Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Detecting life events from twitter based on temporal semantic features

Maryam Khodabakhsh[a], Mohsen Kahani[a,*], Ebrahim Bagheri[b], Zeinab Noorian[b]

[a] *Web Technology Laboratory, Dept. of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*
[b] *Laboratory for Systems, Software and Semantics (LS^3), Ryerson University, Toronto, Canada*

A R T I C L E   I N F O

Article history:
Received 23 April 2017
Revised 11 January 2018
Accepted 11 February 2018
Available online 15 February 2018

Keywords:
Social web
Social networks
Event detection
Personal life events

A B S T R A C T

The wide adoption of social networking and microblogging platforms by a large number of users across the globe has provided a rich source of unstructured information for understanding users' behaviors, interests and opinions at both micro and macro levels. An active area in this space is the detection of important real-world events from user-generated social content. The works in this area identify instances of events that impact a large number of users. However, a more nuanced form of an event, known as life event, is also of high importance, which in contrast to real-world events, does not impact a large number of users and is limited to at most a few people. For this reason, life events, such as marriage, travel, and career change, among others, are more difficult to detect for several reasons: i) they are specific to a given user and do not have a wider reaching reflection; ii) they are often not reported directly and need to be inferred from the content posted by individual users; and iii) many users do not report their life events on social platforms, making the problem highly class-imbalanced. In this paper, we propose a semantic approach based on word embedding techniques to model life events. We then use word mover's distance to measure the similarity of a given tweet to different types of life events, which are used as input features for a multi-class classifier. Furthermore, we show that when a sequence of tweets that have appeared before and after a given tweet of interest (temporal stacking) are considered, the performance of the life event detection task improves significantly.

1. Introduction

Social networking platforms are considered to be among the foremost means of communication and social interaction. The popularity of these platforms leads to the fast and real-time spread of information. Twitter is a popular platforms with special characteristics that makes it ideal for the fast and wide distribution of information. Users tweet in various domains such as daily activities [58], life events [18,20,21], and the latest local and global news [34,35,54,56], just to name a few. Several authors have already shown that the consideration of Twitter content leads to faster access to news compared to traditional news outlets [23,32]. For instance, on the day of the 2016 U.S. presidential election, Twitter proved to be among the largest sources of breaking news with 40 million tweets sent on the topic by 10 p.m. that day. As another example, there were many user tweets about the death of the celebrity singer Whitney Houston before it was even mentioned on traditional media [2,57]. Many users even exploit Twitter for more personal purposes and share their daily activities, life happenings, feelings and opinions.

The focus of our work in this paper is on life events, which is a subset of events that affect an individual's life. Marriage, graduation, career change, travel, and job promotion are examples of life events. Li et al. [36] have extensively identified a set of forty two life events in their work while the work by Glück and Bluck [30] provides a set of life events from an autobiographical memory perspective. It is also worth mentioning that life events can in some cases intersect with general broader events. For instance, an award ceremony can be considered to be a public event but at the same time it can be regarded as a personal event for the award recipients.

Lin et al. [37] believe that people have become more inclined to share their life events via social media such as Twitter making it is possible to identify signs of personal experiences, emotions, and life decisions from users' social traces. Therefore, the identification of life event information from social networks can have important practical applications. For instance, banks can recommend appropriate loans to couples who have just recently become engaged or real-estate agents could identify and engage with customers who are about to have a new baby to explore the possibility of moving into a larger house. Advertising companies can also

* Corresponding Author.
E-mail addresses: maryamkhodabakhsh@mail.um.ac.ir (M. Khodabakhsh), kahani@um.ac.ir (M. Kahani), bagheri@ryerson.ca (E. Bagheri), znoorian@ryerson.ca (Z. Noorian).

https://doi.org/10.1016/j.knosys.2018.02.021
0950-7051/Crown Copyright © 2018 Published by Elsevier B.V. All rights reserved.

benefit from the identification of life events whereby they can promote rings to-be-wed couples or baby care products to expecting mothers. Customer insight gathering is an additional form of exploiting life event information. One way in which marketers gain insight about consumers is by identifying the occasions in which consumers use their products. Identifying occasions, such as birthdays, helps in consumer segmentation, answering why consumers purchase a product, and where and when they use it [9]. Having mentioned these applications, it is of significant importance to highlight the privacy implications of these applications. The respect for user privacy needs to be taken with utmost consideration and these applications are to be exercised only when explicit permission has been granted by the user. For instance, often times, users grant permission for companies to access their social content and only under such consent can these applications be executed.

From an entertainment perspective, projects such as Museum of Me (MoM), Facebook Lookback (FL) and Google Awesome try to produce video clips for users to summarize and visualize interesting and important moments in their life. These projects understand the necessity of the existence of methods for life event detection from personal yet social content. The detection of life events allows such platforms to augment user profiles with information that might not be explicitly available or provided by the users [31].

The information retrieval community has seen abundant techniques that identify events from social content. The primary method for identifying events has been to monitor surge in specific categories of content in order to pick up mentions of real world events. Event detection methods based on Graph theory [41], Wavelet and Fourier transform [59], and term frequency changes [49] are among the methods that primarily rely on changes in the volume of social content for identifying real world event mentions. While these methods have shown impressive performance, they are not well-suited for the identification of life event mentions from personal social timelines for several reasons: i) Life events have very low frequency in a person's social stream and happen sporadically; therefore, relying on volume surge would not be an effective method to identify life event mentions; ii) Life events are personal events that appear on a single person's social stream (either through content posted by that person or other users). Therefore, methods that focus on content that is generated by a large user base would not be suitable for the detection of life events; and iii) There is no guarantee that every user will post about her life events and in fact it has been reported that life event mentions are not very frequent, making them harder to identify. For these reasons, existing event detection methods are not applicable for identifying life events.

Another aspect of life event detection that makes it even more challenging is the prevalence of social content that are similar in nature to life events but are in fact not mentions of life events. For instance, there are many travel agencies, event planners and employment companies that post promotional content on Twitter that look very similar to life events. For instance, '`#blackfriday offer get our prewedding Diet Plan for #brides`' is a tweet that has all the right components of a life event but is in fact the promotion of a Black Friday deal and not the report of a personal life event. Therefore, the identification of life events would also need to consider the *personal self-report* aspect of content as well [36].

In this paper, we address the problem of identifying personal self-report mentions of life events on Twitter. The objective of our work is to determine whether a given tweet of a user includes a mention of a life event and if so which specific life event it references. To this end, our work rests on and explores two fundamental ideas:

1. Given the sporadic and low occurring nature of life events, features that are based on some notion of frequency would not be suitable features for determining life events. It is our hypothesis that life events are appropriately identified if semantic features are taken into consideration. For this purpose, we model life events based on well-known word embedding techniques such as GloVe [47] to create a representation for life events. This representation is then used to determine whether a tweet is discussing a certain life event or not.
2. Furthermore, we hypothesize that life events have a certain build up nature to them in that people who are about to or are reporting a life event usually show signs of that life event in their past posts and continue to discuss this event in the future. Therefore, a temporal consideration of a user's social stream could serve as a good indication of a life event. Such an approach will be able to discriminate between personal self-reported life events and mentions of 'pseudo-life event mentions' coming from advertising companies given the different nature of reporting behavior. For instance, a user reporting his wedding plans will not constantly and solely talk about the wedding and might post other content as well whereas a wedding planning company will solely post about weddings.

The work proposed in this paper consists of two layers: 1) In the first layer, we build a multi-label classifier to identify mentions of personal life events on Twitter. The distinguishing aspect of our work from the work in the literature is that our work explores the possibility of using features based on word embeddings. In the experiments, we will show that using such features significantly improve the quality of the results in terms of recall; however, this comes at the cost of precision. 2) In the second layer, we propose the idea of temporally stacking the classifier learnt in layer 1 to improve performance. Our experiments show that the proposed temporal stacking model improves both precision and recall of layer 1 classifiers.

More specifically, the key contributions of our work can be summarized as follows:

1. We show that a semantic feature derived based on the idea of representing life events through word embeddings provides strong discriminatory power that can be used for the purpose of detecting self-reported mentions of life events.
2. We introduce the concept of *temporal stacking* to show that when weaker life event classifiers are applied to a certain set of tweets prior to or after a given tweet and the generated labels are used as features of a second layer classifier that the life event detection performance improves significantly.
3. We demonstrate the performance of our work on a gold standard dataset that consists of six distinct life events and compare our work with the state of the art and show that our work outperforms the state of the art in both precision and recall metrics after temporal stacking.

The rest of this paper is organized as follows. In the next section, we review the related work. The overview of the proposed framework for detecting self-reported life event mentions is introduced in Section 3. Section 4 provides the technical details of our proposed work. Section 5 is dedicated to the details of our experimentation and our findings. Finally, in Section 6, we conclude the paper.

2. Related work

Event detection from within a stream of document collections is one of the active research topics in information retrieval [4] and is considered to be one of the main five central themes within the Topic Detection and Tracking (TDT) domain [5]. Document streams can be collected using social streams, online conversations, email exchanges, blog posts, or corporate communication. There is significant research work being conducted in event detection with special focus being given to streaming corpora as well as emerging forms of content including microblog posts such as Twitter.

Within the literature, events are considered to be real world incidents that occur in a specific time [55,61] and are attributed to particular locations [60] and agents who are affected by or affect the outcome [17]. As such, events on social media are a reflection of real world events through the content generated by users, e.g., by the tweets that are published to describe or react to a real world event [22]. Event detection on Twitter has been more challenging due to the special characteristics of tweets being short in nature and the prevalence of misspelling, abbreviations, slangs and the invention of Twitter-specific jargon, e.g., *adventuritter* is a new term on Twitter used to refer to a twitterer who is adventurous. Also as discussed in the literature the syntax structure as well as semantics of content shift given the nature of the Twitter platform [24,26,27]; therefore, existing methods for event detection for classic corpora would not work too well for microblogging platforms.

Aiello et al. [3] have compared several techniques for event detection on Twitter, and promoted a technique based on term clustering for finding trending topics. The six techniques introduced in their work fit into two main categories: document clustering versus term clustering, where a cluster represents a potential topic of interest. These approaches can be further categorized into three different classes: probabilistic models, e.g., Latent Dirichlet Allocation (LDA), classical Topic Detection and Tracking methods, e.g., Document-Pivot Topic Detection, and feature-pivot methods, e.g., n-gram clustering. Abdelhaq et al. additionally discuss the role of spatio-temporal features for event detection in their EvenTweet work [1]. Their work is based on an initial clustering of keywords according to their spatial signature. Keywords that appear in the same location will be included into the same cluster. These keywords receive a score according to their level of burstiness, their spatial distribution and other temporal features. Closer to the theme of our paper that focuses on the role of word embeddings, the recent work by Ertugrul et al. [25] introduces a method that benefits from the embedding representation of words in tweets. In their work, the representation of each tweet is developed based on the vector representations of its constituting words, which is then used to calculate the distance between pairs of tweets to be used in hierarchical clustering.

Within the event detection literature, a significant amount of work has been dedicated to extracting features that can be used for modeling and detecting events. One of the most common features is the 1-gram (bag of words) feature that models a tweet as a collection of words that have appeared in it. For instance, Di Eugenio et al. [20] and Dickinson et al. [21] have examined various types of features and found that n-grams are amongst the most discriminative features for a host of tasks. However, the authors have also reported that the use of n-grams and bag of words models suffer from the curse of dimensionality, which is aggravated within the context of Twitter given the wide range of slangs, acronyms and abbreviations. Another downside to these types of features is that they overlook the temporal evolution of n-grams and hence significant information could be lost in the process. Several researchers have moved beyond n-grams and used named entity mentions to model events on Twitter. The objective has been to extract the four main WH questions related to events through named entity recognition [21] and semantic role labeling [20]: who, what, when and where. If properly extracted, named entity mentions can be strong indicators for events in the real world as repeated mentions of unique locations, organizations, time and date, among others could lead to the detection of an event. In addition to named entities, depending on the event that is being tracked or identified, users' sentiments can also point to mentions of events [18].

