related work

This section presents an overview of relevant previous research efforts on event detection from social media streams. The aspects covered include unsupervised learning, semi-supervised learning, supervised learning, and semantic-based approaches for event detection in social media streams.

Unsupervised learning for event detection in social media stream

Unsupervised learning is a type of learning that draws inductions from an unlabeled dataset [19]. Due to several iterations required to compute similarity or dissimilarity in the observed dataset, all of the datasets ought to be accessible in memory before running the algorithm in most cases. However, with data stream clustering, the challenge is searching for a new structure in the data as it evolves, characterizing the streaming data in clusters to leverage them to report events in the data stream. The clusters are then ordered based on the scoring function [1]. Some studies on event detection based on unsupervised learning are presented next.

Authors in [15] worked on streaming first story detection with an application to Twitter. The authors used a hash function called Locality Sensitive Hashing (LSH) to place similar documents in the same bucket. Shannon entropy was used to measure the information contained in the cluster. Clusters were ranked based on the value of the entropy. Event detection in Twitter was carried out by [20]. The paper focused on detecting real-life events from tweets using Event Detection with Clustering of Wavelet-based Signals (EDCoW). Authors in [21] employed Term Frequency (TF) and Kullback–Leibler divergence (KLD) to propose real-time summarization of scheduled events from Twitter streams. The work addressed summarization of tweet content to provide the user with summed upstream describing the key sub-events by employing a two-step process: sub-event detection using an outlier-based sub-event detection technique and selection of tweets related to the sub-event detected to provide a summary. For the summary, TF and KLD techniques were compared and found out that KLD performed better. Authors in [16] proposed a framework to detect events in social streams using similarity score and cluster summarisation techniques. Authors used content- and network-stream-based clustering for event detection. Mining Spatio-temporal information on microblogging streams using a density-based online clustering method was proposed by [22]. The paper investigated the extraction of spatio-temporal features of social media streams by employing an Incremental Density-based Spatial Clustering Application with Noise (DBSCAN) algorithm to enhance event awareness. A weighting factor called BursT, a sliding window technique to address concept drift, was employed. However, none of these outlined approaches focused on handling or analysing SAB terms prevalent in social media streams.

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) clustering algorithm was proposed by [23]. The research work was based on detecting localized events and tracking the evolution of such events. Spatio-temporal characteristics of keywords were continuously extracted using the entropy of the spatial signature. A single-pass clustering algorithm, Birch, was used to group event keywords based on the cosine similarity of their spatial signatures. Top-k scoring clusters were considered possible event clusters. In the pre-processing stage, stop-words were removed, stemming was applied, and WordNet (a lexical dictionary for English words) dictionary lookups were performed. Authors in [24] employed multiple social media feeds features such as titles, description, location-textual, location proximity, and date along with Term Frequency—Inverse Document Frequency (TF-IDF) and

Normalized Mutual Information Frequency to detect events. In the same vein, [25] traced the German centennial flood in the stream of tweets. The authors applied density-based clustering called Ordering Points to Identify the Clustering Structure (OPTICS) to group a set of flood-related tweets with respect to time and location. The result was validated with subsidiary data sources. Fuzzy hierarchical agglomerative clustering was used to propose a TweetMogaz framework for identifying new stories in social media [26]. The framework used an adaptive method to track relevant tweets. Fuzzy Hierarchical Agglomerative clustering with term co-occurrence probability as a distance measure was used to identify hot stories tweets with enough content. To overcome the problem of duplicate story detection, cosine similarity was used to compute vectors of the two clusters.

Also, real-time entity-based event detection for Twitter was proposed by [17]. The proposed approach identified bursty named entities and then clustered tweets based on the occurrence of the named entities using a cosine distance similarity score. Multiscale event detection in social media was presented by [27]. The authors explored the properties of the wavelet transform. They proposed a novel algorithm to compute a data similarity graph at appropriate scales and simultaneously detect events of different scales by a single graph-based clustering process. The clustering process is based on comparing common terms between pairs of tweets. Authors in [28] proposed a bursty event detection from a microblog framework using a distributed and incremental approach. The paper focuses on detecting events from Weibo (microblog) on Spark engine framework (taking into consideration of topic drift) by employing distributed and incremental temporal topic model, Bursty Event dEtection (BEE+). Online indexing and clustering of social media data for emergency management was presented by [29]. The authors implemented online indexing techniques: incremental TF-IDF; Skewness; and Learn & Forget Model. Clustering was evaluated using Silhouette and Davies-Bouldin metrics. Authors in [13] presented three different approaches to merging information from two different social media sources using time-evolving graphs. It was demonstrated that using information from multiple data streams increases the quality and quantity of detected events. An event detection system that uses inverted indices and incremental clustering algorithms was proposed by [30]. Burst detection based on the volume of tweets without considering the tweets' context may be misleading because co-occurrence terms in tweets may not be synonymous when the context in which they are used is taken into consideration. Real-time event detection on social data streams was conducted by [31]. Events were modelled as a list of clusters of trending entities over time using entity co-occurrence, Louvain clustering, and aggregate ranking. The approach only considered the length of words contained in a tweet but did not look at such representative's local and global importance. In addition, SAB terms were not handled. Authors in [32] proposed a multimedia big data system that used both incremental clustering event detection approach enriched with the analysis of multimedia content and a bio-inspired influence analysis technique to support alert spread and situation awareness over the network. Target-aware holistic influence maximization in spatial social networks was carried out by [33]. The authors came up with a diffusion model which takes care of both

Table 1 Unsupervised learning approaches used for event detection

References	Data sources/ Features	Algorithms	Inclusion of local vocabulary	Treatment of Slang, Acronym, and Abbreviation (SAB)
[14]	Twitter/ Textual	LSH, Shannon entropy	No	No
[20]	Twitter/ Textual	EDCoW, Term Frequency	No	No
[21]	Twitter/ Textual	Term Frequency, Kullback–Leibler divergence	No	No
[15]	Twitter/ Textual	Similarity score, Cluster summarization	No	No
[26]	Twitter/ Textual	Fuzzy Hierarchical, Agglomerative Clustering	No	No
[24]	Twitter/ Textual, Spatiotemporal	TF-IDF, Normalised Mutual Information Frequency	No	No
[16]	Twitter/ Textual	Named entity, TF-IDF, Sigma rule	No	No
[23]	Twitter/ Textual, Spatio-temporal	BIRCH	No	No
[25]	Twitter/ Textual	OPTICS	No	No
[27]	Twitter/ Textual, Spatiotemporal	Wavelet decomposition, Modularity-based clustering	No	No
[28, 34]	Weibo, Twitter/ Textual	Expected Maximization, MapReduce, K-means, Hierarchical agglomerative	No	No
[29]	Twitter, Flickr, YouTube/Textual	Incremental TF-IDF, Skewness, Learn and Forget term selection, Growing, Gaussian Mixture Model	No	No
[12]	Twitter, Tumblr/ Textual	Time-evolving graphs	No	No
[30]	Twitter/ Textual	Longest Common, Subsequence, Incremental clustering	No	No
[31]	Twitter/ Textual	Entity co-occurrence, Louvain clustering, Aggregate ranking	No	No
[32]	Twitter/Multimedia	Incremental clustering, Influence maximization algorithm	No	No
[33]	Twitter, Facebook, Weibo/ Textual	Indexed based algorithm	No	No
[35]	Twitter, Weibo/Textual	Sub-event representation learning	No	No
[36]	Twitter/Textual	Spatiotemporal clustering	No	No

physical and cyber user interactions. They also proposed a spatial, social index based on an R-tree algorithm that computes users' interest similarity concerning online keyword queries. Both synthetic and three datasets were used to validate the effectiveness of the proposed model. Authors in [35] improved on the drawbacks of conventional methods to detect sub-events from social media by proposing a hashtag-based sub-event detection framework for social media. In the same vein, authors in [36] proposed a spatiotemporal clustering-based method to detect traffic events using geosocial media data.

However, these approaches did not handle the semantic analysis of SAB in social media content. The summary of unsupervised learning approaches to event detection is shown in Table 1.

Semi-supervised learning for event detection in social media stream

Semi-supervised learning models are trained by combining both the unlabeled and labelled data. More specifically, a little proportion of labelled data with a great deal of unlabeled data. Some of the event detection efforts based on semi-supervised learning are now presented.

Authors in [37] identify and characterize social media events by using generic event detection and topic-specific event detection with TF-IDF and Naïve Bayes. Civil unrest prediction with a Tumblr-based exploration was reported by [38]. The authors focused on detecting civil unrest by continuously applying text-based filters (keyword, location and future date filters) to the Tumblr data stream. A semi-supervised method for Automatic Targeted-domain Spatio-temporal Event Detection (ATSED) in Twitter using historical and real-life Twitter streams was proposed by [39]. The proposed method was suitable for event detection from historical data but not for real-time event detection. SPOTHOT: Scalable detection of geo-spatial events in large textual streams was proposed by [40]. The authors proposed a SigniTrend event detection system capable of tracking unusual occurrences of arbitrary words at arbitrary locations in real-time without specifying the terms of interest in advance. None of these outlined methods handled the noisy characteristics inherent in social media data.

Also, [41] implemented various algorithms like k-means, Hierarchical agglomerative, and Latent Dirichlet Allocation (LDA) topic modelling on Twitter stream to analyze real-time Twitter data to empower citizens by keeping them updated about what is happening around the city. In the pre-processing stage, removal of hashtag, stop-word, URL and special characters and stemming were done, but there was no treatment of SAB terms. Authors in [42] proposed a model for detecting and tracking

Table 2 Semi-supervised learning approaches used for event detection

References	Data sources/ Features	Algorithms	Inclusion of local vocabulary	Treatment of Slang, Acronym, and Abbreviation (SAB)
[37]	Twitter/Textual	TF-IDF, Naive Bayes	No	No
[38]	Tumblr/Textual, Spatiotemporal	MapReduce, Pig	No	No
[39]	Twitter/Textual, Spatiotemporal	ATSED	No	No
[40]	Twitter/Textual, Spatiotemporal	Geometric Discretization, Administrative Hierarchies	No	No
[41]	Twitter/ Textual	K-means, LDA, Hierarchical agglomerative	No	No
[42]	Twitter/ Textual	Multinomial Naïve Bayes, Incremental DBSCAN	No	No
[18]	Twitter/ Textual	ANN, AvgW2V, Mini-batch cluster	No	No
[43]	Twitter/ Textual	Hierarchical Dirichlet Process	No	No

breaking news from Twitter in real-time by employing Multinomial Naïve Bayes Classifier and DBSCAN algorithms. The proposed model could not dynamically learn from the available new sources. The pre-processing stage removed tags, mentions, URLs, and non-ASCII characters but did not address SAB terms prevalent in social media posts. Authors in [18] proposed a framework for detecting news events from the Twitter stream in real-time. The approach used ANN to classify news relevant tweets from the stream based on AvgW2V and Mini-batch cluster to group detected tweets into events. Authors in [43] worked on sub-story detection in Twitter with a hierarchical Dirichlet process. The paper proposed a Hierarchical Dirichlet Process (HDP) to address the problem of automatic substory detection associated with the main story. Like others, none of these research efforts considered resolving the ambiguity of SAB terms during the analysis of social media content. The summary of semi-supervised approaches to event detection is presented in Table 2.

Supervised learning for event detection in social media stream

The supervised learning models are the class of machine learning algorithms that can extrapolate a prediction or classification function after being trained on labelled sample data. The training examples contain a couple of input (vector) and output (supervisory signal). Instances are of the format (x,y), where x is a vector and y is referred to as the class or target attribute (or scalar). Supervised learning approaches typically build a model that maps x to y by finding a mapping m(.) such that m(x) = y. Given an unlabeled instance, m(x) and m(.) learned from training data, the outcome of an unlabeled instance can be computed. Subsequently, some instances of supervised learning applied to event detection are presented next.

Authors in [44] proposed Geo-spatial event detection in a Twitter stream. Machine learning algorithms (Naïve Bayes, Multilayer perceptron, and Prune C4.5) were used to analyse whether the geo-spatial clusters contain real-life events. The detected events (candidate clusters) were displayed according to the individual tweet ranking score in descending order on a map with their locations in real-time. Authors in [45] proposed a graphical-based model, location-time-constraint topic, and LTT (an improvement over LDA) to capture social media time, content, and location data. Kullback Leibler, KL-divergence was used to measure the similarity of uncertain media content. Social events were detected using a hash-based indexed scheme, Variable Dimensional Extendible Hash (VDEH). The LTT model was refreshed after every block of tweets in an incoming time slot to accommodate topic drift. Transaction-based Rule Change Mining (TRCM) framework that applied Association Rule Mining to extract association rules from the tweet's hashtag was proposed by [46]. Unexpected changes in the consequent and conditional rules in each time slot were ranked. Hashtags detected were then compared with the key terms in the ground truth from BBC Sport commentary within the same time frame. Authors in [47] studied real-time top-R topic detection on Twitter with topic hijack filtering. The extraction of meaningful topics and noisy filtering messages over the Twitter stream were integrated using Streaming Non-negative Matrix (NMF). There were false detections (false negatives) of hijacked topics due to the model misspecification. Twitter Life Detection Framework was presented by [48] using TF-IDF for similarity score and SFPM for classification.TF-IDF directly computes document similarity on the word-count space, which is usually slow for large vocabularies. None of these reported approaches focused on treating SAB terms in tweets.

A multimodal classification of events in social media using TF-IDF and SVM was presented by [49]. The pre-processing stage removed stop-words, special characters, numbers, emoticons, HTML tags, and words with less than four characters. Authors in [50] proposed an audio-based multimedia event detection using recurrent neural networks. The authors introduced longer-range temporal information with a recurrent neural network for feature representation and classification to determine whether a given event can be traced to a video. In the same vein, multimedia event detection was presented by [51]. The author proposed algorithms for detecting complex event detection from web videos by engaging a two-stage convolutional neural network. Our focus in this paper is to use social media contents which are very noisy due to user-generated content (social media content). These reported efforts on multimedia event detection did not specifically address SAB terms and grammatical errors that may be contained in video data.

Also, [52] proposed an approach to detect Foodborne disease from Weibo data using TextRank and SVM. The SVM was used to filter unwanted tweets. However, the proposed framework was found to perform poorly in the face of sparsity and concept drift. A deep learning approach for traffic accident detection from social media was developed by [53]. Tokens and paired tokens were extracted from over 3 million tweets. Deep Belief Network and Long Short-Term Memory deep learning models were implemented on the extracted tokens and paired tokens to detect traffic accident information. Authors in [54] proposed a hate speech detection model to identify hatred against vulnerable minority groups using Amharic text data on Facebook. Apache Spark distributed platform was used for data pre-processing and feature extraction. Feature extraction was done using

Table 3 Supervised learning approaches used for event detection

References	Data sources/ Features	Algorithms	Inclusion of local vocabulary	Treatment of Slang, Acronym, and Abbreviation (SAB)
[45]	Twitter/Textual, Spatial	Variable Dimensional Extendible Hash	No	No
[46]	Twitter/Textual	Association Rule Mining	No	No
[44]	Twitter/Textual, Geo-spatial	Naïve Bayes, Multilayer perceptron, and Prune C4.5	No	No
[47]	Twitter/Textual	Streaming Non-negative Matrix	No	No
[48]	Twitter/Textual	Soft Frequent Pattern Mining	No	No
[49]	Flickr, Instagram/ Multimodal	TF-IDF, SVM	No	No
[50]	Video/Multimedia	RNN, LSTM	No	No
[51]	Video/Multimedia	CNN, Smoothing technique	No	No
[52]	Weibo/Textual	TextRank, SVM	No	No
[53]	Twitter/Textual	Deep Belief Network, LSTM	No	No
[54]	Facebook/Textual	Word2Vec, Random Forest, Gated Recurrent Unit, LSTM	No	No

Word2Vec as an embedding model. Gated Recurrent Unit (GRU) was used for the classification stage. Table 3 summarizes supervised learning approaches that were applied to event detection.

Semantic-based approaches for event detection in social media stream

Authors in [55] worked on scalable distributed event detection using Twitter streams. The paper proposed scalable automatic distributed real-time event detection by incorporating a lexical key partitioning strategy (hash key grouping borrowed from LSH) to spread the detection process across multiple machines while avoiding partitioning as a series of subsets. The proposed framework was implemented on the Storm topology. It was identified that no pre-processing was done, even though Twitter streams are noisy, temporal, and full of slang. Authors in [56] proposed Locality Sensitive Hashing (LSH) to detect events from Twitter and Facebook. LSH was used twice in the event detection process; it was used to obtain events from Twitter and Facebook independently. It was later applied to detect cross-over events in the two social media streams. LITMUS, a system that used keywords to extract social media data related to "landslide", was proposed by [57]. The system then employed an augmented Explicit Semantic Analysis (ESA) algorithm using a semantic interpreter by extracting a subset of Wikipedia as classification features to classify data into relevant and irrelevant. Semantic clustering based on semantic distance was used for location estimation. Only geo-tagged data were considered and not the entire dataset. These semantic-based approaches did not consider the treatment of SAB terms in their analysis.

Authors [58] presented a system, ArmaTweet, which used Natural Language processing techniques to extract structured information from tweets and then integrated the structured information with RDF from DBpedia and WordNet. The system used semantic queries to identify tweets matching the user interest and passed them to the anomaly detection algorithm to determine their correspondence to actual events. This improves the keyword search and is suitable for topic-specific event detection. However, the precision of the pre-processing component was not investigated in the face of acronyms, slangs, abbreviations, and passive words prevalent in social media data.

Table 4 A summary of semantic-based approaches used for event detection

References	Data sources/ Features	Algorithms	Inclusion of local vocabulary	Treatment of Slang, Acronym, and Abbreviation (SAB)
[56]	Twitter, Facebook/ Textual	LSH	No	No
[55]	Twitter/ Textual	Hash key grouping	No	No
[58]	Twitter/ Textual	RDF	No	No
[59]	Twitter/ Textual	TF-IDF, Named entity Recognition, Page Rank, CfsSubsetEval	No	No
[60]	Twitter/Textual	IPLSA, EM, RS Scoring algorithm, word2vec	No	No

A framework for event classification in tweets based on hybrid semantic enrichment using TF-IDF, Named Entity Recognition, Page Rank, CfsSubsetEval was proposed by [59]. Semantic enrichment was combined with external document enrichment and named entity extraction to classify tweets. Authors in [60] proposed an event detection model based on scoring and word embedding to discover key events from a high volume of data streams. In the pre-processing stage, stop words, modal auxiliary verbs, URLs, and emoticons were removed. Word2vec was used for embedding, and improved Expected maximization was used for the event detection stage. Word2Vec is limited to calculating word similarities. However, none of these approaches considered resolving the ambiguity associated with SAB terms in social media. A summary of the instances of applying semantic-based approaches for event detection methods is presented in Table 4.

Our review of the literature revealed that existing event detection methods had focused mostly on filtering out SAB, removing noisy terms including SAB, or ignoring them entirely during the pre-processing stage of social streams. They did not perform semantic analysis of noisy terms like SAB to determine their contextual meanings and their impact on the accuracy of results. These noisy terms include short messages, slangs, acronyms, mixed languages, grammatical and spelling errors, dynamically evolving, irregular, informal, abbreviated words, and improper sentence structure, which make it challenging for the efficient performance of the learning algorithms [7, 61]. This gap that was not addressed by previous research efforts necessitated our study.

According to [30], the representation of social media stream must be in a way that the semantics of social media content is preserved. Hence, using the contextual clues surrounding a social media stream is critical for useful and accurate results. Thus, there is a need to develop an event detection framework that will focus on semantic analysis of slangs, acronyms, and abbreviations (SAB) terms in social media streams; and the ambiguity associated with their usage to improve the accuracy of event detection in social media streams. Most of the previous research efforts have not addressed this problem, which is where SMAFED seeks to make a difference. The summary of the strength and weaknesses of existing event detection techniques and their attributes is presented in Table 5.

Table 5 Summary of strength and weaknesses of event detection techniques and their attributes

Unsupervised learning approaches	Semi-supervised learning approaches	Supervised learning approaches	Semantic-based approaches
Strength •Detecting events without any particular regard to their nature •Can handle a large volume of data in real-time Weakness •Difficult in dealing with a high dimensionality data stream •It does not consider spatial relationships in the data	•Strength •Particularly useful when it is difficult to extract relevant features from data •Small amount of data can lead to a significant accuracy improvement Weakness •Iteration results are not stable •Low accuracy	Strength •Results are highly accurate and trustworthy Weakness •Time-consuming •Large amount of data to be trained •Handling concept drift •Labels for input and output variables require expertise	Strength •Provide contextual knowledge •Valuable for sense disambiguation •User-centric results •More precise results Weakness •Difficult to construct

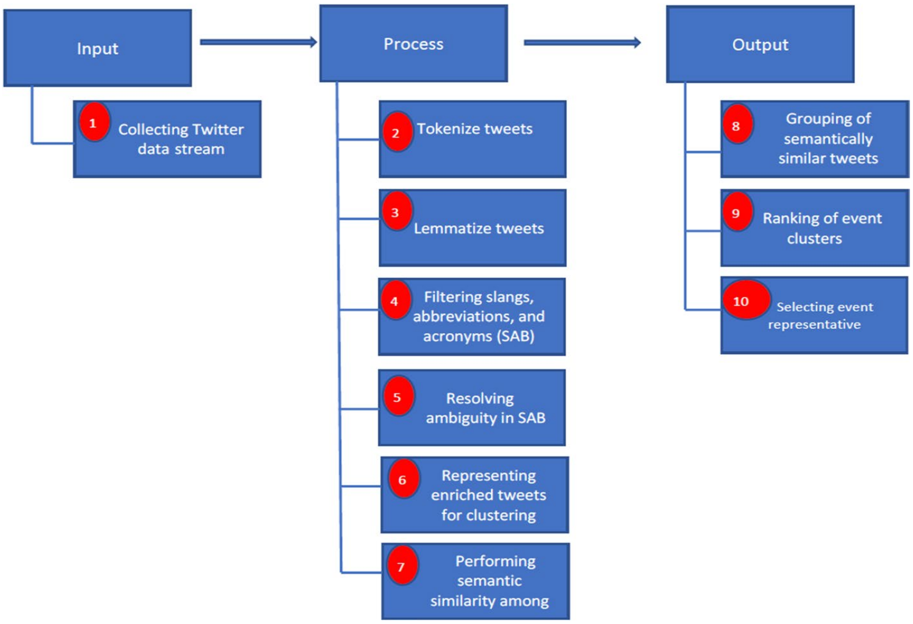

Input
Process
Output
1 Collecting Twitter data stream
2 Tokenize tweets
3 Lemmatize tweets
4 Filtering slangs, abbreviations, and acronyms (SAB)
5 Resolving ambiguity in SAB
6 Representing enriched tweets for clustering
7 Performing semantic similarity among
8 Grouping of semantically similar tweets
9 Ranking of event clusters
10 Selecting event representative

Fig. 1 Main Tasks During Event Detection using SMAFED

Methodology

This paper proposes the Social Media Analysis Framework for Event Detection (SMAFED) as an efficient and integrated social media stream analysis approach incorporating social media stream pre-processing and enrichment to improve event detection results.

Description of main tasks during event detection using SMAFED

Our proposed approach to event detection in social media streams consists of ten main tasks (1–10) that are sequentially structured and can be abstracted—by the Input-Process-Output model as shown in Fig. 1. The input phase consists of task 1, tasks 2–7 constitute the process phase, while tasks 8–10 constitute the output phase. The tasks are outlined as follows:

1. Collecting Twitter data stream;
2. Tokenize tweets;
3. Lemmatize tweets;
4. Filtering slangs, abbreviations, and acronyms (SAB) from tweets;
5. Resolving ambiguity issues in the usage of slangs, abbreviations, and acronyms through disambiguation;
6. Representing enriched tweets in a way that will be suitable for clustering;
7. Performing semantic similarity among tweets
8. Grouping of semantically similar tweets into clusters
9. Ranking of event clusters
10. Selecting event representative.

Formal definition of main tasks in SMAFED

We now present the formal definitions of the main tasks of our approach to event detection as follows:

Definition T1 (*Data Streams Collection)* A stream $S = e_1, e_2, \ldots, e_n$ is an ordered sequence of objects or points where e_i indicates the *ith* object or point observed by the algorithm. For $t > 0$, let *S(t)* symbolizes the first t entries of the stream: $e_i, e_{i+1}, \ldots, e_t$. For $0 < i \leq j$, let $S(i,j)$ designate the substream $e_i, e_{i+1}, \ldots, e_j$. Define $S = S(1,n)$ be the whole stream observed until e_n, where n is, as before, the total number of objects or points observed so far.

Definition T2 (*Tokenize Tweets)* Given a stream $S = e_1, e_2, \ldots e_n$where e_i represents an individual tweet, tokenize e_i such that $w_i \in W$ and $i = 1, 2, \ldots, m$, where w_i is the individual words and W is all the words in e_i.

Definition T3 (*Lemmatize Tweets)* For $w_i \in W$, lemmatize w_i such that $r_i = lemma(w_i, k_i)$, where k_i is the pos_tag with wordnet value and r_i is the root word of w_i.