Syntactic features, for example those extracted through part of speech tagging, have also been widely used for modeling and detecting events on traditional corpora; however, such syntactic features are not the best features for user generated content on microblogging platforms given the predominant informal language used in these platforms [20,36]. More specific to microblogging and messaging platforms, users can include related keywords or topics in the form of hashtags or express their emotions through the use of emoticons. The use of hashtags and emoticons had also been shown to have strong discriminative power for classification purposes on social network data [18]. Furthermore, activity features are a novel set of features that account for the activities of users in specific time intervals on social networks. For example, the number of tweets posted by a user, number of replies given by the user to other users and the number of retweets posted by a user are some examples of activity features. The motivation for using activity features is based on the simple logic that important events are bound to generate more attention and activity within the immediate personal network of an individual [17,18,21].

Attention features [17] are a different set of features that can be defined as signs of notice taken by other users expressed through reply and retweets that the user has received. These features reflect how many times the user is addressed/talked about by other users in a given time interval. Similarly, Dickinson et al. [21] have referred to attention features as interaction features; however, rather than just considering the number of retweets, favorites or replies, they consider who are the users performing these actions and their interaction patterns with the user of interest. While these features capture an important aspect of user activity, these authors found almost no effect of interaction features on life event detection when applied to Twitter. Choudhury and Alani [17] investigated a number of user activity and attention features to detect personal life events in tweets. The focus of their work was on identifying whether the daily collection of tweets from a user contained the reporting of any personal events. Contrary to [21], they concluded that life event detection based on attention features performed best, followed by activity based features. In another work by the same authors [18], they used activity and attention features along with n-gram sentiment, and emoticons to detect life events and found that activity and attention features did not yield substantial improvement contrary to the expectation.

It is common to identify events from social network data based on supervised or unsupervised classification techniques. Based on a recent survey by Atefeh and Khreich, [6], the most widely used techniques for unspecified event detection from Twitter rely primarily on clustering approaches. In this context, unspecified events are typically expressed as emerging events, breaking news, and general topics that attract the attention of a large number of Twitter users. By considering studies on life event detection [12,13,18,20,21], it can be said that using supervised techniques is common for life event detection similar to unspecified event detection. One of the main obstacles in building models for life event detection is the curation of labeled life event datasets, which is a time consuming and laborious task. Li et al. [36] identified common categories of major life events by leveraging large quantities of unlabeled data and obtained a collection of tweets corresponding to each type of life event. By using the idea that major life events will elicit signs of congratulations or condolences from the user's followers, they collected large volumes of high-precision personal life events which can be used to train models to recognize the diverse categories of major life events discussed by social media users. An LDA based topic model was then used to cluster the gathered tweets to automatically identify important categories of major life events in an unsupervised way. Also they adopted a semi-supervised bootstrapping approach to expand event-related tweets. In contrast, the work in [21] has used Crowdflower[1] as an annotation tool to curate their training life event dataset. Crowdflower is an online crowdsourcing platform, where uploaded datasets are annotated by a large audience.

While most previous works predominantly focus on the use of machine learning techniques for life event detection, Cavalin et al. [13] have been among the few to propose a hybrid life event detection approach, which introduces the role of rules. Their system is composed of three modules, namely Ingest, Filter, and Detect. The first Ingest module captures a database of tweets to be used for the search of life events. This is done by considering a set of words that can possibly relate to all life events of interest in the system. The Filter module selects the set of tweets that is more likely to contain life events. That is, by considering a set of simple rules such as a combinations of words, the posts that match these rules are marked with the corresponding possible life events. The Detect phase is then carried out to validate the identified life events. For each tweet found in the Filter phase, the authors then applied a machine learning classifier to compute the probability of each tweet belonging to a given life event.

It is also worth noting that the body of literature on event detection from multimedia content also cover similar work to the work in event detection. For instance, Chen and Roy [16] and Becker et al. [7] focus on identifying a group of photos from Flickr that collectively represent a real-world event. Their work is primarily based on the temporal and spatial distributions of tags associated with photos, which are employed within a wavelet transform-based method to find tags with significant peaks. These tags are then used to cluster the associated photos into event-related groups. In another work [48], the authors also aim at identifying social events based on multimedia content on Flickr. The authors move beyond the work of Chen and Roy in that they additionally benefit from visual descriptors to cluster images into groups related to real-world events. Ma et al. [39] also address the issue of multimedia event detection with special focus on identifying rare events in video content. The novel aspect of this work is to use a variation of transfer learning for using partially overlapping features given there is usually insufficient positively labeled instances for rare events. Finally, the work by Chang et al. [15] addresses the problem of pooling frames of a long untrimmed video such that only relevant video frames to a specific event are retrieved and ordered.

3. Approach overview

The main objective of our work is to identify personal reports of life events of users on Twitter. To this end, we propose to turn the life event detection problem into a supervised machine learning problem where information extracted from the social feed of the users would be used to form discriminative and indicative features for life events. The overview of the components and flow of our proposed approach is shown in Fig. 1. As seen in the figure, our approach consists of two main layers. The first layer is responsible for detecting whether a tweet is a self-reported case of a specific life event by training a multi-label classifier. In order to build such a classifier, we use the Word Mover's Distance (WMD) between the tweet and the different set of life events and use these distances as input features for training the classifier. We employ a word embedding-based representation of both tweets and life events that capture the semantics of the content and at the same time allow us to measure distance between the tweet and the life event space. In order to learn the word embedding-based vectors, the Skip-gram model is employed and applied on a large corpus of tweets. Based on the produced word vectors and an initial set of seed words representing life events, we model each life event as a collection of word vectors. In the next step, the similarity of each life event and an input tweet is computed by using the WMD measure. We will discuss in the experiments section of this paper that the state of the art baseline methods have very low recall rates but reasonable precision. These methods use features such as hashtags, emoticons, and sentiments, among others. In contrast, our proposed features set based on the WMD measure results in higher recall at the cost of precision.

In the second layer, our aim is to improve the precision of the life event classification process by proposing the idea of *temporal stacking*. In temporal stacking, we propose that if features are extracted and used from a certain time period before and after the date in which the input tweet is posted that a more precise prediction of the life event can be achieved. To do so, we build a second layer classifier based on the information of the first layer classifier and the tweets from before and after the tweet of interest. We show in our experiments that regardless of the features used in the first layer classifier (our proposed WMD feature versus other baseline features, e.g., hashtags or emoticons), the performance of the life event classifier improves significantly and hence addresses the issue of low precision in the first layer life event classifier.

4. Proposed approach

Let us first provide some preliminary definitions that will help us formalize our approach.

Definition 1 (Tweet)**.** A tweet $tw_u^t = (text, u, t)$ is a triple where $tw_u^t.text$, $tw_u^t.u$ and $tw_u^t.t$ denote the tweet content, the user who posted the tweet and its posting time, respectively.

Now, given the objective of our work is to identify life event mentions, we formalize a concrete representation of a life event as a set of word vectors that represent that life event in a discriminative way. In our model, each life event is demarcated using a set of words. For instance, a set of words such as 'marriage, engagement, bride, groom, honeymoon' could form the representation of the Wedding life event. However, in order to go beyond a bag of words representation of each life event, we employ the vector-based word embedding representation of each of these words. This way, a life event such as Wedding will be represented as a set of vectors denoting each of the discriminative words computed through a word embedding mechanism.

Definition 2 (Life event likelihood)**.** Let t be a specific timestamp, given $LE = \{le_1, le_2, \ \dots, le_k\}$, which denotes a collection of k life events and tw_u^t, the life event likelihood of user u in time t, called LEL_u^t, is represented by a set of probabilities $\{p_{u,1}^t, p_{u,2}^t, \dots, p_{u,k}^t\}$ where $p_{u,k}^t$ denotes the likelihood of life event k for user u in time t.

One of the underlying assumptions of our work is that in any given timestamp, there is a likelihood distribution over the life event set for each user. In other words, it is possible to calculate a likelihood distribution over the life events for each user in timestamp t. It should be noted that we also include a *no-event* situation, which shows that the user is not engaged with any life events in timestamp t. Now, based on these definitions, we can formally define our problem statement as follows:

Definition 3 (User life event detection)**.** Let t be a specific timestamp, given the set $LE = \{le_1, le_2, \dots, le_k\}$ and tw_u^t, the goal of the User Life Event Detection problem is to find LEM_u^t, which is the life event that user u is engaged with at time t, from the set LE.

[1] https://www.crowdflower.com/.

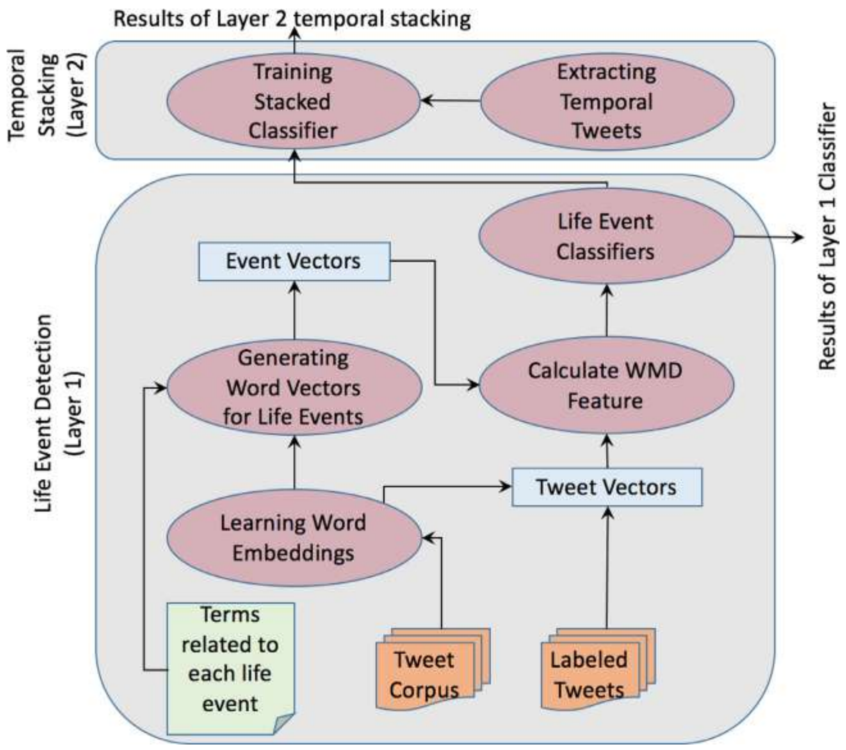

Results of Layer 2 temporal stacking
Temporal Stacking (Layer 2)
Training Stacked Classifier
Extracting Temporal Tweets
Life Event Detection (Layer 1)
Life Event Classifiers
Results of Layer 1 Classifier
Event Vectors
Generating Word Vectors for Life Events
Calculate WMD Feature
Tweet Vectors
Learning Word Embeddings
Terms related to each life event
Tweet Corpus
Labeled Tweets

Fig. 1. Overview of the proposed approach.