Definition T4 (*Filtering SAB)* Given a stream $S = e_1, e_2, \ldots e_n$ at a time t where e_i represents an individual tweet containing words $w_i \in W$ and $i = 1, 2, \ldots, m$. Find w_i such that $w_i \nexists D$ where D is the set of English words.

Definition T5 (*Disambiguating SAB)* Let the size of the social media stream context window, 2n + 1, be denoted as N. Given local vocabulary (IKB) as the integrated database containing definitions, usage examples, and related terms of SAB. Let the IKB SAB terms in the context window be represented as $\mathrm{W_i}$, $\mathrm{i} \leq 1 \leq \mathrm{N}$. If the number of IKB SAB terms is less than 2n + 1, all of the IKB SAB terms in the instance serve as the context.

Each SAB term $\mathrm{W_i}$ has one or more possible senses. Let the number of senses of the SAB term be represented as $|\mathrm{W_i}|$. Each possible combination of senses for SAB in the context window will be evaluated. There are $\Pi_{i=1}^{N}|W_i|$ such combinations, each of which is referred to as a candidate combination. A combination score is computed for each candidate combination. The target SAB term is assigned the sense of the candidate combination that attains the maximum score.

Definition T6 (*Semantic Tweets Representation)* Given context or source embedding v_w and target embedding u_w for each word w in the vocabulary with embedding dimension h and $k = |v|$. The tweet embedding is the average context word embeddings of constituent words augmented by learning n-grams. The tweet embedding v_s for current tweet S is modelled as:

$$v_s := \frac{1}{|R(S)|} V_{l_{(RS)}} = \frac{1}{|R(s)|} \sum_{w \in R(S)} v_w \tag{1}$$

where *R*(*S*) designates the list of n-grams, including unigrams present in sentence *S*.

Definition T7 (*Semantic Similarity among tweets)* Assume two tweets *x* has m words $x_1, x_2, \ldots, x_m$ and y has n words $y_1, y_2, \ldots, y_n$. The semantic similarity matrix (SSM) for two tweets *x* and *y* is given as:

$$SSM = \begin{pmatrix} sim\left(x_1, y_1\right) & \cdots & sim(x_1, y_n) \\ sim(x_2, y_1) & \ldots & sim(x_2, y_n) \\ \vdots & \cdots & \vdots \\ sim(x_m, y_1) & \cdots & sim(x_m, y_n) \end{pmatrix}$$

The semantic similarity between word $\mathrm{x_s}$ and tweet b is given as follows:

$$sim(x_s, b) = \max(sim(x_s, b_1), \ldots, sim(x_s, b_n)) \tag{2}$$

The semantic similarity between tweets *x* and *y* is calculated as:

$$SIM\left(x, y\right) = \frac{\sum_{s=1}^{m} similarity(x_s, b)}{m} \tag{3}$$

The semantic relatedness of x_i and y_i is calculated by comparing glosses of synsets related to x_i and y_i through explicit relationships of *IKB*.

Definition T8 (*Grouping of semantically similar tweets into clusters)* Given the assumption of the distribution of data stream uploads belonging to an event, an incremental clustering algorithm can be defined as follows:

First, the small size of the window C_1 of the data stream of size, *N* is clustered such that $C_1 \leq N$. As a new data stream arrives on window C_2 Clustering is performed again with $|C_2| = 2 * |C_1|$.

If certain clusters detected in the window C_1 are re-detected in C_2, then those clusters that are "stable" remain stable, and their items are removed from further clustering. For subsequent data stream clustering on window C_3, there is likely to be $|C_3| = |C_2| - \sum_i C_i + |C_i|$, where C_i is the set of stable clusters.

Definition T9 (*Event Cluster Ranking)* The importance/information richness of a cluster is based on the number of important words it contains, the Weight of a cluster *C*, $W(C)$ is computed as follows:

$$W(C) = \sum_{w \in C} \log(1 + count(w)) \tag{4}$$

where $count(w)$ is the count of the word w in the input collection and the $count(w)$ is greater than a given threshold.

Definition T10 (*Representative Event Selection)* A candidate for representation is selected based on the importance of its constituent words. Let the local importance of word w be given as $\log(1 + CFT)$, where CFT is the cluster term frequency. Let the global importance be $\log(1 + CF)$, where CF is the cluster frequency. The importance of word w is the average of the local and global importance given as $Weight(w) = \alpha_1\log(1 + CTF) + \alpha_2\log(1 + CF)$, where $\alpha_1 = \alpha_2 = 0.5$ (constant). Tweet with $max(Score(D_i))$ is selected as an event candidate for each of the clusters.

Summarily, the formal definition of the research problem is specific to social media streams and can be defined as follows:

Given a set of Twitter streams, T is a set of tweets $T = \{t_1, t_2, \ldots, t_n\}$ and$T \in R^{nXt}$. The k-NN of data point $t \in R^t$ in T can be denoted as $N_k(t)$, we need to extract a set of $\{t, N_k(t)\}$ where $0<=k<=n$ based on semantic distance $diff(t, N_k(t))$ between the data point t and $N_k(t) : d(t, N_k(t)) = diff|t, N_k(t)|$ and then classify $\{t, N_k(t)\}$ as $eventE = e_1, e_2, \ldots, e_m$, where $\{t, N_k(t)\} \geq$ thresholdϵ.

High-level overview of SMAFED

We now present the high-level process view of SMAFED. The process workflow of SMAFED (see Fig. 2) is divided into four main steps described below.

Step 1: A user interface is built around the underlying API provided by Twitter using Python Programming Language to collect tweets in English or Pidgin English from Nigeria origin. Python is chosen due to its efficiency and suitability for building high traffic and data-heavy workflows. Collected tweets within each window period are stored in a queue. The collected tweets are passed to the pre-processing stage.

Step 2: From the data stream collected, URLs, Tags, mentions and non-ASCII characters were automatically removed through the use of a regular expression. Then, the

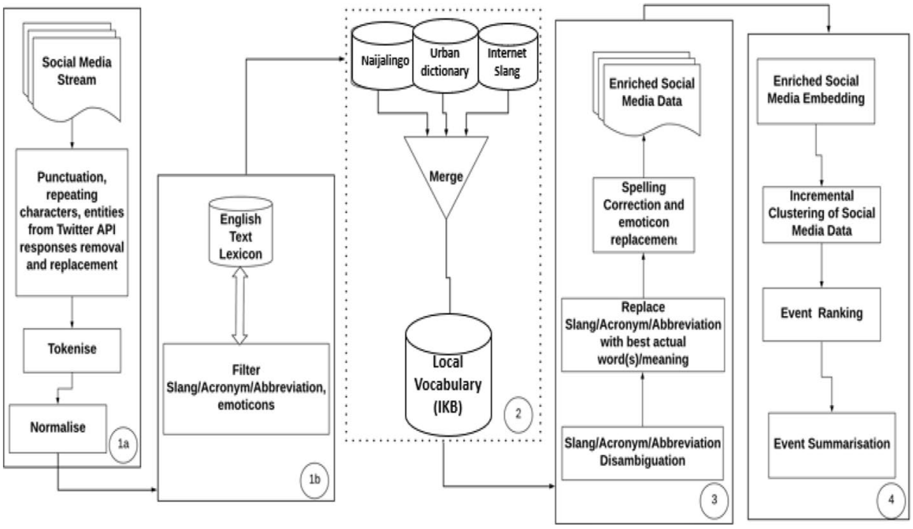

Social Media Stream
Punctuation, repeating characters, entities from Twitter API responses removal and replacement
Tokenise
Normalise
1a
English Text Lexicon
Filter Slang/Acronym/Abbreviation, emoticons
1b
Naijalingo
Urban dictionary
Internet Slang
Merge
Local Vocabulary (IKB)
2
Enriched Social Media Data
Spelling Correction and emoticon replacemen
Replace Slang/Acronym/Abbreviation with best actual word(s)/meaning
Slang/Acronym/Abbreviation Disambiguation
3
Enriched Social Media Embedding
Incremental Clustering of Social Media Data
Event Ranking
Event Summarisation
4

Fig. 2 A View of SMAFED Process Workflow

next data preparation stage was to perform tokenisation and normalisation. This basic pre-processing reduces the number of features and addresses the problem of overfitting (Romero & Becker, 2019). After that, slang, acronyms, and abbreviations (SAB) are filtered from the tweets using corpora of English words in the natural language toolkit (NLTK). The filtered SAB terms are then passed to the local vocabulary (IKB) for further processing.

Step 3: Meanings of SAB are extracted from IKB. Due to several meanings attached to each SAB, there is a need to disambiguate the ambiguous terms and select the best sense from the several meanings provided. This is done by leveraging the Slang, Acronym, and Abbreviation, Disambiguation Algorithm (SABDA) based on the ambiguous SAB's context in the tweet. The peculiarity informed the choice of SABDA of the IKB to provide a rich source of information and improve overall disambiguation accuracy. The data enrichment stage is then concluded with spelling correction (using Python's automatic spell-checker library) and emoticon replacement.

Step 4: The enriched tweets produced from the previous stage must be transformed to enable the clustering algorithm to build on them. The enriched tweets are transformed into a vectorial form using the sent2vec model. Sent2Vec provides a significant improvement over state-of-the-art supervised and unsupervised methods for sentence or paragraph embedding, as revealed in the literature [62]. The embedded tweets are clustered as they arrive using semantic histogram-based incremental clustering (SHC) [63]. The idea is to have clusters representing the same event as much as possible. SHC maintains high cohesiveness within clusters, implying a high distribution of similarities. This necessitates the choice of SHC. The tweets in each cluster are ranked based on the information richness of their constituent words. Lastly, the representative tweet that best describes each of the top *n* candidate event clusters is selected.

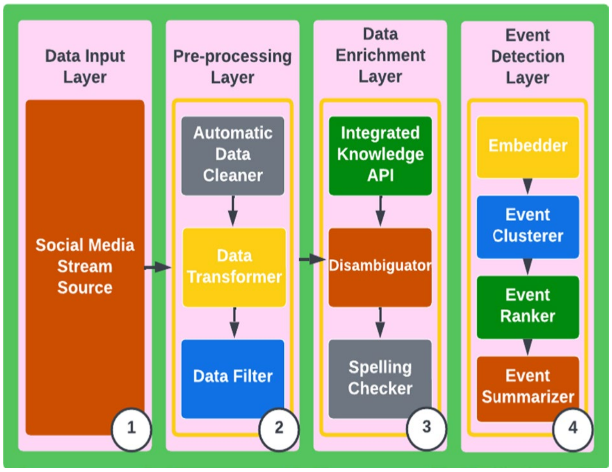


Fig. 3 Overview of the SMAFED Architecture

The conceptual architecture of SMAFED

The conceptual architecture defines the structure of components of the SMAFED**.** It consists of four modules: data collection, data pre-processing, data enrichment, and event detection, as presented in Fig. 3.

Data input layer

The data layer serves as the input layer of SMAFED. It is responsible for streaming tweets from Twitter to SMAFED for processing. A user interface enabled by a Twitter API is used for tweet streaming in JSON formats. The input to SMAFED is data at rest stored either in the Comma Separated Value (CSV) or JavaScript Object Notation (JSON) format.

Data pre-processing layer

The data pre-processing layer comprises three sub-layers: data cleaner, data transformer, and data filter. The data cleaner handles data cleaning of responses fetched through the Twitter API: punctuations, repeated characters elimination, and substitution. The data transformer and the data filter, both in the pre-processing layer of SMAFED, perform the feature extraction. The data transformer performs tokenisation and normalisation using the Python NLTK library. After that, the data filter extracts the SAB from tweets collected using corpora of English words in the Python NLTK library. In other words, any normalized token that is not found in the corpora of English words is taken as slang or abbreviations or acronyms or emoticons. The sifted SAB is transferred to the data enrichment layer for additional processing. The tweet being analysed, and the SAB serve as input to the enrichment layer.

Data enrichment layer

This layer has three sub-components: IKB API, Disambiguator, and Spelling Checker. To better represent tweets, there is a need to provide meaning for slang, acronyms, and abbreviations (SAB) found in tweets because these noisy contents in tweets have hidden meanings that can form part of the rich context of tweets well defined. The IKB component of SMAFED is a lexicon of SAB that stores all the contents of the three knowledge sources: *Naijalingo*, *Urban dictionary*, and *Internet slang* in MongoDB. *Naijalingo* is an online Nigerian Pidgin English and slang words reference that gives definitions to Nigerian words and expressions. *Urban dictionary* is a publicly supported online word reference for English slang words and expressions. *Internet slang* is a word reference containing a pool of slang terms, acronyms, and abbreviations on online blogs, Twitter, chat rooms, SMS, and internet forums. The IKB includes about 2 million defined SAB terms and their usage examples and related terms.

The Disambiguator, which is a sub-component of the IKB API, is responsible for the disambiguation of ambiguous SAB. The Slang, Acronym, and Abbreviation Disambiguation Algorithm (SABDA) determines the semantic sense associated with specific terms in the IKB. SABDA adapts the original Lesk algorithm [64] to disambiguate slang, acronyms, and abbreviations in social media content. Instead of looking at the glosses of definition for a term in the WordNet as the Lesk algorithm does it, SABDA looks at the usage examples for SAB terms in the IKB. SABDA derives the proper sense in which noisy terms which are not available in the WordNet (WordNet is a database of regular English lexicon) are used. Since the current context (i.e. the tweet being analysed) is similar to how the SAB term is used, measuring the overlap between the usage examples and the current context would produce a better result. SABDA measures the overlap between senses of usage examples of SAB terms as defined in the IKB and the usage context of the SAB term in a tweet that is being analysed. The usage example with the highest overlap is then mapped with the respective definition, which replaces the SAB term in the target tweet.

The last stage of the enrichment layer is to perform a spelling check on the tweet content. JamSpell version 1.0.0 from the python library is a spell-checking tool that is efficient and effective. It considers words context for better correction (accuracy), can correct up to 5000 words/sec (speed) and is available for many languages (multi-language). The choice of JamSpell was informed by its better performance when compared to other spell-check libraries such as Norvig, Hunspell, and Dummy in terms of speed and accuracy.

The formal definition of the SABDA model The formal definition of the SABDA model is as follows:

If $u = u_1, u_2, \ldots u_n$ and $c = c_1, c_2, \ldots c_n$ are the usage gloss and the context, respectively, we build their semantic representation u and c in the semantic space through the addition of word vectors belonging to them:

$$u = u_1 + u_2 + \cdots + u_n$$

$$c = c_1 + c_2 + \cdots + c_m$$

The measure of relatedness between u and c is a measure of the similarities between u and c given as:

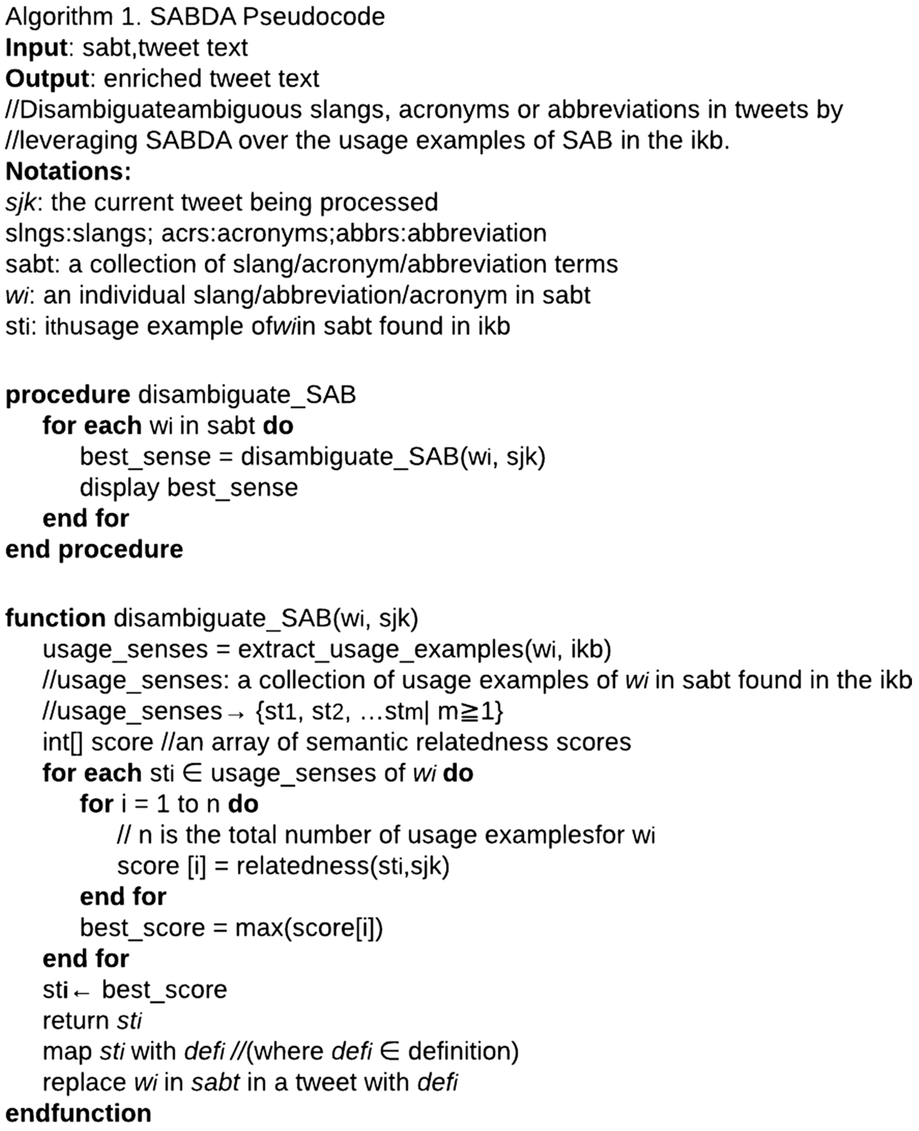
$$Relatedness(u, c) = \sum_{i=1}^{N} score(R_{1i}(u_i), R_{2i}(c_i)),$$

where N is the number of pair relations in *RELPAIRS* which is defined in a reflexive relation given as:

$$RELPAIRS = (R_1, R_2)|R_1, R_2 \in RELS;$$

if $(R_1, R_2) \in RELPAIRS, thenR_2, R_1 \in RELPAIRS$
where RELS is a set of relations.
To choose the best $def_{\,i}$,
Map the $\max(Relatedness(u, c))$ with the corresponding $def_{\,i} \in definition.$

The algorithm for disambiguation of SAB (SABDA) The pseudocode of the Slang, Acronym, and Abbreviation Disambiguation algorithm (SABDA) is presented in Algorithm 1.



```
Algorithm 1. SABDA Pseudocode
Input: sabt,tweet text
Output: enriched tweet text
//Disambiguateambiguous slangs, acronyms or abbreviations in tweets by
//leveraging SABDA over the usage examples of SAB in the ikb.
Notations:
sjk: the current tweet being processed
slngs:slangs; acrs:acronyms;abbrs:abbreviation
sabt: a collection of slang/acronym/abbreviation terms
wi: an individual slang/abbreviation/acronym in sabt
sti: ithusage example ofwiin sabt found in ikb

procedure disambiguate_SAB
    for each wi in sabt do
        best_sense = disambiguate_SAB(wi, sjk)
        display best_sense
    end for
end procedure

function disambiguate_SAB(wi, sjk)
    usage_senses = extract_usage_examples(wi, ikb)
    //usage_senses: a collection of usage examples of wi in sabt found in the ikb
    //usage_senses → {st1, st2, …stm| m≧1}
    int[] score //an array of semantic relatedness scores
    for each sti ∈ usage_senses of wi do
        for i = 1 to n do
            // n is the total number of usage examplesfor wi
            score [i] = relatedness(sti,sjk)
        end for
        best_score = max(score[i])
    end for
    sti ← best_score
    return sti
    map sti with defi //(where defi ∈ definition)
    replace wi in sabt in a tweet with defi
endfunction
```

Illustration of SABDA pseudocode For clarity, the disambiguation and interpretation of the noisy terms process are illustrated in Example 1.

Example 1: Consider the case of disambiguating the term "*baddo*" in the tweet: "I am a baddo when it comes to this profession." Given the following senses from the ikb as shown in Table 6, pick the sense with the most word overlap between the context (tweet in question, sjk) and the usage examples (usage_senses). The overlap between the context and the usage example is shown in Table 7.

The tweet that is being considered, *sjk*, "I am a baddo when it comes to this profession", is contrasted with all the usage examples (usage_senses) in the ikb identifying with the word "baddo". The score for every comparison (relatedness(*sti*,*sjk*)) is stored

Table 6 Senses from the IKB

baddo[1]	Definition: When something bad happens, or something goes wrong Usage example: She said she wants a break, baddo Related term: bad
baddo[2]	Definition: Someone who is highly respected or seen as very good at what they do Usage example: I be baddo when it comes to computing Related term: baddo, best, respected, influential
baddo[3]	Definition: A shortened, more legit name for a badass Usage example: I know, what a baddo Related term: baddo, finger food
baddo[4]	Definition: A rear-end that generates noxious emission Usage example: The fume from this generator is baddo Related term: harmful, poisonous, unpleasant

Table 7 Senses from IKB with overlap

baddo[1]	Definition: When something bad happens, or something goes wrong Usage example: She said she wants a break, baddo Related term: bad	2 overlaps
baddo[2]	Definition: Someone who is highly respected or seen as very good at what they do Usage example: I be baddo when it comes to computing Related term: baddo, best, respected, influential	6 overlaps
baddo[3]	Definition: A shortened, more legit name for a badass Usage example: I know, what a baddo Related term: baddo, finger food	2 overlaps
baddo[4]	Definition: A rear-end that generates noxious emission Usage example: The fume from this generator is baddo Related term: harmful, poisonous, unpleasant	2 overlaps

as an array (score). The comparison with the most noteworthy (highest) score is taken as the best score.

I am a [baddo] when it comes to this profession.

The usage example 2 with 6 overlaps is picked as the most proper *sti*. The best usage example is mapped to its corresponding definition. The usage example 2 is mapped with the meaning of baddo^2. Consequently, the best sense for "baddo" in this context is "someone who is highly respected or seen as very good at what he/she does".

Event detection layer

The event detection layer has four components: Embedder, Event Clusterer, Event Ranker, and Event Summarizer.

The embedder

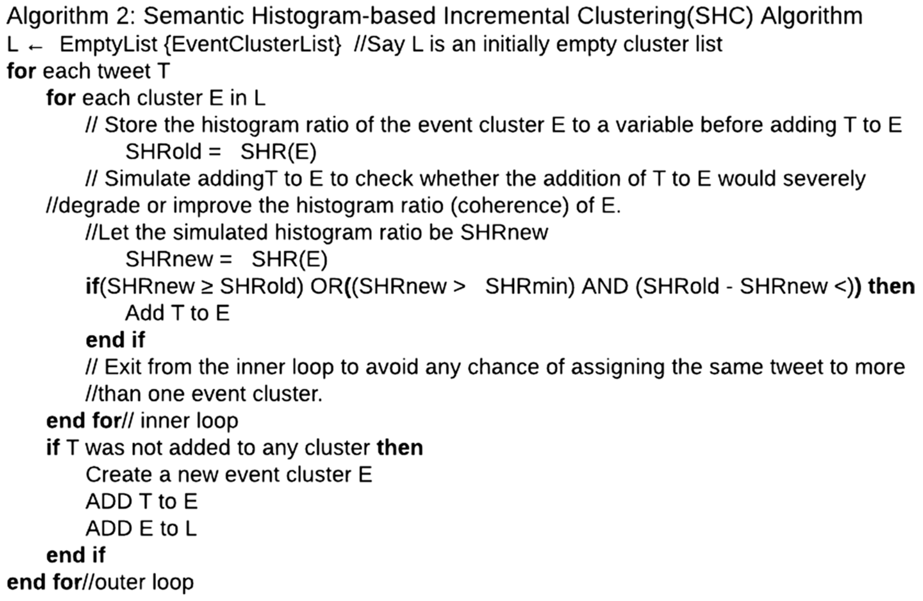
The embedder converts the enriched tweets into a vector form. This is done using a language model called sent2vec, developed by [63]. The model uses an unsupervised objective to train distributed representation of phrases/sentences. Words that are not found in the dictionary of the model are represented as zero vector, which implies that such words have no contribution to the mean vector.

The Event clusterer

The Event Clusterer of the Event Detection Layer in SMAFED performs the incremental grouping of the embedded tweets into event bins using semantic histogram-based incremental clustering [64]. Semantic histogram-based incremental clustering is a dynamic incremental method of building clusters that makes use of the semantic histogram concept to maintain a high degree of cluster coherency. New tweets are compared with each event cluster histogram to maintain the incremental creation of coherent clusters. If the addition of a new tweet will largely degrade the distribution, such a tweet is not added; otherwise, it is added. The quality of event cluster cohesiveness (semantic histogram) is measured by the ratio of similarity count above a certain similarity threshold to the total similarity count. The higher the semantic histogram ratio, the more the cluster cohesiveness.