We divide this problem into three sub-problems: *Life Events Modeling* (extracting *LE*), *Tweet Modeling* and *Life Event Detection* (determining LEM_u^t).

4.1. Life event modeling

The first step of our work is to produce a concrete representation for each life event. As mentioned earlier, we represent each life event as a set of word vectors that correspond to discriminative words related to the life events. We refer to these words as life event features.

Definition 4 (Life event features)**.** Let *LE* be a list of *k* life events, i.e., $LE = \{le_1, le_2, \ldots, le_k\}$, Life Event Features for $le_n \in LE$, denoted by $LEW_n = \{w_{1,n}, w_{2,n}, \ldots, w_{N,n}\}$, is a set of words that describe life event le_n.

As already discussed by Choudhury and Alani [18], it is possible to identify the set of such discriminative words for each life event. In order to identify such words, we manually curate a set of hashtags related to each of the life events. For instance, the Wedding life event would be related to a set of hashtags such as `#wedding, #engagement, #bride, #groom, #bridal,` and `#weddingdress`, among others. We then use snowballing to identify other related and relevant hashtags for each life event. Once a comprehensive set of hashtags are identified, we retrieve a set of tweets that are tagged with these hashtags. Based on this process, we collect a set of tweets for each life event type. Once the tweets are collected, they are reviewed to ensure that they are in fact self-reporting tweets about life events. There are many cases where tweets such as `‘Don’t forget 20% off our wedding photography packages this winter #uksmallbiz #wedding #weddinghour #bride #groom #offer #savings’` are retrieved because they have relevant hashtags, but they are not about life events. Such tweets are manually removed from our set . Now based on the filtered collection of tweets, we use term frequency and inverse document frequency of terms within the context of each life event and across all life events to identify terms that are specific to each life event. We select top-5 words for each life event based on this process as shown in the second column of Table 1. An interesting observation is that the word ‘I’ is seen in all life event types showing its significance as an indicator for the self-reporting aspect that is of interest to our work.

Once the top-5 words are identified, we use the word embedding approach to identify highly similar words from within our tweets collection to the set of top-5 words in each life event. In other words, we look for words based on their similarity in the vector space that are collectively closest to the set of five words for each life event. Various models have already been introduced for learning word embeddings including neural network language models [8,43] and spectral models [19]. More recently, Mikolov et al. [42] have proposed two log-linear models, namely the Skip-gram and CBOW models to efficiently learn word embeddings. These two models have a low time complexity and hence can be efficiently applied on large-scale corpora. The geometric properties of the semantic space prove to be semantically and syntactically meaningful, that is, words that are semantically or syntactically similar tend to be close in this space. Given the fact that the experiments in [42] have shown that the Skip-gram model outperforms CBOW in identifying semantic relationships among words, we employ the Skip-gram model for discovering similar words in our study. On this basis, LEW_n is finally built by including the top-5 words identified in the previous step and the most semantically similar words to the collection of top-5 words for each life event based on the Skip-gram model. The collection of words is shown in Table 1.

As a final step, we represent each life event as a set of word vectors where each of the vectors represent the individual words shown in Table 1. We transform LEW_n to a set of words vectors $le_n = \{v(w) | w \in LEW_n\}$ where v is a function that computes the vector of word w in life event le_n using the Skip-gram model.

Table 1
The list of life events and LEW_n for each life event.

Life event	Most co-occurring terms (top-5)	Most semantically similar terms to the top-5
Broken Device	Phone, I, Break, Drop, iPhone	shatter, deraydailydose, detection, Smartphone, varsityblue, skip, device, protector
Device Upgrade	Phone, I, Time, Wait, iPhone	out-of-pocket, upgrade, engadgetce, thunderbolt, thetekguy, overdue, tech, itox
Moving	Move, I, House, Hunting, Wait	move, happen, break, highschoolmemory, stairmaster, property, rent, place
New Job	Job, I, Start, Excite, Hire	job, itjdb, parttime, overlandpark, practice, rehearsal, work, collegeperk
Travel	I, Vacation, Book, Trip, Holiday	travel, tripadvisor, heybackpacker, inclusive, frommer, traveltip, agent, package
Wedding	Wedding, I, Party, Plan, Husband	bride, groom, engagement, bridal, bhldn, dress, shower, ring

4.2. Tweet modeling

The objective of this step of our work is to extract features from tweets that could be used for determining whether a life event is being mentioned or discussed in the tweet or not. It is common practice to represent a tweet as a bag of words or a bag of n-gram [13,44]. In such approaches, both the life event and the tweets are considered to be a bag of words or bag of n-grams and hence matching between life events and tweets can happen naturally. However, such a modeling approach, often known as textual features, often faces problems due to the need to deal with slangs, acronyms, abbreviations, and misspelling errors that are prevalent in tweets. Extracting syntactic features such as part of speech tagging [36] is another method for modeling tweets. While such features can show reasonable performance on well-structured textual content, they are less accurate in the context of tweets where observing correct grammatical rules is less common due to the informal language that the users adopt. One of the more effective set of features for modeling tweets is semantic features such as named entities [36], semantic role labels [20], and sentiments [18]. This is particularly true for determining life events given the fact that life events are often associated with certain named entities as well as emotions that can be effectively extracted from tweets using named entity recognition and sentiment analysis.

While textual, syntactic and semantic features are the primary set of features used for the classification of pure textual content-based corpora, as mentioned earlier, some additional features can be defined based on the nature of social microblogging platforms such as activity or term features. Activity features [17,21] are based on the actions that users can perform on the social network such as retweeting, replying or favoriting a certain tweet, which can in itself carry significant context and meaning that can hence be used as features. Other features such as hashtags and emoticons are known as term features. Hashtags are often known to represent the core topic or sense of a tweet and can therefore be considered to be quite a powerful feature. Furthermore, there are many emoticons, which are topic-specific (specific life events) and also can carry sentiment value.

Now while many of these features are expected to have good discriminative power, but as we will show later in the experiments, they do not perform too well for life event detection given the informal and short nature of tweets that lead to feature sparsity. Therefore, while we will adopt these features to build baselines for comparison as suggested in [18,36], in our own work, we use the word vector representation of tweets that can be obtained from word embedding techniques. These features incorporate both syntactic and semantic aspects of the content and therefore show to be effective features for identifying life events.

Definition 5 (Tweet word vectors)**.** Given tw_u^t, a Tweet Word Vector, denoted by twV_u^t, is a set of word vectors, each representing the words in tw_u^t.

We model each tweet as a bag of word vectors where vectors of words are computed using the Skip-gram model.

4.3. Life event detection

Now, given that we have built representations for both life events as well as user tweets, our objective is to identify mentions of life event self-reports on Twitter. Both of the representations that we have adopted are based on the word embedding technique where words are represented through their vector model from the embedding space. Therefore, it is possible to directly calculate the similarity between a given tweet and the life event vector representation. Let tw_u^t be a tweet written about a life event le_n by user u at time t, we consider the tweet tw_u^t to be related to life event le_n if the word vectors in tw_u^t are semantically similar to the word vectors of le_n.

The intuition behind our approach is based on how users adopt terminology to express their intent. The assumption is that if the set of word vectors representing a life event is accurately selected, then any other life event self-report observed from the users will use words or terminology that will be semantically close to the life event representation in the embedding space. Therefore, given the fact that the position of a life event can be determined based on the position of its constituent words within the embedding space and also the position of a tweet can be determined in the same way in the same space, it is possible to calculate the distance of each tweet to the set of life events under consideration. For this reason, we define similarity between tw_u^t and life event le_n by a Tweet-Life Event Similarity function as follows:

Definition 6 (Tweet-life event similarity)**.** Given twV_u^t and $le_n \in LE$, Tweet Life Event Similarity is equivalent to the inverse of the Word Mover's Distance between the vector representations of the words in twV_u^t and le_n and formally denoted as $sim(twV_u^t, le_n)$.

By considering the fact that the less the distance of two vectors is, the greater their similarity would be, we have adopted the inverse of the Word Mover's Distance (WMD) as a way to calculate similarity [33]. WMD is an instance of the Earth Mover's Distance and measures distance between twV_u^t and le_n as the minimum amount of distance that the embedded words of twV_u^t need to 'travel' to reach the embedded words of le_n. In WMD, the distance between two documents is calculated by the minimum cumulative distance of the best matching embedded word pairs in the two documents. In the context of our work, the distance between twV_u^t and le_n will be based on transporting words in twV_u^t to words in le_n. The transportation matrix $\boldsymbol{T}$ is a *flow matrix* in which $\boldsymbol{T}_{i,j}$ shows to what degree word i in twV_u^t is transported to word j in le_n. Matrix $\boldsymbol{T}_{i,j}$, which essentially determines what word pairs from the two documents should be connected to each other, needs to be learnt based on a linear optimization program. The distance between two documents can be calculated by minimizing the following linear optimization function:

$$distance(twV_u^t, le_n) = min \sum_{i=1}^{|twV_u^t|} \sum_{j=1}^{|le_n|} \boldsymbol{T}_{i,j} \times d(i,j) \quad (1)$$

where $d(i, j)$ is the distance between word i of twV_u^t and word j of document le_n.

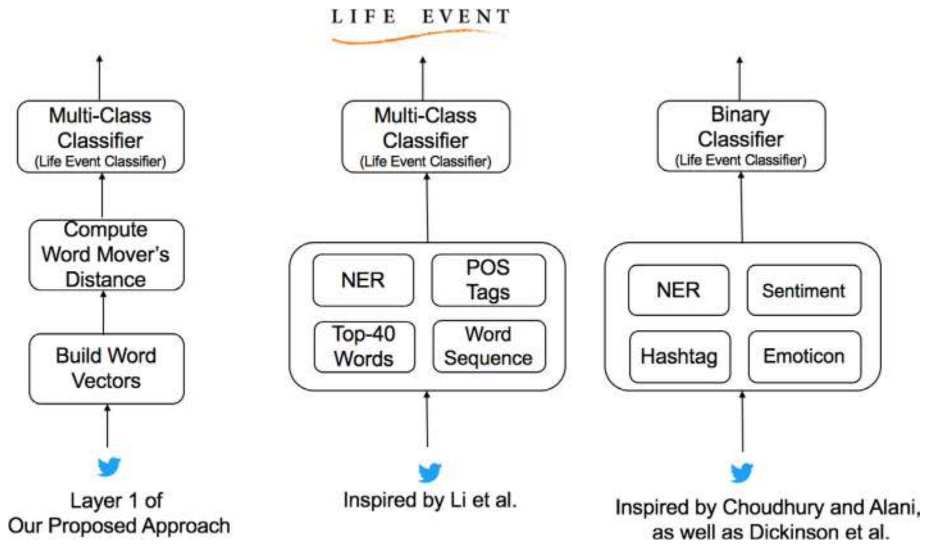

LIFE EVENT
Multi-Class Classifier (Life Event Classifier)
Compute Word Mover's Distance
Build Word Vectors
Layer 1 of Our Proposed Approach
Multi-Class Classifier (Life Event Classifier)
NER
POS Tags
Top-40 Words
Word Sequence
Inspired by Li et al.
Binary Classifier (Life Event Classifier)
NER
Sentiment
Hashtag
Emoticon
Inspired by Choudhury and Alani, as well as Dickinson et al.

Fig. 2. Schema of different life event detection approaches.

In order to build LEL_u^t and considering $0 \le sim(twV_u^t, le_n) \le 1$, $sim(twV_u^t, le_n)$, which is the inverse of $distance(twV_u^t, le_n)$, can be the appropriate representation for $p_{u,n}^t$ in Definition 2.