The Event clusterer algorithm

The event clusterer algorithm (Semantic Histogram-based Incremental clustering) is presented in Algorithm 2. The computation of semantic similarities between tweets is based on how they are semantically represented. The Sent2vec model is used to obtain the semantic representation of tweets. When a new semantic similarity value between two tweets is determined, it augments the semantic comparability check (count) inside the bin (cluster) where such similarity is found. To add another tweet, the new tweet is compared against each semantic histogram cluster. On the off chance that the distribution is degraded, it is not added; else, it is added. At this stage, the issue of concept drift is implicitly taken care of because when there is an arriving tweet that does not fit into the existing clusters, a new cluster is created for it.



Algorithm 2: Semantic Histogram-based Incremental Clustering(SHC) Algorithm

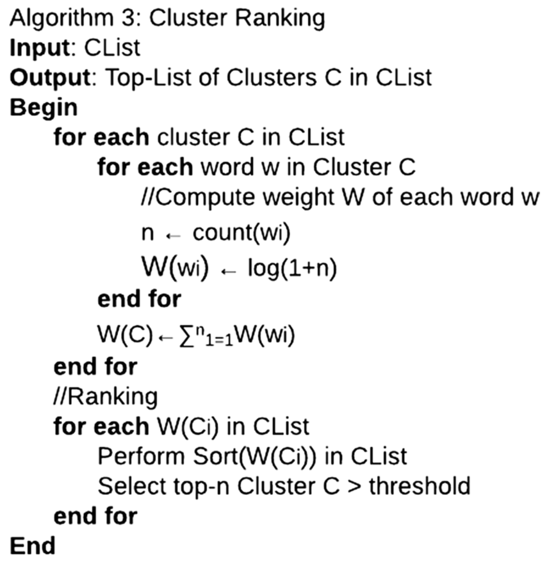
```
L ←  EmptyList {EventClusterList}  //Say L is an initially empty cluster list
for each tweet T
    for each cluster E in L
        // Store the histogram ratio of the event cluster E to a variable before adding T to E
            SHRold =   SHR(E)
        // Simulate addingT to E to check whether the addition of T to E would severely
    //degrade or improve the histogram ratio (coherence) of E.
        //Let the simulated histogram ratio be SHRnew
            SHRnew =   SHR(E)
        if(SHRnew ≥ SHRold) OR((SHRnew >   SHRmin) AND (SHRold - SHRnew <)) then
            Add T to E
        end if
        // Exit from the inner loop to avoid any chance of assigning the same tweet to more
        //than one event cluster.
    end for// inner loop
    if T was not added to any cluster then
        Create a new event cluster E
        ADD T to E
        ADD E to L
    end if
end for//outer loop
```

The Event ranker

The Event Ranking component of SMAFED orders the contents in each cluster based on the importance of the constituent words. Since the event clustering is unsupervised and the number of clusters is not known in advance, it is necessary to determine the clusters that would contribute to the representative summary. In other words, the importance of the information richness of a cluster is based on the number of important words it contains.

The Event ranker algorithm

The event ranker algorithm is implemented (as in definition T9) using Algorithm 3. The focus of the event ranker algorithm is to determine which of the detected event clusters are actual events. For each of the clusters, the weight is computed by summing up the weights of all the important words in each event cluster. The event clusters are then sorted in descending other based on their weights. Any event cluster whose weight is greater than a given threshold is considered an event.



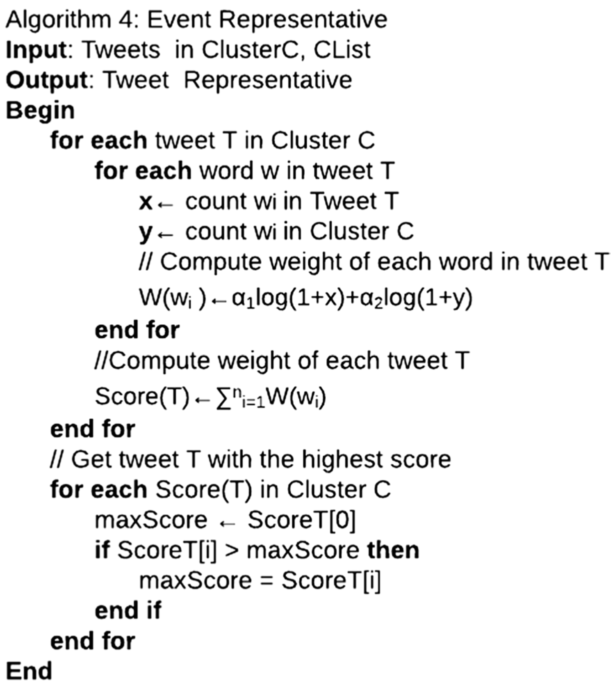
```
Algorithm 3: Cluster Ranking
Input: CList
Output: Top-List of Clusters C in CList
Begin
    for each cluster C in CList
        for each word w in Cluster C
            //Compute weight W of each word w
            n ← count(wi)
            W(wi) ← log(1+n)
        end for
        $W(C) \leftarrow \sum^{n}{}_{1=1} W(wi)$
    end for
    //Ranking
    for each W(Ci) in CList
        Perform Sort(W(Ci)) in CList
        Select top-n Cluster C > threshold
    end for
End
```

The Event summarizer

The Event Summarizer component of SMAFED finds a suitable representative summary of the candidate event cluster with a coherent and fluent summary using the extractive summary approach. Ideally, candidate event clusters should have tweets that belong to the same event, but there is a need to find a representative that can represent individual clusters. The tweet with the highest score based on its local and global importance is selected. The local significance of a word found in each tweet shows how much contribution the word makes to the central tweet concept. The global importance corresponds to the word's contribution in the subtopics formation spread over the cluster of tweets.

The Event summarizer algorithm

Algorithm 4 presents the event summary based on definition T10. The event summarizer algorithm looks at each tweet found in each event cluster to find which of the tweets in each cluster can serve as representative. In other words, which of the tweets in each event cluster can we pick and use as a summary of all the tweets in an event cluster? The algorithm answers this question by counting the frequency of each important word in a tweet and in the cluster in which the tweet appears. After that, the average local and global importance is computed. This computation is done for each important word in the tweet, and the sum is taken as the weight of the tweet in the event cluster. This is how the weight of all tweets in the event cluster is computed. The tweet with the highest score is taken as the summary of the event cluster.



```
Algorithm 4: Event Representative
Input: Tweets in ClusterC, CList
Output: Tweet Representative
Begin
    for each tweet T in Cluster C
        for each word w in tweet T
            x ← count wi in Tweet T
            y ← count wi in Cluster C
            // Compute weight of each word in tweet T
            $W(w_i) \leftarrow \alpha_1 \log(1+x)+\alpha_2 \log(1+y)$
        end for
        //Compute weight of each tweet T
        $Score(T) \leftarrow \sum^{n}_{i=1} W(w_i)$
    end for
    // Get tweet T with the highest score
    for each Score(T) in Cluster C
        maxScore ← ScoreT[0]
        if ScoreT[i] > maxScore then
            maxScore = ScoreT[i]
        end if
    end for
End
```

Evaluation experiment

In this section, we report the evaluation of the SMAFED framework. The evaluation was divided into two parts. The first part gives a detailed summary of the impact of the data enrichment layer of SMAFED, which focused on semantic analysis of SAB compared to when there is no treatment of SAB. The second part focuses on the performance of SMAFED when used for event detection from social media streams.

Experiment I: impact of the data enrichment layer of SMAFED

SMAFED was evaluated by benchmarking it with the General Social Media Feed Preprocessing Method (GSMFPM) to determine the impact of the enrichment layer of SMAFED. The difference between GSMFPM and SMAFED is highlighted in Table 8.

Table 8 Difference between GSMFPM and SMAFED

Feature	GSMFPM	SMAFED
Disambiguation	No	Yes
Abbreviation handling	No	Yes
Acronym handling	No	Yes
Inclusion of localised knowledge source	No	Yes
Spell-checking module	No	Yes

Table 9 Summary of twitter sentiment analysis training corpus and Naija-tweet dataset

S/N	Dataset	Source(s)	Total	Selected	Training/Testing
1	Twitter sentiment analysis training corpus	1. University of Michigan Sentiment Analysis on Kaggle 2. Twitter sentiment corpus by Niek Sanders	1,578,627 1,048,575 (after download)	104,857 (10%)	83,886/20,971
2	Naija-Tweets	Extracted from Nigeria origin	12,920	12,920 (100%)	10,336/2,584

Table 10 Summary of feature extraction and representation

S/N	Dataset	Unigram	Bigram
1	Twitter sentiment analysis training corpus	76,522 Top-K word (50,000)	501,026 Top-K (150,000)
2	Naija-Tweets	3,296 Top-K (3,000)	10,187 Top-K (8,000)

Dataset description

Two datasets for the first experiment were Twitter sentiment analysis training corpus and Naija-tweets. A summary of the two datasets is presented in Table 9.

Feature extraction and representation

We extracted two types of features, namely, unigram and bigram, from the datasets. The summary of the feature extraction and representation is shown in Table 10.

Global Vector for Word Representation (GloVe) was used for the feature extraction. GloVe is an unsupervised learning algorithm for obtaining word-word co-occurrence statistics from a corpus. This results in representations that showcase interesting linear structures of the word vector space. GloVe is a log-bilinear model with weighted least-squares, which combines the features of the local context window and global matrix factorization methods. The underlying intuition of the model is that the ratios of word-word co-occurrence have some form of meaning encoding potential.

Classifiers

We used supervised learning techniques for text classification, namely, multilayer perceptron (MLP) and convolutional neural networks (CNN). MLP trains on input–output pairs and models the dependencies between the inputs and outputs. CNN is a deep

Table 11 Multi-layer perceptron cross-entropy loss function for GSMFPM and SMAFED on twitter sentiment analysis training corpus

	FEATURES					
	UNIGRAM		**BIGRAM**		**UNIGRAM + BIGRAM**	
EPOCH	**SMAFED**	**GSMFPM**	**SMAFED**	**GSMFPM**	**SMAFED**	**GSMFPM**
1	0.5478	0.5499	0.5612	0.5729	0.5535	0.5405
2	0.4911	0.5132	0.4405	0.4872	0.5004	0.5192
3	0.4537	0.4658	0.3909	0.4427	0.471	0.5087
4	0.4045	0.4147	0.3015	0.3528	0.4047	0.4584
5	0.3741	0.3762	0.2181	0.3177	0.3484	0.3698

Table 12 Four-Layer convolutional neural network cross-entropy loss function for GSMFPM and SMAFED on twitter sentiment analysis training corpus

	Convolution layer							
	1-LAYER		**2-LAYER**		**3-LAYER**		**4-LAYER**	
EPOCH	**SMAFED**	**GSMFPM**	**SMAFED**	**GSMFPM**	**SMAFED**	**GSMFPM**	**SMAFED**	**GSMFPM**
1	0.5852	0.4369	0.636	0.7843	0.6279	0.7762	0.5584	0.7067
2	0.4761	0.3928	0.5399	0.6232	0.4927	0.576	0.4713	0.5546
3	0.4213	0.3725	0.4674	0.5162	0.4434	0.4922	0.4263	0.4751
4	0.3622	0.3556	0.4195	0.4261	0.3943	0.4009	0.3836	0.3902
5	0.3026	0.3397	0.3797	0.4168	0.3515	0.3886	0.332	0.3691
6	0.2241	0.3255	0.3403	0.4417	0.299	0.4004	0.277	0.3784
7	0.1864	0.3002	0.3045	0.4183	0.2736	0.3874	0.2501	0.3639
8	0.1496	0.2892	0.2576	0.3972	0.2425	0.3821	0.2169	0.3565

learning architecture model that aims to learn higher-order features present in data through convolutions.

Experiment I: result and discussion

The proposed SMAFED was benchmarked with the General Social Media Feed Pre-processing method (GSMFPM) by testing their impact on two classifiers. The essence of assessing the impact of the general pre-processing method – GSMFPM and the proposed SMAFED on the classifiers was to determine whether analysing SAB terms and resolving ambiguity in SAB in social media streams can affect event detection results. To do this, we compared the cross-entropy loss function of the classifiers (MLP and CNN) when GSMFPM and SMAFED were used. The comparison based on the loss function indicates how good a classifier accurately predicts the expected outcome. The cross-entropy result for sentiment classification of Twitter sentiment analysis training corpus for Multilayer Perceptron and Convolutional Neural Network with five epochs and eight epochs, respectively, are shown in Tables 11 and 12. At the same time, the Naija-tweets dataset is presented in Tables 13 and 14.