As mentioned above, using the Skip-gram model helps to preserve the semantic relationship between word vectors and the distances between the embedded word vectors. WMD utilizes this property of word embeddings and therefore translates this into the relationship between the vector-based representation of tweet and life events. The distance between a tweet and life event is the minimum cumulative distance between the words in the tweet to their matching words in the life event representation.

After computing LEL_u^t, we need to find LEM_u^t, i.e., the life event that user u reported in tweet tw_u^t. Our solution for finding LEM_u^t is to train a classifier, named life event classifier, with the set of LEL_u^t as its input and the detected life event of interest as output. The reason we adopt a classifier to predict the life event is that the event with the highest $p_{u,n}^t$ does not necessarily point to the life event being discussed. The detection of the correct LEM_u^t depends not only on the LEL_u^t for each individual life event but also on the distribution of LEL_u^t over the whole set of life events. Another important consideration is the 'no-event' label that needs to be predicted. If only the set of life events and their LEL_u^t are considered, we will end up selecting one of the life events that has the highest similarity to the tweet under consideration even if the tweet is not discussing any life events. Therefore the life event classifier is effective in helping not only identify the correct life event but also prevents us from producing too many false positives.

Fig. 2 shows a schematic overview of our proposed approach compared to other state of the art baselines. We use classical supervised learning techniques such as Support Vector Machine (SVM), Random Forest (RF) and Gradient Boosting Tree (GBT) in order to train various classifiers for the life event classifier. SVM [14] is one of the most effective supervised classification methods. Given a set of training data as input, the SVM training algorithm builds a model that assigns new examples into one label, based on the margin maximization strategy. Choudhury and Alani [18] have already used classifiers such as SVM, Naive Bayes, and Decision Tree for life events detection on Twitter [18]. In their experiments, SVM showed the best performance in 4 out of 5 life events.

Random Forest and Gradient Boosting Tree are ensemble learning methods. Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone [53]. Friedman [28] has shown that gradient boosting trees compete with the state-of-the-art machine learning algorithms such as SVM with much smaller models and faster decoding time [29]. The main idea in boosting is that a set of weak learners create a single strong learner. This improves accuracy by reducing bias, and also variance. Gradient boosting trees have already been applied as a high-precision classifier for event detection on Twitter [50,51]. Pennacchiotti and Popescu [46] also used gradient boosting decision trees in order to build a general and robust machine learning framework for large-scale classification of social media data especially from Twitter users posts.

Bootstrap aggregating (bagging) is another method built on top of the random forest idea to improve stability and accuracy of classification by reducing variance and helping to avoid over-fitting [10]. Castillo et al. [11] have proposed a supervised learning approach for the automatic discovery of relevant and credible news events from Twitter [11] and studied how different learning algorithms perform in their particular learning scenario and found that random forests achieve high accuracy rates. Liu and Huet [38] also propose an event-based media classification framework in order to study feature importance for modeling the relation between events and media, and how to deal with missing and erroneous metadata often present in social media data [38]. Their experimental results show that the best model is learned by Random Forests in combination with spatio-temporal and tag features. Furthermore, Meij et al. [40] have proposed to combine high-recall concept ranking and high-precision machine learning methods including random forests and gradient boosted regression trees for automatically mapping tweets to Wikipedia articles.

While the above examples show that various classification algorithms have been effective for different tasks on Twitter data; nevertheless, as demonstrated in [12,13], the life event detection task seems to be among the more difficult tasks as it is categorized as an unbalanced classification problem, which means that the number of tweets that does not contain any life events is much higher compared to the number of tweets that do in fact talk about life events. The main reason is that, besides the actual life events, a lot of non-personal content is generally posted on social networks, such as advertisements, comments related to celebrities and jokes. As a result the training of a machine learning classifier to detect actual life event self-reports with a reasonable precision and recall rates is challenging.

4.4. Temporal stacking

As we will see later in the experiments section, although the life event classifier built in the previous section outperforms the state of the art baseline in terms of F-score, it is still far from being a strong classifier. We further endeavour to build a stronger classifier by temporally stacking the weaker classifier (life event classifier) built in the previous step. Our intuition is based on how people react and report life events. We hypothesize that users report personal life events through a sequence of messages. In other words, observations of life event mentions on social networks has a trajectory. For instance, if someone is about to get married, they are very likely to post several messages on their social feed or their friends' social feeds. This could be messages about how they are preparing for the wedding, and all the related activities. As an example, a Twitter user reporting his marriage on Twitter says: `‘Holy [... ] I am getting married Today!!!! -Maught’`. When looking at the previous and next message on this user's social stream, one would find the following tweets as well: `‘Best ring bearer ever!’`, `‘Having fun at the wedding’` and `‘Seeing you two happy makes me the happiest person alive. Congratulations to my two favourite people’`. As seen here, there are indications of the life event in the post and pre related content. Based on this, we hypothesize that it might be possible to build a stronger classifier by temporally stacking the classifier from the previous step.

In order to utilize the life event classifiers from the previous step for building a stronger classifier, we use them to label tweets before and after the tweet of interest up to a maximum of m tweets. We use the labels produced by the weaker life event classifiers to train a new classifier with the goal of predicting the life event class labels. More concretely, we employ the classifier built in the previous step for the various life events to label tweets before and after the life event of interest. We use these labels as the input features of a second classifier, which we call a stacked classifier.

In a study of trends on Twitter, Kwak et al. [34] discovered that most trends last for one week once they become ‘active’. Since we follow a similar intuition, we select a time period of one week prior and one week after the date that a given tweet has been posted. We chose not to use a fixed number of tweets within the temporal stacking process because the frequency of tweeting differs from one user to another. For example, while one user might tweet 20 times in one day, another user might take two months to tweet the same number of tweets. Therefore, for the latter type of user, taking 20 tweets would mean that information from two months ago would be considered when deciding on a recent life event, which is not relevant. Therefore, tweets from a one week time period were considered in our work so as to ensure their relevance to the current life event.

The schematic view of the temporal stacking model is shown in Fig. 3. We will show that when such temporal model is built based on the weaker life event classifiers, it will produce very strong classifiers of life event mentions on Twitter reinforcing our hypothesis that the temporal stacking of the results of a weak classifier on a sequence of social content can lead to the development of a strong classifier. Fig. 4 shows an example case for when temporal stacking was effective. In this case, the proposed life event detection method detected the tweet of interest (in the green box) as a tweet containing the wedding life event. However, it is clear that this is a promotional material. When temporal stacking was used, i.e., the weak life event classifier was applied to tweets from one week before and after the tweet of interest, and those labels were used to predict the life event, it was correctly determined that the tweet does not contain any life events. The reason that the temporal stacking method is able to determine that this is not a personal life event is that all tweets mentioned in this Twitter timeline are all related to weddings, which is a typical pattern observed when promotional material are presented; however, for an actual user who reports his/her wedding, it is likely that they would discuss other topics other than their wedding on their timeline or be engaged with responding to other users (congratulatory) messages about their wedding, which would have a different pattern than that of the promotional material which is consistently focused on wedding content across time.

Table 2
Specification of the corpus and the gold standard dataset.

	Life event	Number of tweets	
Corpus	N/A	10 million	
Gold Standard Dataset	Broken Device	5768	8.69%
	Device Upgrade	2822	4.25%
	Moving	4001	6.03%
	New Job	4001	6.03%
	Travel	3480	5.24%
	Wedding	4015	6.05%
	Negative Samples	42,275	63.7%

5. Experiments

We perform extensive experimentation to answer the following research question: ‘given a certain tweet, would it be possible to determine whether the tweet is about a self-reported life event or not?’ In this section, we describe the experimentally obtained results and evaluate the proposed models for life event detection.

5.1. Dataset

In our experiments, we benefited from three sources of tweets: (1) The first set of tweets was a collection of 10 million English language tweets that were selected from the Spritzer Twitter stream grab, which is publicly available.[2] The *first* 10 million tweets from this tweet collection were selected. The purpose of this first corpus was to learn the word embeddings; therefore, we initially preprocessed the tweets in this corpus by removing URLs as well as reposting marks such as RT and //. We then removed all English stop words from the tweets. Finally, we benefited from the Stanford CoreNLP package to lemmatize all the words in the tweets. Once the preprocessing was completed, the cleaned set of 10 million tweets was used for learning the vector representation of the words used in tweets as well as learning the vector representation of the life events as explained earlier using the Deeplearning4J framework.[3] (2) Furthermore, and in order to evaluate our work, in collaboration with our industrial partner, we developed a second corpus in the form of a gold standard dataset of tweets that included instances of different types of life events as well as negative samples. This gold standard[4] consists of 66,362 tweets that are either labeled with a specific life event or labeled as ‘no-life event’. The no-life event tweets were a collection of tweets that were reviewed by the experts and determined not to be a mention of a life event. The details of the distribution of the life events in the gold standard are shown in Table 2. (3) Finally, and for the purpose of accessing the tweets from one week before and after the tweet of interest, we used Twitter API to retrieve tweets from this two week time frame per tweet in the gold standard dataset.

[2] https://archive.org/details/archiveteam-twitter-stream-2012-01.
[3] https://deeplearning4j.org/.
[4] Could be available after signing NDA.

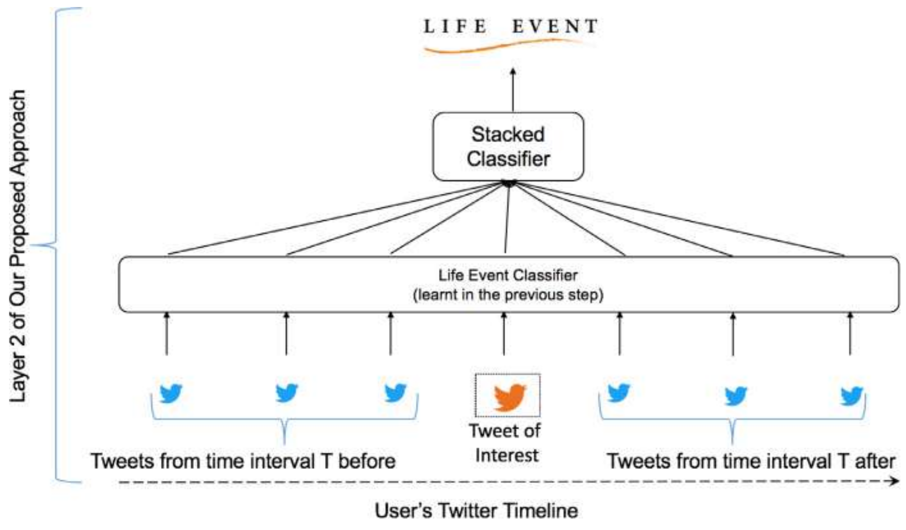
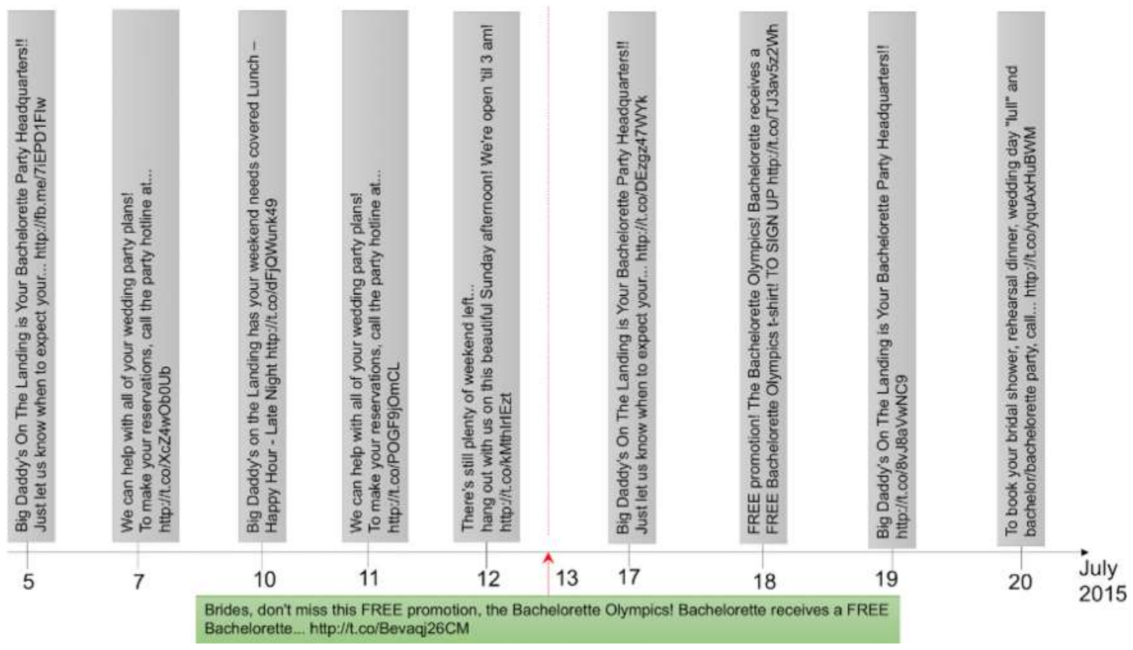