Table 11 presents the multilayer perceptron cross-entropy loss function for GSMFPM and SMAFED on Twitter Sentiment Analysis Training Corpus based on Unigram, Bigram, and Unigram + Bigram. The table shows that SMAFED

Table 13 Multi-layer perceptron cross-entropy loss function for GSMFPM and SMAFED on Naija-tweets dataset

	Features					
	UNIGRAM		BIGRAM		UNIGRAM + BIGRAM	
Epoch	SMAFED	GSMFPM	SMAFED	GSMFPM	SMAFED	GSMFPM
1	0.366	0.3943	0.2911	0.3927	0.2632	0.3527
2	0.2552	0.2712	0.2362	0.2661	0.1908	0.2583
3	0.2108	0.2223	0.2009	0.2149	0.1785	0.2003
4	0.1688	0.2172	0.1665	0.2035	0.1216	0.1814
5	0.1534	0.2102	0.1759	0.2058	0.1112	0.2007

Table 14 Four-Layer convolutional neural network cross-entropy loss function for GSMFPM and SMAFED on Naija-tweets dataset

	Convolution layer							
	1-LAYER		2-LAYER		3-LAYER		4-LAYER	
EPOCH	SMAFED	GSMFPM	SMAFED	GSMFPM	SMAFED	GSMFPM	SMAFED	GSMFPM
1	2.1429	2.351	0.8408	0.9112	0.7448	0.7953	0.7055	0.7676
2	0.7991	0.8535	0.6119	0.6663	0.6419	0.6949	0.6336	0.6747
3	0.4748	0.5268	0.5363	0.5783	0.6036	0.7111	0.6344	0.6643
4	0.3693	0.4804	0.4152	0.5161	0.5511	0.6353	0.612	0.6609
5	0.2316	0.3457	0.2479	0.3622	0.3897	0.4465	0.559	0.6303
6	0.1429	0.2999	0.1266	0.2736	0.2231	0.2822	0.4445	0.4895
7	0.0878	0.1489	0.0873	0.1484	0.1469	0.1852	0.2588	0.3581
8	0.0501	0.0852	0.052	0.0771	0.1031	0.1431	0.178	0.1982

outperformed GSMFPM with respect to matching between the predicted and the actual sentiment by returning lower loss function values. It should also be noted that the lowest loss function in each of the unigram, bigram, and unigram + bigram of both approaches was obtained at epoch_5, meaning that the more the number of epochs, the better the performance of the classifier.

The Cross-Entropy Loss Function of CNN with kernel size = 3 and one-four convolution layers using eight epochs for SMAFED compared with GSMFPM on Twitter Sentiment Analysis Training Corpus are presented in Table 12. From the table, it can be deduced that the pre-processing coupled with the data enrichment components of SMAFED outperformed GSMFPM in matching the predicted and the actual sentiment. It should also be noted that the loss function of the first layer of CNN cross-entropy for both approaches is lower than that of other layers.

The Cross-Entropy Loss Function of Multilayer Perceptron with Unigram, Bigram, and Unigram + Bigram features using five epochs for SMAFED compared with GSMFPM on Naija-Tweets dataset are presented in Table 13. The table shows that SMAFED outperformed GSMFPM with respect to matching between the predicted and the actual sentiment. As noted with Twitter Sentiment Analysis Training Corpus and Naija-Tweets, the lowest loss function in both approaches' unigram, bigram, and unigram + bigram was obtained at epoch_5. The outcome of the experiment revealed that the proposed SMAFED performed better than the general pre-preparing technique. It also shows an improvement in terms of accuracy (on the obtained results). This underscores the significance of using a local vocabulary in pre-processing social media feeds to disambiguate the noisy terms that are contained in the social media feeds from a specific origin.

Table 14 presents the Cross-Entropy Loss Function of CNN with kernel size = 3 and one-four convolution layers using eight epochs for SMAFED compared with GSMFPM on the Naija-Tweets dataset. From the table, the performance of pre-processing coupled with data enrichment components of SMAFED outperformed GSMFPM with respect to matching the predicted and the actual sentiment. It should also be noted that the loss function of the first layer of CNN cross-entropy for both approaches is lower than that of other layers.

SMAFED efficiency

The performance of the SMAFED framework was assessed using run-time performance metrics [65] to measure the efficiency and practicability of the framework. We implemented the proposed event detection method using Python (v 3.7). We used Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz processor for testing, with 12 GB RAM and a 64-bit Windows 10 operating system. The framework was deployed on the cloud using a 4 GB Docker Droplet hosted on DigitalOcean services.

The tweets used for the prototype implementation were sourced from tit took 5 secondshe Nigeria location. It was found out that the average number of tweets from Nigeria per minute is 45 without the application of a filter. This shows that tweeting in Nigeria is very small compared to an overall average of 350,000 tweets per minute. The part of the processing that took the longest time to process was spell checking. Before the clustering stage, it took 5 s to pre-process and enrich 40 tweets. The average processing time for each tweet is about 0.125 s. This is well within the limits needed to manage the estimated average of 1 tweet from Nigeria. In an improbable circumstance, where all tweets are recognized as events, the framework will have the option to handle eight times the normal volume of the tweets from Nigeria origin. With SMAFED, the life span of an event cluster is 4 days, after which it is deleted. This assumption was due to the fact that the potential value of data lies in its freshness. Choosing a lifespan of 4 days for an event cluster was done so as to: 1) avoid the clogging of the memory, 2) maintain a limited number of clusters in memory, and 3) limit the number of comparisons to be made. SMAFED efficiency in terms of tweet streaming from Nigeria origin and pre-processing is depicted in Fig. 4.

Experiment II: accuracy of SMAFED

This experiment aims to determine how well the SMAFED can detect events in social media streams. Three metrics, including Precision, Recall and F-measure, were used to benchmark SMAFED with other existing frameworks. These metrics are regular assessment measurements for event detection techniques [1]. Precision alludes to the number of actual events detected. Recall gives the actual similar event level that the framework can identify, and F-measure speaks to the harmonic mean of Precision and Recall. The formulae for the metrics used to ascertain event detection accuracy are given as follows:

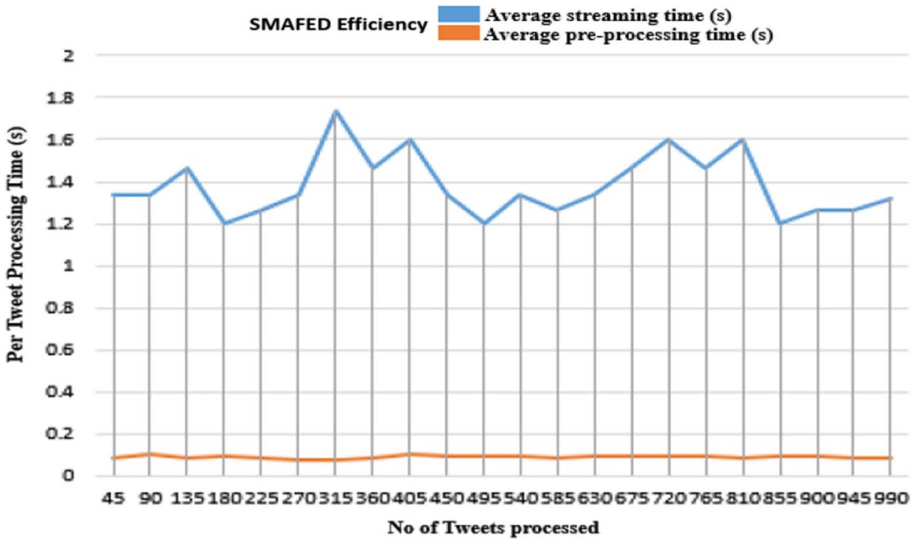


Fig. 4 SMAFED Evaluation. The average processing time for each tweet is 0.125 s

Table 15 Final dataset

Type	Number of Tweets
Event	82,887
Non-Event	142,652
Total	225,554

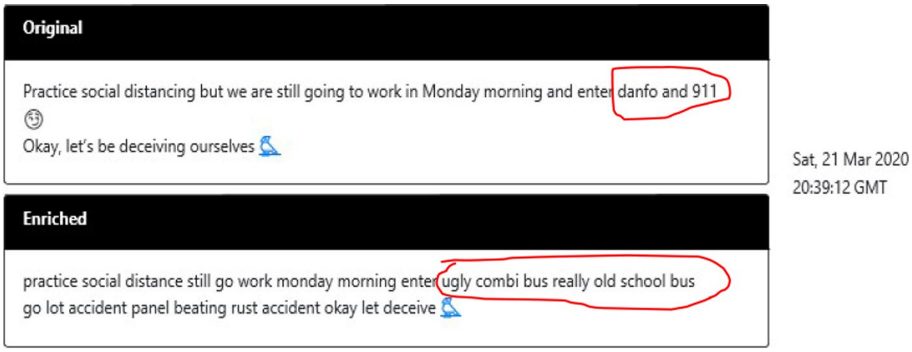
$$Precision = \frac{no\ of\ similar\ event\ detected}{total\ no\ of\ event\ detected}$$

$$Recall = \frac{no\ of\ similar\ event\ detected}{total\ no\ of\ actual\ event}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

Dataset description

To evaluate SMAFED, tweet IDs and the event relevance judgment (which is made available based on Twitter's terms of service) provided by [66] were used to obtain the tweet dataset. This was done using Tweepy with Twitter REST API. All 152 900 relevant tweets could not be extracted because some user accounts had been deleted and were no longer accessible. A total of 82,887 labelled tweets and additional 142,652 irrelevant tweets were collected. The distribution of the final dataset used by SMAFED for evaluation is presented in Table 15.



Original

Practice social distancing but we are still going to work in Monday morning and enter danfo and 911
😏
Okay, let's be deceiving ourselves 🐦

Sat, 21 Mar 2020
20:39:12 GMT

Enriched

practice social distance still go work monday morning enter ugly combi bus really old school bus go lot accident panel beating rust accident okay let deceive 🐦

Fig. 5 A snapshot of the Original and the Enriched Tweet

Experiment II: result and discussion

The section presents the results of the enrichment layer of SMAFED and the evaluation of SMAFED as benchmarked with existing frameworks in terms of accuracy. The final result of the enrichment stage is shown in Fig. 5.

After the data enrichment stage, with a typical result presented in Fig. 5, there is event clustering, ranking and summarisation. The enriched tweets from the enrichment layer are used as input to the event detection module to detect events from tweets. This module has four sub-modules: embedding, clustering, ranking, and summarisation. Sent2vec model vectorizes cleaned and enriched tweets. The Sent2vec model is wrapped in class Sent2VecWrapper and has a method "vectorize_sentences," which returns an array of vectorized sentences. The model is downloaded from the cloud service, DigitalOcean. Vectorized tweets are clustered with a semantic histogram clustering algorithm. The event ranking phase follows this. This task is implemented in the "Ranker" class. The last stage of the event detection stage is an event summary involving cluster summary computation. The resulting sample of the event detection stage, which includes event clustering, ranking and summarisation, is presented in Table 16.

The section also presents a report on the accuracy of the event detection by SMAFED as compared with Locality Sensitive Hashing (LSH), Cluster Summary (CS), Entity-based approach and the Repp framework. The difference between SMAFED and four event detection approaches is presented in Table 17.

The results include the Precision, Recall, and F-measure for each approach. The result is presented in Table 18 and depicted by the line graph in Fig. 6.