Layer 2 of Our Proposed Approach
LIFE EVENT
Stacked Classifier
Life Event Classifier
(learnt in the previous step)
Tweets from time interval T before
Tweet of Interest
Tweets from time interval T after
User's Twitter Timeline

Fig. 3. Schematic view of the proposed temporal stacking model.



Big Daddy's On The Landing is Your Bachelorette Party Headquarters!! Just let us know when to expect your... http://fb.me/7iEPD1Flw

We can help with all of your wedding party plans! To make your reservations, call the party hotline at... http://t.co/XcZ4wOb0Ub

Big Daddy's on the Landing has your weekend needs covered Lunch – Happy Hour - Late Night http://t.co/dFjQWunk49

We can help with all of your wedding party plans! To make your reservations, call the party hotline at... http://t.co/POGF9jOmCL

There's still plenty of weekend left... hang out with us on this beautiful Sunday afternoon! We're open 'til 3 am! http://t.co/kMthlrlEzt

Big Daddy's On The Landing is Your Bachelorette Party Headquarters!! Just let us know when to expect your... http://t.co/DEzgz47WYk

FREE promotion! The Bachelorette Olympics! Bachelorette receives a FREE Bachelorette Olympics t-shirt! TO SIGN UP http://t.co/TJ3av5z2Wh

Big Daddy's On The Landing is Your Bachelorette Party Headquarters!! http://t.co/8vJ8aVwNC9

To book your bridal shower, rehearsal dinner, wedding day "lull" and bachelor/bachelorette party, call... http://t.co/yquAxHuBWM

Brides, don't miss this FREE promotion, the Bachelorette Olympics! Bachelorette receives a FREE Bachelorette... http://t.co/Bevaqj26CM

Fig. 4. Sample tweets prior to and after the tweet of interest used for temporal stacking.

As shown in Table 2, we have preserved the significant class imbalance that is prevalent in life events on Twitter in this dataset. Given most tweets do not necessarily talk about a life event on Twitter, in our gold standard dataset, we have over 63% of no life event labeled tweets.

5.2. Benchmarks, features and metrics

In order to benchmark our work, we use the most related and state of the art work in the literature to compare with our work. We adopt the work by Li et al. [36] to serve as the baseline. The authors have proposed a pipeline system to detect major self-reported life events on Twitter. Given the code for this work was not publicly available, we replicated their work based on the details provided in their paper with the exact same settings mentioned by the authors where appropriate. To do so, we have extracted features as described in that paper such as sequence of words, named entity mentions, and the top-40 most relevant terms to the life event. For the top-40 most relevant terms, we follow the details provided by the authors to automatically create a dictionary based on topic models. Now, if a dictionary word exists in a tweet then the left and right context words within a window of size three along with their part of speech tags are considered as well. In order to generate the named entity mentions and the part of speech tags, we use Ritter et al.'s [52] Twitter NER system and Twitters POS package [45] for this purpose. We train a multi-class maximum entropy classifier with all these features as proposed by Li et al. [36], which showed the best performance in their experiments.

In addition and in order to build additional baselines beyond the one proposed in [36], we have adopted other features as explained earlier in the paper, namely semantic and term features based on the work in [18]. The classification of these features is shown in Table 3. The table shows the feature classes, types and the methods with which they were extracted. Finally and in order to train our own proposed model, we implement the feature described based on the Word Mover's Distance (WMD) formalized in

Table 3
Types of features and their extraction methods.

Features	Type	Tools
Semantic	Named entities	TAGME, TextRazor
	Sentiment analysis	Stanford CoreNLP
Term	Hashtag	*not required*
	Emoticon	*not required*
Our Feature	Word Movers Distance	Gensim

Table 4
Performance of the proposed life event detection classifiers based on WMD.

Life event/classifier		RF	GBT	RBF SVM
Broken Device	Precision	**0.62**	0.57	0.34
	Recall	**0.56**	0.54	0.61
	F-Score	**0.58**	0.55	0.43
Device Upgrade	Precision	0.63	**0.72**	0.31
	Recall	**0.33**	0.32	0.64
	F-Score	0.43	**0.44**	0.41
Moving	Precision	0.64	**0.66**	0.36
	Recall	**0.47**	0.46	0.54
	F-Score	**0.54**	**0.54**	0.43
New Job	Precision	0.74	**0.76**	0.54
	Recall	0.67	0.65	**0.77**
	F-Score	**0.70**	**0.70**	0.63
Travel	Precision	**0.76**	0.73	0.61
	Recall	0.71	**0.75**	0.73
	F-Score	0.73	**0.74**	0.66
Wedding	Precision	0.51	**0.57**	0.02
	Recall	0.21	0.11	**0.64**
	F-Score	**0.29**	0.18	0.03

Definition 6. We use these features to train several machine learning methods such as random forests, gradient boosted trees and SVM to detect life event mentions in tweets.

In the next subsection, we will show that while the classifier built based on the WMD feature outperforms all the other baselines in terms of F-score, it would still be considered to be a weak classifier. Therefore, as explained earlier, we use this classifier to train stronger classifiers by temporally stacking the results applied to a certain time interval of tweets before and after the tweet of interest.

In terms of evaluation metrics and as proposed in [36], we compare the performance of our work with the baselines using standard information retrieval measures including precision (how many of the life events detected by our method were part of the ground truth set), recall (how many life events in the ground truth set were detected by our method) and F-Score. In calculating the metrics, we employ a *10-fold cross validation* strategy for both the experiments of the first layer classifier as well as the temporal stacking.

5.3. Experiment setup

5.3.1. Life event modeling

We model life events as bag of word vectors where each word vector has been derived from a Skip-gram model that we implemented using Deeplearning4j.[5] In order to run the Skip-gram model on our corpus, we used the default parameter settings of the Deeplearning4j library and set the layer size to 100, which means that the length of the word vectors will be equivalent to 100. Further, the learning rate is set to 0.025, the value of window size is set to 6 because the average number of words in a tweet is close to 6 based on an experiment that we did on 1 million tweets. Also, words are ignored if they have been observed less than 5 times.

5.3.2. Life event detection

In our experiments, we apply SVM, Random Forest and Gradient Boosting Tree classifiers implemented in LIBSVM,[6] Java Machine Learning[7] and Sklearn[8] packages, respectively. For SVM, the Radial Basis Function (RBF) kernel is used and the parameter c and gamma are set to 1 and 0.5, respectively. Learning rate and Maximum of depth of tree parameters are set to 0.01 and 3 respectively in Gradient Boosting Tree. For Random Forest, we set parameters to default values used in the Java-ML package.

5.4. Results and discussion

5.4.1. Life event detection

We used our ground truth annotations to evaluate the life event detection models with 10-fold cross-validation. That is, the annotated instances are randomly split into 10 subsets: 9 subsets were used to train a classifier and the remaining subset used as test data. The final result is averaged over 10 iterations so that each subset can be used as a test case once. We used three classifiers, Gradient Boosting Tree, Random Forest and Support Vector Machine. These classifiers took the word movers distance as the input feature set and classified a given tweet as either one of the six life events or no life event at all. Table 4 shows the precision, recall obtained for the classifiers individually for each life event. As shown in Table 4, there is a clear trade-off between precision and recall. RF and GBT produce better quality results compared to SVM; while there are no significant differences between RF and GBT. Given RF and GBT provide comparative and similar results, we adopt GBT to further compare our work with the baseline methods and also to build the temporal stacking model.

The results of comparing our proposed approach to the baseline proposed in [36] for each life event is reported in Table 5 in terms of precision, recall and F-score. As shown in this table, our layer 1 approach shows superior performance compared to the baseline in the F-score measure in all life event classes. However, the table also shows that the baseline has better precision in 5 out of the 6 life event classes.

As mentioned, the baseline method uses four features, namely word, named entity mentions, dictionary and context window. Among these features, named entity mentions and context window capture semantic and syntactic aspects of tweets respectively, while, word and dictionary models capture the textual aspects of tweets. One of the main reasons for the high precision of the baseline can be attributed to the top-40 word dictionary that is built based on topic models; however, this is also the explanation for the low recall of the baseline as it restricts its search space to the words in the dictionary.

Furthermore, the problem of detecting personal life event detection from a tweet can also be viewed as a text classification task. Therefore, we used a strong end-to-end deep learning based convolutional neural network model, known as Crepe, that has already shown to be an accurate text classifier as another baseline [62]. We train and evaluate the deep learning model based on a 10-fold cross validation strategy with 7 classes including six life event classes and one non-life event class. The results are included in Table 5, which shows that the performance of the end-to-end deep learning-based classifier is comparable to the baseline and

[5] http://deeplearning4j.org/.
[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[7] http://java-ml.sourceforge.net/.
[8] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.

Table 5
Comparison between our proposed approach and the baseline method inspired by [36]. All results are statistically significant at *p-value* < 0.01[9].

Life event/classifier		Baseline [36]	Crepe [62]	Our proposed approach
Broken Device	Precision	**0.67**	0.66	0.57
	Recall	0.33	0.22	**0.54**
	F-Score	0.44	0.327	**0.55**
Device Upgrade	Precision	0.70	0.29	**0.72**
	Recall	0.19	0.29	**0.32**
	F-Score	0.30	0.29	**0.44**
Moving	Precision	**0.67**	0.3	0.66
	Recall	0.24	0.15	**0.46**
	F-Score	0.35	0.2	**0.54**
New Job	Precision	0.13	**0.78**	0.76
	Recall	0.57	0.07	**0.65**
	F-Score	0.66	0.09	**0.70**
Travel	Precision	**0.79**	0.61	0.73
	Recall	0.59	0.19	**0.75**
	F-Score	0.68	0.29	**0.74**
Wedding	Precision	**0.63**	0.54	0.57
	Recall	0.04	**0.18**	0.11
	F-Score	0.07	**0.27**	0.18

Table 6
Comparison between our proposed approach and the baseline method inspired by Chaudury and Alani [18].