Table 18 shows the results of four different approaches compared with SMAFED. A cluster is considered a candidate event for the two baselines (LSH and CS) if it contains more than 30 tweets. The entropy-based method used 75 and above tweets with the best run, [17] used 10 + tweets with a mean over 20 runs, and SMAFED considered clusters weight > 100 threshold. Out of 120 million tweets available in the Event2012, [17] discovered 152,900 tweets that can be considered event tweets. Instead of using 120 million unlabeled tweets, SMAFED focused on the relevant tweets (150,000 +) used as ground truth for benchmarking purposes. However, since tweets may be deleted or users can delete their own Twitter account, making them unavailable, the total 152,900 relevant

Table 16 Event Clustering, Ranking, and Summarisation

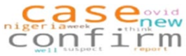
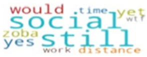
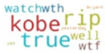
ID	Tweet		# of Tweets	Word Cloud
4695 1727.62	@Fmohnigeria has confirmed 10 new cases of #COVID19 in #Nigeria Of the 10 new cases, 3 are in Federal Capital Territory, 7 are in Lagos 9 out of the 10 cases have travel history outside Nigeria in the last one week. The 10th case is a close contact of a confirmed case. Sad	Sat, 21 Mar 2020 14:00:06 GMT	102	
	@Fmohnigeria has confirmed 10 new cases of #COVID19 in #Nigeria Of the 10 new cases, 3 are in Federal Capital Territory, 7 are in Lagos 9 out of the 10 cases have travel history outside Nigeria in the last one week. The 10th case is a close contact of a confirmed case!!	Sat, 21 Mar 2020 13:37:03 GMT		
	FLASH: The Federal Ministry of Health has just confirmed 10 new cases of #coronavirus in Nigeria. Of the 10 new cases, 3 are in Federal Capital Territory, Abuja and 7 are in Lagos state.	Sat, 21 Mar 2020 13:45:46 GMT		
5283 1386.10	Who would have predicted a time like this when social spacing will be preferable to social gathering. A Saturday where NKANBẸ has swallowed OWANBẸ. Pandemic and pandemonium are two terrible brothers from same mother. None of them is good	Sat, 21 Mar 2020 19:48:20 GMT	89	
	@Osi-Suave This could be a template. If a total shutdown wouldn't work, let's then pursue massive spacing and social distancing. A massive re-orientation campaign needs to be done for people to change their social practices	Sat, 21 Mar 2020 17:18:43 GMT		
	Social distancing not working. Taxis still cram 5-6 passengers even in Abuja. Only middle-class people are distancing. We need extra ideas	Sat, 21 Mar 2020 11:00:53 GMT		
4379 1224.72	Even I don't follow basketball that much, two names are pop-up in my head each time I remember the sport. LeBron James and Kobe Bryant R.I.P Kobe Bryant	Sun, 26 Jan 2020 20:26:52 GMT	127	
	Life is too short. I don't watch basketball that much, but I've heard the name Kobe Bryant several times like with lebron James. May his soul rip	Sun, 26 Jan 2020 21:07:56 GMT		
	A special moment for Kobe Bryant yesterday at the Grammy 2020 what a sad moment RIP to Kobe and many on the helicopter	Mon, 27 Jan 2020 21:07:56 GMT		

Table 17 Comparison of smafed against other Frameworks in terms of scope, coverage and approach

Approach	Handles		Includes		Clustering	Ranking	Summarisation
	Concept Drift	SAB	Disambiguation	Localised Knowledge Source			
CS	No	No	No	No	Similarity Score	No of tweets	No
LSH	No	No	No	No	Hash function	Shannon Entropy	No
Entity-based	No	No	No	No	TF-IDF	No of tweets	No
Repp Framework	No	No	No	No	Cosine distance	No of tweets	No
SMAFED	Yes	Yes	Yes	Yes	Semantic similarity	Weighting scheme	Yes

tweets could not be all extracted. A total of 82,887 labelled tweets were downloaded. To introduce noise to the dataset and assess the performance of SMAFED, an additional

Table 18 The results of LSH, CS, Entity-based approach, Repp Framework and SMAFED on Event2012 Twitter Dataset

Approach	References	Precision	Recall	F-Measure
LSH	[14]	382/1340 (0.285)	156/506 (0.308)	0.296
CS	[15]	53/1097 (0.048)	32/506 (0.063)	0.054
Entity-based	[16]	181/586 (0.302)	159/506 (0.310)	0.306
Repp framework	[17]	271/300 (0.901)	112/150 (0.749)	0.818
SMAFED	Authors (2021)	**296/321 (0.922)**	**119/150 (0.793)**	**0.853**

The values in bold indicate the highest value of the precision, Recall, and F-Measure in the approaches compared

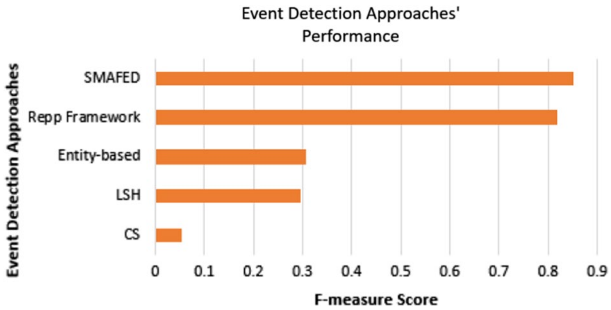


Fig. 6 The comparison of F-Measure score for event detection approaches

142,652 irrelevant tweets were collected from the pool of irrelevant tweets in the 120 million tweets.

All the four approaches compared with SMAFED, except for the framework proposed by [15] (LSH), used the number of tweets in a cluster as criteria for it to be considered as a relevant event cluster. This method does not perform well as clusters with more tweets may be less informative [18, 67]. The ranking algorithm used in SMAFED focused on the important constituents of each cluster. For a cluster to be considered as an event, it must have a weight > 100. This weight was chosen after testing for different weight size (50, 100, 150) and having in mind the number of event clusters in the ground truth dataset. None of the existing approaches compared with SMAFED in this study used a tweet representative to summarize clusters. The baseline approaches reported in [66], along with [17], used crowdsourcing for categorization. SMAFED used a weighting approach for each tweet in a cluster to determine a representative tweet. The tweet with the highest weight in each event cluster is selected as a tweet representative. Choosing a tweet representative in each event cluster gives a quick summary which can be used for reporting what is happening.

SMAFED can adapt to changes in the social media streams as it does not use any restriction in streaming tweets except eliminating retweets, tweets with more than three hashtags and or more than two URLs. This was done to eliminate spam tweets which may adversely affect the event detection process. Unlike using an already built classifier to filter tweets of interest from social media streams during data collection, an approach used by [18], such approach would not be able to adapt to changes in the social media streams as it requires a continuous update of the classifier to incorporate changes in the social media stream.

Even though Twitter has been recognized as an important resource for event detection, the scarcity of publicly available dataset coupled with different parameter settings makes it difficult to judge against existing approaches [68]. Since Event2012 Twitter dataset is the only publicly available event detection dataset with annotation, it was used as the ground-truth dataset for evaluating SMAFED. The result of the evaluation was compared with existing works that made use of the same dataset to evaluate event detection. The four existing works that were compared with SMAFED include two baselines used to generate the collection of Event2012 Twitter dataset (Locality Sensitive Hashing (LSH) and Cluster Summary (CS) proposed by [15] and [16], respectively), the entity-based approach proposed by [17], and a framework for event detection proposed by [18].

From Fig. 6, it can be deduced that SMAFED performed better than the existing event detection approaches. SMAFED also has the highest value for F-measure compared to existing methods for event detection. This indicates that both precision and Recall are reasonably high and that the SMAFED has an excellent ability to detect events in social media streams. SMAFED was measured closely against the best event detection framework (amongst the four event detection approaches compared with SMAFED) proposed by [18]. Comparing event detection approaches may seem difficult in the sense that there are slight differences in the parameters or criteria which were earlier pointed out. SMAFED was measured closely against the best event detection framework (amongst the four event detection approaches SMAFED was compared with) proposed by Repp (2016). Even though the number of tweets (categorized as irrelevant) in Events2012 Twitter dataset added to the available 82,887 tweets (relevant) was more than that of Repp's framework, SMAFED still performed better.

Conclusion and further work

In this paper, a Social Media Analysis Framework for Event Detection (SMAFED) that can analyse the rich but hidden knowledge in social media streams to improve the accuracy of event detection was presented. SMAFED, as proposed in this paper, serves as an improvement on the existing event detection approaches with better metric scores in terms of Precision (0.922), Recall (0.793) and F-Measure (0.853). In addition, an evaluation experiment was carried out to determine the impact of the data enrichment layer by benchmarking SMAFED with GSMFPM. The cross-entropy result for sentiment classification of Twitter sentiment analysis training corpus and Naija-Tweets dataset for Multilayer Perceptron and Convolutional Neural Network with five epochs and eight epochs, respectively, showed that SMAFED outperformed GSMFPPM. This paper contributes to big data analytics research, particularly event detection in social media streams. More precisely, it caters for the observed limitations of existing event detection approaches by (1) performing semantic analysis of SAB terms along with ambiguity in their usage. This leads to better comprehension and interpretation of social media streams noisy terms; (2) evolving SABDA to disambiguate ambiguous SAB terms; (3) creating an integrated knowledge base to facilitate semantic analysis of noisy terms in social media streams.

In this paper, SMAFED used only social media stream texts for event detection in social media streams. The integration of images and correlated text from social media streams will further strengthen the event detection result. While Twitter is a well-known research data source, exploring and or combining it with other social media sources will lead to more events being detected and harmonisation of event detection results. This is still open to further research as few approaches have exploited this medium.

Abbreviations

ATSED	Automatic Targeted-domain Spatio-temporal Event Detection
BEE	Bursty Event dEtection
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CNN	Convolutional Neural Networks
CSV	Comma separated value
EDCoW	Event Detection with Clustering of Wavelet-based Signals
ESA	Explicit Semantic Analysis
GloVe	Global Vector for Word Representation
GRU	Gated recurrent unit
GSMFPM	General Social Media Feed Pre-processing Method
HDP	Hierarchical Dirichlet Process
IKB	Integrated knowledge base
JSON	JavaScript Object Notation
KLD	Kullback–Leibler divergence
LDA	Latent Dirichlet Allocation
LSH	Locality sensitive hashing
MLP	Multilayer perceptron
NMF	Non-negative matrix
OPTICS	Ordering points to identify the clustering structure
SAB	Slang, Acronym, and Abbreviation
SABDA	Slangs, acronyms, and abbreviations Disambiguation Algorithm
SMAFED	Social Media Analysis Framework for Event Detection
TF	Term frequency
TF-IDF	Term Frequency—Inverse Document Frequency
TRCM	Transaction-based rule change mining
VDEH	Variable dimensional extendible hash

Acknowledgements

The research was supported by the authors' institutions: Cape Peninsula University of Technology, South Africa; Federal University Lokoja, Kogi State, Nigeria; Landmark University, Omu-Aran, Kwara State, Nigeria.

Author contributions

TK conceptualized the research methodology, curated the data, wrote the codes, wrote the original draft, reviewed and edited the paper. OD validated resources, reviewed and edited the paper, and supervised the paper's writing. AAA reviewed and edited the paper and supervised the paper's writing. All authors read and approved the final manuscript.

Funding

This work was supported by the National Research Foundation of South Africa under Grant No. STGR 180414320796, and Grant No. 129255.

Availability of data and materials

Twitter Sentiment Analysis Corpus is available at http://thinknook.com/Twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/.

Declarations

Ethics approval and consent to participate

Not Applicable.

Consent for publication

All authors have read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 15 December 2021 Accepted: 27 June 2022
Published online: 12 July 2022

References

1. Panagiotou N, Katakis I, Gunopulos D. Detecting events in online social networks: definitions, trends and challenges. In: Michaelis S, editor. Solving large scale learning tasks: challenges and algorithms. Cham: Springer; 2016. p. 42–84.