Life event/classifier		NER	Emoticons	HashTag	Hybrid of NER, Sentiment, Hashtag, Emoticons	Our approach
Broken Device	Precision	0.61	0.43	**0.82**	0.63	0.57
	Recall	0.35	0.001	0.05	0.39	**0.54**
	F-Score	0.45	0.001	0.09	0.48	**0.55**
Device Upgrade	Precision	0.64	0.2	**0.81**	0.72	0.72
	Recall	0.16	0.001	0.12	0.26	**0.32**
	F-Score	0.26	0.001	0.20	0.38	**0.44**
Moving	Precision	0.71	0.62	**0.86**	0.74	0.66
	Recall	0.1	0.01	0.02	0.14	**0.46**
	F-Score	0.17	0.02	0.04	0.24	**0.53**
New Job	Precision	0.66	0.48	**0.79**	0.7	0.76
	Recall	0.27	0.01	0.1	0.36	**0.65**
	F-Score	0.39	0.02	0.16	0.47	**0.7**
Travel	Precision	0.57	0.52	**0.84**	0.77	0.73
	Recall	0.06	0.025	0.14	0.2	**0.75**
	F-Score	0.10	0.05	0.24	0.32	**0.74**
Wedding	Precision	0.56	0.42	**0.82**	0.74	0.57
	Recall	0.09	0.01	0.13	**0.24**	0.21
	F-Score	0.15	0.03	0.22	**0.36**	0.31

our proposed approach in terms of precision; however, Crepe suffers from low recall rates similar to the baseline.[9]

Now as proposed in [18], we further build more baselines for comparison with our layer 1 work based on additional features not considered in [36] including NER, emoticons, hashtags and sentiment analysis. We use these features to learn gradient boosting trees and evaluate the developed models through a 10-fold cross-validation mechanism on the gold standard dataset. Table 6 shows the results of using the additional features to build life event classifiers and how their performance compares to our layer 1 approach. The results in this table provide significant insight into the effective features that produce high precision life event classification. As seen in the table, our layer 1 approach provides superior performance in terms of F-score on 5 out of 6 life events. Its recall is also reasonable and better than the other baselines in almost all life events. However, similar to the comparison with baseline from [36], our layer 1 method shows a lower precision for life event detection. When considering the different features, it is clear that hashtags when used as features provide the highest precision for detecting life events given they produced the highest precision in all six life events. In other words, hashtags are quite discriminative features for life event detection. However, they suffer from a very low recall. The interpretation of this would be that when available, hashtags are very strong indicators of life events but it turns out that such hashtags are not very frequently observed with *life event tweets* as shown in Table 6.

Furthermore, as seen in Table 6, sentiment based features are also reasonable indicators for life events. This feature provides balanced results for precision and recall for the various life events. One of the significant observations is that the sentiment-based feature produces the best F-score for the Wedding life event. This shows that when sentiments are abundant for a life event such as weddings, in contrast to other life events such as travel or a new job, sentiment features provide a strong indication of the life event.

Another important observation is that emoticons are also strong indicators of life events when present. The precision of the classifier learnt based on emoticons is reasonable but the recall rate is very low. Furthermore, it is important to mention named entity mentions as features. These features suffer from the recall problem similar to emoticons and hashtags. One of our observations is that although we used a Twitter specific API to identify and pick out named entity mentions, due to the informal and unique nature of tweets, the detection of the named entities is extremely difficult leading to the poorer recall rates.

[9] Given the implementations of these techniques are not available, we have implemented these methods according to the available publications and hence use the word ‘inspired’.

Table 7
Improvement of the performance of the features when our proposed embedding features is concatenated to the baseline features.

Life event/classifier		NER +Embed	Emoticons +Embed	HashTag +Embed	Hybrid +Embed	Our approach (Embed)
Broken Device	Precision	0.64	0.63	0.63	**0.64**	0.57
	Recall	0.58	0.58	**0.59**	0.58	0.54
	F-Score	**0.61**	**0.61**	**0.61**	**0.61**	0.55
Device Upgrade	Precision	0.72	0.72	**0.74**	**0.74**	0.72
	Recall	0.34	0.34	**0.36**	**0.36**	0.32
	F-Score	0.47	0.46	**0.49**	**0.49**	0.44
Moving	Precision	0.66	**0.67**	**0.67**	**0.67**	0.66
	Recall	0.45	0.46	0.44	**0.47**	0.46
	F-Score	0.53	0.54	0.53	**0.55**	0.53
New Job	Precision	0.77	**0.78**	0.77	**0.78**	0.76
	Recall	0.71	0.74	0.71	**0.75**	0.65
	F-Score	0.74	0.76	0.74	**0.77**	0.7
Travel	Precision	0.72	0.74	0.73	**0.75**	0.73
	Recall	**0.75**	0.74	0.74	**0.75**	0.75
	F-Score	0.73	0.74	0.74	**0.75**	0.74
Wedding	Precision	0.61	0.57	0.62	**0.64**	0.57
	Recall	0.12	0.16	0.17	**0.21**	0.21
	F-Score	0.19	0.25	0.27	**0.32**	0.31

We further performed additional experiments to see whether our proposed features based on the word mover's distance has synergistic impact on the other features included in Table 6. To do so, we retrained the models by concatenating our proposed features to the baseline features and the results are reported in Table 7. As seen in the table, when our proposed features is concatenated to the baseline features, the performance of the baseline features improves significantly especially as it pertains to the recall metric. When comparing Tables 6 and 7, it can be seen that the recall rates of the baseline features are extremely weak, while the recall rate of our proposed approach is strong. However, the inclusion of our feature with the baseline features substantially improves the recall rates. In conclusion, the Hybrid + Embed approach that includes NER, Sentiments, Emoticons, Hashtags as well as our proposed features (Embed) shows the best performance across all life event types.

5.4.2. *Temporal stacking*

Now, we temporally stack layer 1 classifier over past and future tweets. In other words, we apply the layer 1 classifier over the tweets from one week before and after the given tweets posting date and use the outcome as the feature set for training a second classifier in the second layer, a process which we refer to as temporal stacking. Based on the performance of the Gradient Boosted Tree in building a more effective classifier in the first step, we have adopted it for training the temporal stacking classifier.

In addition to performing a temporal stacking on our layer 1 approach, we also perform stacking on the baselines introduced in Table 6, to see if temporal stacking also provides improvement on those models or not. We report the results based on a 10-fold cross validation scheme in Fig. 5(a)–(c). These figures show to what extent precision, recall and F-score changed when temporal stacking was applied. It should be noted that the results reported in these three figures are *statistically significant* compared to the results reported in Table 6 (before temporal stacking) for all life events and all classifiers with a *p-value* < 0.01.

The first observation is based on the results in Fig. 5(a), which shows that temporal stacking, when applied on features such as Hashtags and NER, can result in lower precision. In other words, when a larger number of tweets from before and after a certain tweet of interest are taken into consideration, the chances of accurately predicting the correct life event for the tweet of interest reduces. This might be explained by the fact that the likelihood of observing discriminatory features for the life events increases, which might not necessarily all be related to the tweet of interest. While this observation is consistent for the NER, HashTag and hybrid features, it is different for the Emoticon and WMD features. When looking into the reason for this, we found that given the fact that the WMD feature only relies on the semantic similarity between the observed words and the representation of a life event to determine the correct life event, it is more resilient towards unrelated words when pooled through temporal stacking. For instance, if several tweets are pooled in the temporal stacking process and there are a few tweets that are discussing unrelated issues to the topic of the tweet of interest, then as long as the other tweets are related, WMD is able to benefit from the related tweets as it calculates semantic similarity based on word mover's distance that maximizes similarity to the closest words.

On the other hand, this does not apply to features such as Named Entities and Hashtags when considering two facts simultaneously: 1) Such features are not very common in tweets (low recall observed in Table 6); 2) When present these features are highly discriminatory (high precision in Table 6). In simple terms, if an instance of such features is observed, it will serve as a strong indicator for a certain life event. Therefore, these features, unlike WMD, are sensitive features that would reduce precision if inaccurate values for the features are included. For this reason, when applied in temporal stacking and given the fact that information from related tweets are used in the temporal stacking process, the precision will drop. In order to show the number of values obtained for these features through temporal stacking, we report the percentage of users that had relevant values for each feature before and after temporal stacking in Table 8. For instance, within the 'broken device' life event, only 2% of the users had Hashtags while this number reached 15% when temporal stacking was applied showing a 13% increase in the number of Hashtags that were considered. While this will have a positive impact on recall, as we will show later, it negatively impacts precision.

Finally, we also looked at the reason why emoticons, unlike NER and Hashtags, show improvement on precision after temporal stacking. We found that this is due to the fact that when used, users adopt a consistent set of emoticons in their tweets. However, it is worth noting that the improvement of precision as a result of temporal stacking on emoticons is quite small ~ 0.05. Therefore, our finding shows that temporal stacking positively impacts the precision of models learnt based on semantic features such as our proposed WMD feature but can negatively impact models primarily built on features such as NER and Hashtags.

The second set of observations relate to the increase in recall shown in Fig. 5(b). As seen in the figure, given additional values for

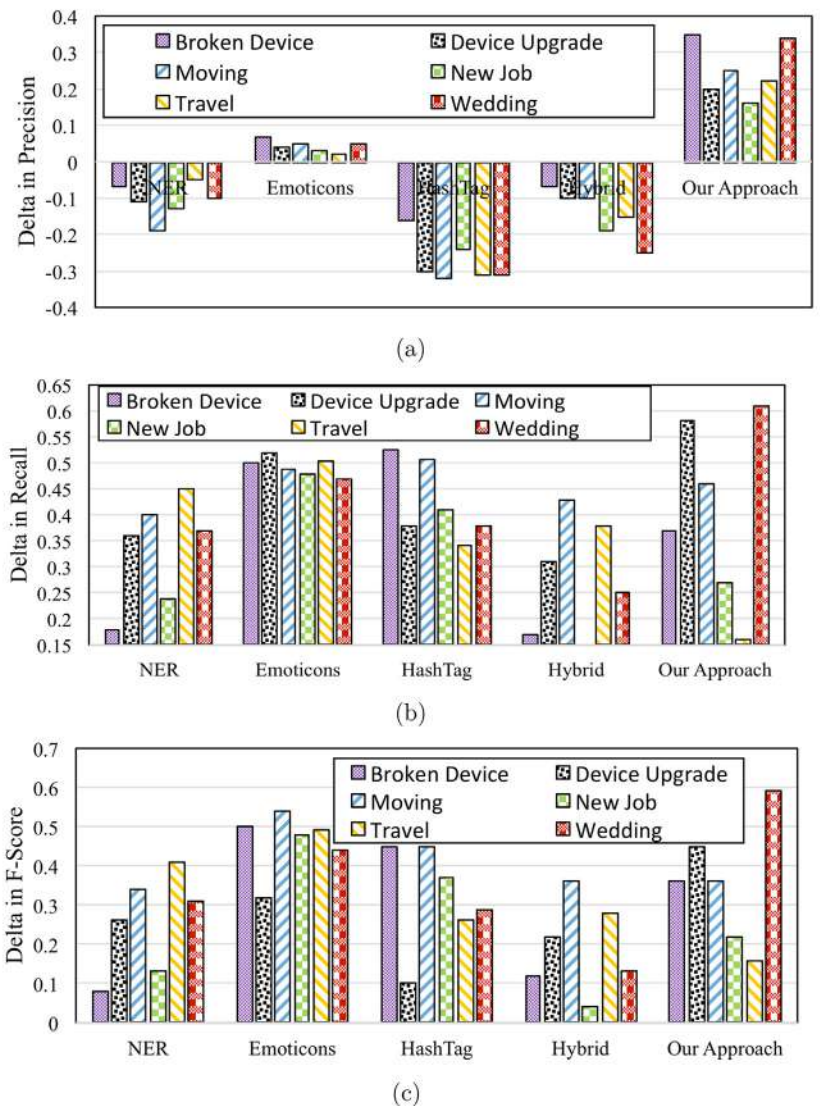


Fig. 5. Changes in (a) Precision, (b) Recall, and (c) F-score after temporal stacking.

Table 8
The percentage of users with a given feature before and after temporal stacking.

	Broken Device		Device Upgrade		Moving		New Job		Wedding		Travel	
	Bef	Aft	Bef	Aft	Bef	Aft	Bef	Aft	Bef	Aft	Bef	Aft
Emoticon	0.26	0.85	0.2	0.86	0.32	0.81	0.22	0.63	0.21	0.61	0.28	0.67
HashTag	0.02	0.15	0.09	0.2	0.06	0.17	0.02	0.17	0.16	0.42	0.08	0.32
NER	0.9	0.99	0.91	0.99	0.91	0.99	0.96	0.99	0.95	0.99	0.4	0.92

all the features are pooled through temporal stacking (also seen in Table 8), recall increases significantly. Compared to the recall values obtained in the first layer classifier, these improved recall values are important as they allow us to predict the correct life event of a larger number of tweets. For instance, the recall rates of the Emoticons feature on the ‘Broken Device’ and ‘Device Upgrade’ life events are 0.001 in the first layer life event classifier, which means that out of 5,768 and 2,822 tweets in these life events, respectively, only between $5-6$ and $2-3$ tweets would have been recalled by the first layer classifier, which is quite insignificant. However, based on temporal stacking and for the same life events and feature, recall has increased by $\sim$50% covering up to 2, 884 and 1,411 tweets, respectively. This can be explained by observing in Table 8 that for the same feature and life events, the percentage of users that had a value for the emoticon feature increased from 26% and 20% to 85% and 86%, respectively after temporally stacking, leading to increased recall values. As discussed in Tables 5 and 6, the main issue with the first layer classifiers was low recall. Based on the results obtained, the recall values have significantly increased, albeit at the cost of losing some precision in the case of Hashtag, NER and hybrid features.

In order to evaluate the overall impact of temporal stacking on life event detection, we report the delta of F-score before and after temporal stacking in Fig. 5(c). The results show that temporal stacking has resulted in an increase in performance over the first layer classifiers across all features and for all different types of life events.

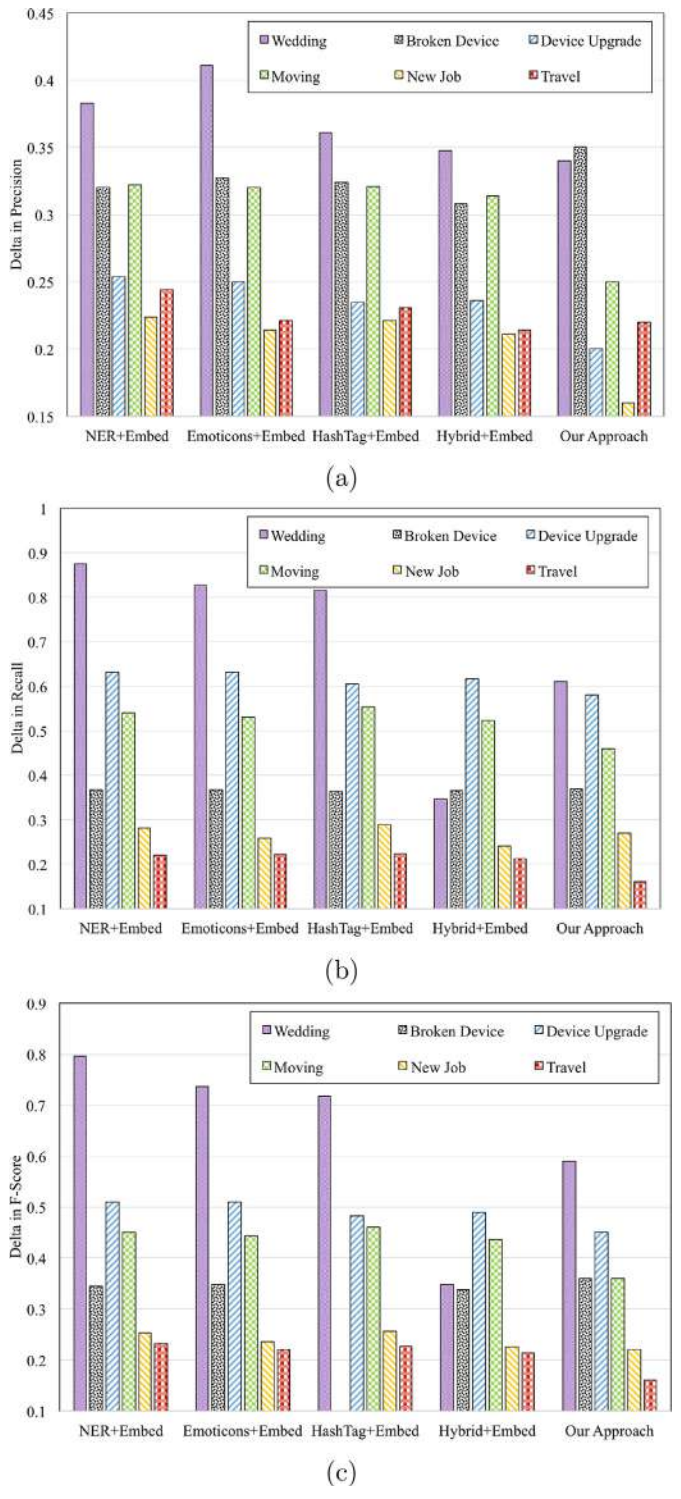


Fig. 6. Changes in (a) Precision, (b) Recall, and (c) F-score after temporal stacking when our proposed metric is included with the baseline features.

Furthermore, in order to validate the above arguments in terms of the impact and role of our proposed embedding feature in terms of robustness for recall and precision, we report the performance of the baseline models before and after temporal stacking for the case when they are trained jointly with our proposed features. The results are reported in Fig. 6. As expected, when our proposed features is included along with the baseline features, the observed improvement is consistently positive for all three metrics across all life event types.

Finally, an important consideration for life event detection could be the need to perform classification in realtime. Therefore, if this is the case the tweets from the next seven days (one week) will not be available for the sake of classification. In this case, the only available information is the tweets from the past time intervals before the tweet of interest was posted. Based on this, we performed experiments to see the impact of 2 factors on the performance of the model: (1) The consideration of only past tweets; and (2) The length of the considered time window. Fig. 7 summarizes our findings in terms of the difference of the F-score of our proposed approach for different window sizes. In this figure, zero days shows the case when no future tweets are taken into account for detecting the life event and only past tweets are considered. The other three intervals *two, four* and *six* show cases when tweets of only two, four and six future days are taken into consideration. The y-axis shows the percentage of change in the F-score of the trained models. As seen, the more future tweets are taken into account, the more accurate the model will be; however, the amount of improvement is not significant and hence a model trained only based on past tweets would be practical for realtime life event detection.

It should be noted that the performance of the temporal stacking model relies on the availability of other tweets other than the tweet of interest; therefore, if such tweets (e.g., those from the past week for that user) are not available due to reasons such as limitation of the Twitter API, then temporal stacking would not be possible. However, to date, there are currently no such limitations in place and it is possible to retrieve the required tweets for temporal stacking from Twitter.

5.4.3. *Execution performance*

The other performance consideration that we explored was the execution time of our proposed method compared to the state of the art methods[10] This is an important issue given the fact that tweets need to be processed and labeled in real-time and slow methods would not be practical. The average and standard deviation of the execution time of the different methods in first layer are reported in Table 9. It can be seen that the execution time of our proposed approach is among the fastest compared to the other methods and is comparable to when only simple features such as emoticons and hashtags are used. However, the slowest execution time belongs to the work proposed by Li et al. [36], which is around 33 seconds per classification, making it too slow to be practical.

We further explored whether the execution time of the second layer classifiers (temporal stacking) was different for the various classifiers. Our observation was that second layer classifiers all had the same execution time with an average of 0.002 seconds regardless of which first layer classifier was used. This is primarily due to the fact that the second layer classifier does not rely on the performance of the first layer classifier and only classifies a tweet based on a set of input labels from the first layer. So once those labels are provided, all the models have the same execution time in terms of temporal stacking. It should be noted that the set of tweets used in temporal stacking for the different models was exactly the same (therefore, the same number of tweets) and hence making the execution times fully comparable.

[10] All our experiments were executed on 40-core Intel(R) Xeon(R) E5-2690 v2 @ 3.00GHz with 256GB of memory.

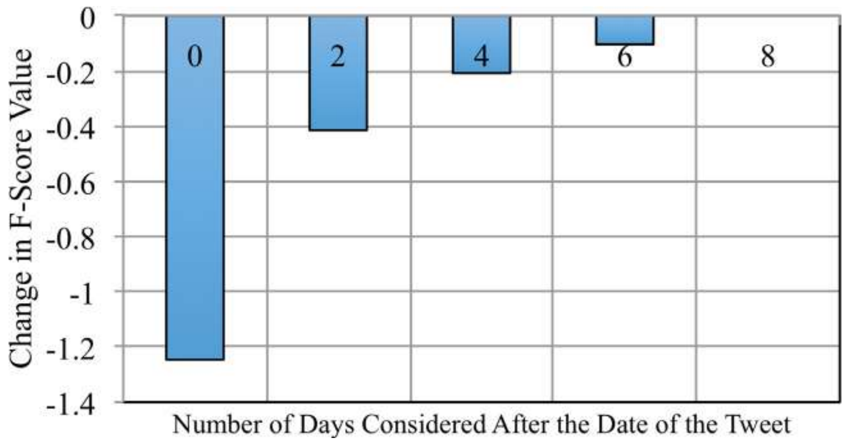


Fig. 7. Impact of different window sizes on performance.

Table 9
The execution time of the first layer classifiers using different approaches. Avg and Std represent average and standard deviation of the execution time in seconds. W/O E and WE denote Without Embedding features and With Embedding features, respectively.

	NER		Emoticons		Hashtags		Hybrid		Li et al.	Ours
	W/O E	WE	W/O E	WE	W/O E	WE	W/O E	WE		
Avg	0.743	0.788	0.486	0.858	0.504	0.768	0.765	0.801	33.681	0.662
Std	0.0006	0.001	0.0002	0.002	0.0003	0.001	0.0007	0.001	2.763	0.0005

6. Concluding remarks

In this paper, we have proposed two incremental methods for the detection of life event mentions on microblogging platforms, most specifically Twitter. We have advanced the state of the art methods for identifying life event mentions on Twitter by using a semantic feature learnt based on the distance between embedding-based life event representations and the vector representation of a tweet calculated using Word Movers Distance (WMD). We have shown that multi-class classifiers learnt using this feature outperform existing techniques in terms of both recall and F-score; however, they are not as efficient from a precision stand-point. To address, this shortcoming, we have proposed the novel idea of temporally stacking the results on an existing life event classifier over the tweets from several time intervals before and after the tweet of interest. The predicted life events for these tweets obtained from existing life event classifiers, such as our proposed classifier based on WMD, are then used as input features for a second-layer classifier, which we refer to as the stacked classifier. Through our experiments, we have shown that the stacked classifier shows impressive improvement in all recall, precision and F-score metrics and outperforms all methods before being stacked. This is an indication that using a weaker life event classifier for labeling a sequence of tweets leads to a highly accurate model for life event detection.