2. Win SSM, Aung TN. Automated text annotation for social media data during natural disasters. Adv Sci Technol Eng J. 2018;3(2):119–27.
3. Olsson T, Jarusriboonchai P, Wozniak P, Paasovaara S, Vaananen K, Lucero A. Technologies for enhancing collocated social interaction: review of design solutions and approaches. Comput Supported Coop Work (CSCW). 2020;29:29–83. https://doi.org/10.1007/s10606-019-09345-0.
4. Carbezudo MAS, Pardo TAS. Exploring classical and linguistically enriched knowledge-based methods for sense disambiguation of verbs in Brazilian Portuguese news texts. Nat Lang Process. 2017;59:83–90.
5. Gutierrez-Vazquez Y, Vazquez S, Montoyo A. A semantic framework for textual data enrichment. Expert Syst Appl. 2016;57:248–69.
6. Alkhatlan A, Kalita J, Alhaddad A. Word sense disambiguation for Arabic exploiting WordNet and word embedding. Procedia Comput Sci. 2018;142:50–60. https://doi.org/10.1016/j.procs.2018.10.460.
7. Kolajo T, Daramola O, Adebiyi A, Seth A. A framework for pre-processing of social media feeds based on integrated local knowledge base. Inf Process Manag. 2020;57(6):102348. https://doi.org/10.1016/j.ipm.2020.102348.
8. Atefeh F, Khreich W. A survey of techniques for event detection in Twitter. Comput Intell. 2015;31(1):132–64.
9. Jain VK, Kumar S, Fernandes SL. Extraction of emotions from multilingual text using intelligent text processing and computational linguistics. J Comput Sci. 2017;21:316–26. https://doi.org/10.1016/j.jocs.2017.01.010.
10. Rao D, McNamee P, Dredze M. Entity linking: finding extracted entities in a knowledge base. In: Poibeau T, Saggion H, Piskorski J, Yangarber R, editors. Multi-source, Multilingual information extraction and summarization. Theory and Applications of Natural Language Processing. Heidelberg: Springer; 2013. p. 93–115.
11. Singh T, Kumari M. Role of text pre-processing in Twitter sentiment analysis. Procedia Comput Sci. 2016;89:549–54. https://doi.org/10.1016/j.procs.2016.06.095.
12. Zhan J, Dahal B. Using deep learning for short text understanding. Journal of Big Data. 2017;4:34. https://doi.org/10.1186/s40537-017-0095-2.
13. Katragadda S, Benton R, Raghavan V. Framework for real-time event detection using multiple social media sources. Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS). Waikoloa, Hawaii, 2017. p. 1716–1725 https://doi.org/10.24251/HICSS.2017.208
14. Xia C, Schwartz R, Xie K, Krebs A, Langdon A, Ting J, Naaman, M. CityBeat: Real-time social media visualisation of hyper-local city data. Proceedings of the 23rd International World Wide Web Conference Committee (IW3C2). Seoul, South Korea. 2014. p. 167–170. https://doi.org/10.1145/2567948.2577020
15. Petrovic S, Osborne M, Lavrenko V, Streaming first story detection with application to Twitter. Proceedings of Human Language Technologies: The Annual Conference of American Chapter of the Association for Computational Linguistics Los Angeles. CA, USA. 2010;2010:181–9.
16. Aggarwal CC, Subbian K. Event detection in social streams. Proceedings of the SIAM International Conference on Data Mining. California, USA, 2012. p. 624–635.
17. McMinn AJ, Jose AM. Real-time entity-based event detection for Twitter. In: Mothe J, editor. Experimental IR Meets Multilinguality, Multimodality, and Interaction. Cham: Springer; 2015. p. 65–77.
18. Repp QK. Event detection in social media: Detecting news event from the Twitter stream in real-time (Master's thesis). Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, 2016.
19. Boushaki SI, Kamel N, Bendjeghaba O. High-dimensional text datasets clustering algorithm based on cuckoo search and latent semantic indexing. J Inf Knowl Manag. 2018;17(3):1–24. https://doi.org/10.1142/S0219649218500338.
20. Weng J, Lee BS. Event detection in Twitter. ICWSM. 2011;11:401–8.
21. Zubiaga A, Spina D, Amigó E, Gonzalo J. Towards real-time summarization of scheduled events from Twitter streams. Proceedings of the 23rd ACM Conference on Hypertext and Social Media. Milwaukee, WI, USA. 2012. p. 319–320.
22. Lee C. Mining Spatio-temporal information on microblogging streams using a density-based online clustering method. Expert Syst Appl. 2012;39(10):9623–41.
23. Abdelhaq H, Sengstock C, Gertz M. EvenTweet: Online localized event detection from Twitter. Proc VLDB Endow. 2013;6(12):1326–9. https://doi.org/10.14778/2536274.2507.
24. Abhik D, Toshniwal F. Sub-event detection during natural hazards using features of social media data. Proceedings of 22nd International Conference on World Wide Web New York, NY: ACM. 2013. https://doi.org/10.1145/2487788.2488046.
25. Fuchs G, Andrienko N, Andrienko G, Bothe S, Stange H. Tracing the German centennial flood in the stream of tweets: First lessons learned. Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, ACM. GEOCROWD '13. Orlando, FL, USA, 2013. p. 31–38.
26. Elsawy E, Mokhtar M, Magdy W. TweetMogaz v2: Identifying new stories in social media. CIKM'14, Proceedings of the 23rd ACM International Conference on Information and Knowledge Management Shanghai, China, 2014. p. 2042–2044.
27. Dong X, Mavroeidis D, Calabrese F, Frossard P. Multiscale event detection in social media. Data Min Knowl Disc. 2015;29(5):1374–405. https://doi.org/10.1007/s10618-015-0421-2.
28. Li J, Wen J, Tai Z, Zhang R, Yu W. Bursty event detection from microblog: a distributed and incremental approach. Concurr Comput Pract Exp. 2016;28(11):3115–30. https://doi.org/10.1002/cpe.3657.
29. Pohl D, Bouchachia A, Hellwagner H. Online indexing and clustering of social media data for emergency management. Neurocomputing. 2016;172:168–79. https://doi.org/10.1016/j.neucom.2015.01.084.
30. Hassan M, Orgun MA, Schwitter R. Real-time event detection from the Twitter data stream using the TwitterNews+ framework. Inf Process Manage. 2019;56(3):1146–65. https://doi.org/10.1016/j.ipm.2018.03.001.
31. Fedoryszak M, Frederick B, Rajaram V, Zhong C. Real-time event detection on social data streams. 25th ACM SIKDD Conference on Knowledge Discovery and Data Mining (KDD' 19) New York, NY: ACM, 2019 (9pgs). doi: https://doi.org/10.1145/3292500.3330689
32. Amato F, Moscato V, Picariello A, Sperli G. Extreme events management using multimedia social networks. Futur Gener Comput Syst. 2019;94:444–52.
33. Cai T, Li J, Mian A, Li R, Sellis T, Yu JS. Target-aware holistic influence maximization in spatial social networks. IEEE Trans Knowl Data Eng. 2020. https://doi.org/10.1109/TKDE.2020.3003047.

34. Kumar S, Liuy H, Mehta S, Subramaniam LV. Exploring a scalable solution to identifying events in noisy Twitter streams. 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '15. Paris, France, 2015. p. 496–499.
35. Lu G, Mu Y, Gu J, Kouassi FAP, Lu C, Wang R, Chen A. A hashtag-based sub-event detection framework for social media. Comput Electr Eng. 2021;94: 107317. https://doi.org/10.1016/j.compeleceng.2021.107317.
36. Xu S, Li S, Huang W, Wen R. Detecting spatio-temporal traffic events using geosocial media data. Comput Environ Urban Syst. 2022;94: 101797.
37. Becker H. Identification and characterization of events in social media. Ph.D. Dissertation. Columbia University, USA. Advisor(s) Gravano L. 2011; 197pgs. Order Number: AAI3480999.
38. Xu J, Lu T, Compton R, Allen D. Civil unrest prediction: A Tumblr-based exploration. In: Kennedy WG, Agarwal N, Yang SJ, editors. SBP 2014, LNCS 8393. Cham: Springer; 2014. p. 403–11.
39. Hua T, Chen F, Zhao L, Lu C, Ramakrishnan N. Automatic targeted domain spatiotemporal event detection in Twitter. GeoInformatica. 2016;20(4):765–95. https://doi.org/10.1007/s10707-016-0263-0.
40. Schubert E, Weiler M, Kriegel, H. SPOTHOT: Scalable detection of geo-spatial events in large textual streams. SSDBM Budapest, Hungary, 2016. p. 1–8. https://doi.org/10.1145/2949689.2949699
41. Modha S, Joshi K. Performance analysis of clustering algorithm in sensing microblog for smart cities. In S. C. Satapathy et al. (Eds.), Advances in Intelligent Systems and Computing. Proceedings of the International Congress on Information and Communication Technology. Singapore: Springer, 2016;439:467–475. https://doi.org/10.1007/978-981-10-0755-2_50
42. Shukla A, Aggarwal D, Keskar, R. A Methodology to detect and track breaking news on Twitter. Proceedings of the 9th Annual ACM India Conference. Gandhinagar, India. 2016. p. 133–136. https://doi.org/10.1145/2998476.2998491
43. Srijith PK, Hepple M, Bontcheva K, Preotiuc-Pietro D. Sub-story detection in Twitter with hierarchical Dirichlet process. Inf Process Manage. 2017;53(4):989–1003. https://doi.org/10.1016/j.ipm.2016.10.004.
44. Walther M, Kaisser M. Geo-spatial event detection in the Twitter stream. In P. Serdyukov et al. (Eds.), Advances in Information Retrieval. ECIR 2013, LNCS 7814 Berlin, Heidelberg: Springer, 2013. p. 356–367. https://doi.org/10.1007/978-3-642-369735_30.
45. Zhou X, Chen L. Event detection over Twitter social media streams. VLDB J. 2014;23(3):38–40. https://doi.org/10.1007/s00778-013-0320-3.
46. Adedoyin-Olowe M, Gaber MM, Dancausa CC, Stahl F. Extraction of unexpected rules from Twitter hashtags and its application to sports events. 13th International Conference on Machine Learning and Applications Detroit, MI: IEEE, 2014. p. 207–212. https://doi.org/10.1109/ICMLA.2014.38.
47. Hayashi K, Maehara T, Toyoda M, Kawarabayash K. Real-time top-k topic detection on Twitter with topic hijack filtering. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, Australia, 2015. p. 417–426. https://doi.org/10.1145/2783258.2783402
48. Gaglio S, Rea GL, Morana M. A framework for real-time Twitter data analysis. Comput Commun. 2016;73:236–42. https://doi.org/10.1016/j.comcom.2015.09.021.
49. Zeppelzauer M, Schopfhauser D. Multimodal classification of events in social media. Image Vis Comput. 2016;53:45–56. https://doi.org/10.1016/j.imavis.2015.12.004.
50. Wang Y, Neves L, Metze F. Audio-based multimedia event detection using deep recurrent neural networks. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016. p. 2742–2746. https://doi.org/10.1109/ICASSP.2016.7472176.
51. Lan Z. Towards usable multimedia event detection. PhD Thesis, Carnegie Mellon University, 2017.
52. Cui W, Wang P, Du Y, Chen X, Guo D, Li J. An algorithm for event detection based on social media data. Neurocomputing. 2017;254:53–8. https://doi.org/10.1016/j.neucom.2016.09.127.
53. Zhang Z, He Q, Gao J, Ni M. A deep learning approach for detecting traffic accidents from social media data. Transp Res Part C. 2018;86:580–96. https://doi.org/10.1016/j.trc.2017.11.027.
54. Mossie Z, Wang JH. Vulnerable community identification using hate speech detection on social media. Inf Process Manage. 2020;57(3): 102087.
55. McCreadie R, Macdonald C, Ounis I, Osborne M, Petrovic S. Scalable distributed event detection for Twitter. 2013 IEEE International Conference on Big Data. Silicon Valley, CA, USA, 2013. p. 543–549.
56. Kaleel SB, Almeshary M, Abhari A. Event detection and trending in multiple social networking sites. Proceedings of the 16th Communications & Networking Symposium. San Diego, CA, USA, 2013:5.
57. Musaev A, Wang D, Shridhar S, Lai C, Pu C. Toward a real-time service for landslide detection: Augmented explicit semantic analysis and clustering composition approaches. 2015 IEEE International Conference on Web Services New York, NY, USA, 2015. p. 511–518. https://doi.org/10.1109/ICWS.2015.74.
58. Tonon A, Cudré-Mauroux P, Blarer A, Lenders V, Motik B. ArmaTweet: Detecting events by semantic tweet analysis. In: Blomqvist E, Maynard D, Gangemi A, Hoekstra R, Hitzler P, Hartig O, editors. The Semantic Web. Cham: Springer; 2017. p. 138–53.
59. Romero S, Becker K. A framework for event classification in tweets based on hybrid semantic enrichment. Expert Syst Appl. 2019;118:522–38. https://doi.org/10.1016/j.eswa.2018.10.028.
60. Sun X, Liu L, Ayorinde A, Pannerselvam J. ED-SWE: event detection based on scoring and word embedding in online social networks for the internet of people. Digital Commun Net. 2021. https://doi.org/10.1016/j.dcan.2021.03.006.
61. Stieglitz S, Mirbabaie M, Rossa B, Neuberger C. Social media analytics - Challenges in topic discovery, data collection, and data preparation. Int J Inf Manage. 2018;39:156–68.
62. Lesk M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. Proceedings of SIGDOC'86. New York, NY, USA: ACM, 1986. p. 24–26.
63. Pagliardini M, Gupta P, Jaggi M. Unsupervised learning of sentence embeddings using compositional n-gram features. Proceedings of NAACL-HLT 2018. New Orleans, LA: ACM, 2018. p. 528–540.
64. Gad WK, Kamel MS. Incremental clustering algorithm based on phrase semantic similarity histogram. Proceedings of the Ninth International Conference on Machine Learning and Cybernetics. Qingdao, China, 2010. p. 2088–2093.

65. Ballarini P, Barbot B, Duflot M, Haddad S, Pekergin N. HASL: A new approach for performance evaluation and model checking from concepts to experimentation. Perform Eval. 2015;90:53–77.
66. McMinn AJ, Moshfeghi Y, Jose AM. Building a large-scale corpus for evaluating event detection on Twitter. Proceeding of the 22nd ACM International Conference on Information Knowledge Management. San Francisco, CA, USA: ACM, 2013. p. 409–415.
67. Alguliyev RM, Aliguliyev RM, Isazade NR, Abdi A, Idris NCOSUM. text summarization based on clustering and optimization. Expert Systems. 2019;36(1):e12340. https://doi.org/10.1111/exsy.12340.
68. Sato K, Wang J, Cheng Z. Credibility evaluation of Twitter-based event detection by a mixing analysis of heterogeneous data. IEEE Access. 2019;7:1095–106. https://doi.org/10.1109/Access.2018.2886312.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.