As a part of future work, we are interested in extending our work in the following two directions:

- One specific area that we are interested in is to predict future life events based on a given user's historical social network activity. For instance, we are interested in predicting whether a given Twitter user will get married or change jobs in the next few weeks without having access to the tweets that include such announcements. A possible approach for addressing this problem is to consider the historical content of the user as well as the content generated by her connection. A natural approach for addressing is to use recurrent neural networks based on long short-term memory networks.
- We are also interested in addressing the same problem of predicting future life events by applying frequent sequence mining techniques on a global set of life events. This could give us an indication of how users transition through life events and what are the likelihood of observing a next type of life event given the observation of a sequence of life events.
- Finally, the work in this paper could be generalized to detecting *local events* on Twitter that are not necessarily personal and dedicated to only one individual user. For instance, the proposed approach could be used to detect a *sale* or *promotion* at a local store that makes the announcement on Twitter. The characteristic of such events is similar to life events and hence the work presented in this paper has the potential to be applied in those contexts as well. As future work, we are interested in collecting tweets that could be used to evaluate our work in such context.

References

[1] H. Abdelhaq, C. Sengstock, M. Gertz, Eventweet: online localized event detection from twitter, Proc. VLDB Endow. 6 (12) (2013) 1326–1329, doi:10.14778/2536274.2536307.
[2] M. Adedoyin-Olowe, M.M. Gaber, F. Stahl, J.B. Gomes, Autonomic Discovery of News Evolvement in Twitter, in: Big Data in Complex Systems, Springer, 2015, pp. 205–229.
[3] L.M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, I. Kompatsiaris, A. Jaimes, Sensing trending topics in twitter, Trans. Multi. 15 (6) (2013) 1268–1282, doi:10.1109/TMM.2013.2265080.
[4] J. Allan, Introduction to Topic Detection and Tracking, in: Topic detection and tracking, Springer, 2002, pp. 1–16.
[5] J. Allan, J.G. Carbonell, G. Doddington, J. Yamron, Y. Yang, Topic detection and tracking pilot study final report, in: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1998.
[6] F. Atefeh, W. Khreich, A survey of techniques for event detection in twitter, Comput Intell 31 (1) (2015) 132–164.
[7] H. Becker, M. Naaman, L. Gravano, Learning similarity metrics for event identification in social media, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, in: WSDM '10, ACM, New York, NY, USA, 2010, pp. 291–300, doi:10.1145/1718487.1718524.
[8] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (Feb) (2003) 1137–1155.
[9] D.B. Bracewell, Long nights, rainy days, and misspent youth: automatically extracting and categorizing occasions associated with consumer products, SocialNLP 2015@ NAACL (2015) 29.

[10] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[11] C. Castillo, M. Mendoza, B. Poblete, Predicting information credibility in time-sensitive social media, Internet Res. 23 (5) (2013) 560–588, doi:10.1108/IntR-05-2012-0095.
[12] P.R. Cavalin, F. Dornelas, S.M. da Cruz, Classification of life events on social media, in: 29th SIBGRAPI (Conference on Graphics, Patterns and Images), 2016.
[13] P.R. Cavalin, M. Gatti, C.S. Pinhanez, Towards personalized offers by means of life event detection on social media and entity matching., HT (Doctoral Consortium/Late-breaking Results/Workshops), 2014.
[14] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[15] X. Chang, Y.-L. Yu, Y. Yang, E.P. Xing, Semantic pooling for complex event analysis in untrimmed videos, IEEE Trans. Pattern Anal. Mach. Intell. 39 (8) (2017) 1617–1632.
[16] L. Chen, A. Roy, Event detection from flickr data through wavelet-based spatial analysis, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, in: CIKM '09, ACM, New York, NY, USA, 2009, pp. 523–532, doi:10.1145/1645953.1646021.
[17] S. Choudhury, H. Alani, Detecting presence of personal events in twitter streams, in: International Conference on Social Informatics, Springer, 2014, pp. 157–166.
[18] S. Choudhury, H. Alani, Personal life event detection from social media, in: 25th ACM Hypertext and Social Media Conference., CEUR, 2014.
[19] P. Dhillon, D.P. Foster, L.H. Ungar, Multi-view learning of word embeddings via cca, in: Advances in Neural Information Processing Systems, 2011, pp. 199–207.
[20] B. Di Eugenio, N. Green, R. Subba, Detecting life events in feeds from twitter., in: ICSC, 2013, pp. 274–277.
[21] T. Dickinson, M. Fernandez, L.A. Thomas, P. Mulholland, P. Briggs, H. Alani, Identifying prominent life events on twitter, in: Proceedings of the 8th International Conference on Knowledge Capture, ACM, 2015, p. 4.
[22] W. Dou, X. Wang, D. Skau, W. Ribarsky, M.X. Zhou, Leadline: Interactive visual analysis of text data through event identification and exploration, in: Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on, IEEE, 2012, pp. 93–102.
[23] P.S. Earle, D.C. Bowden, M. Guy, Twitter earthquake detection: earthquake monitoring in a social world, Annals Geophys. 54 (6) (2012).
[24] J. Eisenstein, What to do about bad language on the internet., in: HLT-NAACL, 2013, pp. 359–369.
[25] A.M. Ertugrul, B. Velioglu, P. Karagoz, Word embedding based event detection on social media, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2017, pp. 3–14.
[26] H. Fani, E. Bagheri, F. Zarrinkalam, X. Zhao, W. Du, Finding diachronic like-minded users, Comput. Intell. (2017 (in press)). http://onlinelibrary.wiley.com/journal/10.1111/(ISSN)1467-8640.
[27] Y. Feng, H. Fani, E. Bagheri, J. Jovanovic, Lexical semantic relatedness for twitter analytics, in: Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on, IEEE, 2015, pp. 202–209.
[28] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.
[29] J.H. Friedman, Recent advances in predictive (machine) learning, J. Classif. 23 (2) (2006) 175–197.
[30] J. Glück, S. Bluck, Looking back across the life span: a life story account of the reminiscence bump, Memory Cognit. 35 (8) (2007) 1928–1939.
[31] M. Hernandez, K. Hildrum, P. Jain, R. Wagle, B. Alexe, R. Krishnamurthy, I.R. Stanoi, C. Venkatramani, Constructing consumer profiles from social media data, in: Big Data, 2013 IEEE International Conference on, IEEE, 2013, pp. 710–716.
[32] H.H. Khondker, Role of the new media in the arab spring, Globalizations 8 (5) (2011) 675–679.
[33] M.J. Kusner, Y. Sun, N.I. Kolkin, K.Q. Weinberger, From word embeddings to document distances, in: Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), 2015, pp. 957–966.
[34] H. Kwak, C. Lee, H. Park, S. Moon, What is twitter, a social network or a news media? in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 591–600.
[35] R. Lee, K. Sumiya, Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection, in: Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks, ACM, 2010, pp. 1–10.
[36] J. Li, A. Ritter, C. Cardie, E.H. Hovy, Major life event extraction from twitter based on congratulations/condolences speech acts., in: EMNLP, 2014, pp. 1997–2007.
[37] H. Lin, J. Jia, L. Nie, G. Shen, T.-S. Chua, What does social media say about your stress? in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 2016.
[38] X. Liu, B. Huet, Heterogeneous features and model selection for event-based media classification, in: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval, ACM, 2013, pp. 151–158.
[39] Z. Ma, Y. Yang, N. Sebe, A.G. Hauptmann, Knowledge adaptation with partially shared features for event detection using few exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 36 (9) (2014) 1789–1802.
[40] E. Meij, W. Weerkamp, M. de Rijke, Adding semantics to microblog posts, in: Proceedings of the fifth ACM international conference on Web search and data mining, ACM, 2012, pp. 563–572.
[41] P. Meladianos, G. Nikolentzos, F. Rousseau, Y. Stavrakas, M. Vazirgiannis, Degeneracy-based real-time sub-event detection in twitter stream, in: Ninth International AAAI Conference on Web and Social Media, 2015, pp. 248–257.
[42] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: 1301.3781 (2013).
[43] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations., in: HLT-NAACL, 13, 2013, pp. 746–751.
[44] L.G. Moyano, P.R. Cavalin, P.P. Miranda, Life event detection using conversations from social media, Brazilian Workshop on Social Network Analysis and Mining, 2015.
[45] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, N.A. Smith, Improved part-of-speech tagging for online conversational text with word clusters, Assoc. Comput. Linguistics (2013).
[46] M. Pennacchiotti, A.-M. Popescu, Democrats, republicans and starbucks afficionados: user classification in twitter, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 430–438.
[47] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation., in: EMNLP, 14, 2014, pp. 1532–1543.
[48] G. Petkos, S. Papadopoulos, Y. Kompatsiaris, Social event detection using multimodal clustering and integrating supervisory signals, in: Proceedings of the 2Nd ACM International Conference on Multimedia Retrieval, in: ICMR '12, ACM, New York, NY, USA, 2012, pp. 23:1–23:8, doi:10.1145/2324796.2324825.
[49] S. Phuvipadawat, T. Murata, Breaking news detection and tracking in twitter, in: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, 3, IEEE, 2010, pp. 120–123.
[50] A.-M. Popescu, M. Pennacchiotti, Detecting controversial events from twitter, in: Proceedings of the 19th ACM international conference on Information and knowledge management, ACM, 2010, pp. 1873–1876.
[51] A.-M. Popescu, M. Pennacchiotti, D. Paranjpe, Extracting events and event descriptions from twitter, in: Proceedings of the 20th international conference companion on World wide web, ACM, 2011, pp. 105–106.
[52] A. Ritter, S. Clark, O. Etzioni, et al., Named entity recognition in tweets: an experimental study, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 1524–1534.
[53] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1–2) (2010) 1–39.
[54] T. Sakaki, M. Okazaki, Y. Matsuo, Tweet analysis for real-time event detection and earthquake reporting system development, IEEE Trans. Knowl. Data Eng. 25 (4) (2013) 919–931.
[55] R. Troncy, B. Malocha, A.T. Fialho, Linking events with media, in: Proceedings of the 6th International Conference on Semantic Systems, ACM, 2010, p. 42.
[56] F. Vis, Twitter as a reporting tool for breaking news: journalists tweeting the 2011 uk riots, Digital Journalism 1 (1) (2013) 27–47.
[57] Q. Wang, Z. Lin, Y. Jin, S. Cheng, T. Yang, Esis: emotion-based spreader–ignorant–stifler model for information diffusion, Knowl. Based Syst. 81 (2015) 46–55.
[58] W. Weerkamp, M.d. Rijke, et al., Activity prediction: A twitter-based exploration, SIGIR Workshop on Time-aware Information Access, 2012.
[59] J. Weng, B.-S. Lee, Event detection in twitter., ICWSM 11 (2011) 401–408.
[60] L. Xie, H. Sundaram, M. Campbell, Event mining in multimedia streams, Proc. IEEE 96 (4) (2008) 623–647.
[61] Y. Yang, J.G. Carbonell, R.D. Brown, T. Pierce, B.T. Archibald, X. Liu, Learning approaches for detecting and tracking news events, IEEE Intell. Syst. 14 (4) (1999) 32–43, doi:10.1109/5254.784083.
[62] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. (2015) 649–657.