Journal of Intelligent Information Systems (2020) 54:101–127
https://doi.org/10.1007/s10844-018-0519-2

Predicting future personal life events on twitter via recurrent neural networks

Maryam Khodabakhsh[1] · Mohsen Kahani[1] · Ebrahim Bagheri[2]

Received: 3 October 2017 / Revised: 7 April 2018 / Accepted: 18 July 2018 /
Published online: 15 August 2018

Abstract

Social network users publicly share a wide variety of information with their followers and the general public ranging from their opinions, sentiments and personal life activities. There has already been significant advance in analyzing the shared information from both micro (individual user) and macro (community level) perspectives, giving access to actionable insight about user and community behaviors. The identification of personal life events from user's profiles is a challenging yet important task, which if done appropriately, would facilitate more accurate identification of users' preferences, interests and attitudes. For instance, a user who has just *broken his phone*, is likely to be upset and also be looking to purchase a new phone. While there is work that identifies tweets that include mentions of personal life events, our work in this paper goes beyond the state of the art by predicting a future personal life event that a user will be posting about on Twitter solely based on the past tweets. We propose two architectures based on *recurrent neural networks*, namely the classification and generation architectures, that determine the future personal life event of a user. We evaluate our work based on a gold standard Twitter life event dataset and compare our work with the state of the art baseline technique for life event detection. While presenting performance measures, we also discuss the limitations of our work in this paper.

Keywords Life event prediction · Recurrent neural networks · Social networks · Twitter

✉ Mohsen Kahani
kahani@um.ac.ir

Maryam Khodabakhsh
maryamkhodabakhsh@mail.um.ac.ir

Ebrahim Bagheri
bagheri@ryerson.ca

[1] Web Technology Laboratory, Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

[2] Laboratory for Systems, Software and Semantics (LS3), Ryerson University, Toronto, Ontario, Canada

1 Introduction

Social microblogging platforms, such as Twitter, are accessible information sharing environments where individuals, governments and even multi-national corporations share their thoughts, views, sentiments and recent news. The live stream of content that is generated and shared on these platforms serve as a valuable source of information for performing macro and micro-level analyses ranging from the understanding of individual users' interests and preferences (Zarrinkalam et al. 2016; Zhao et al. 2016) to detecting and monitoring the formation and evolution of user communities (Fani et al. 2016, 2017a, b) as well as identifying important trending topics (Madani et al. 2015; Kaleel and Abhari 2015) and how they can enable descriptive and prescriptive analytics (Fan and Gordon 2014). Researchers have already extensively explored the possibility of using socially shared data to build predictive models. For instance, work has been done on using social data to offer macro-scale predictions on dependent variables such as stock market prices (Nguyen et al. 2015; Nguyen and Shirai 2015; Mao et al. 2012; Bollen et al. 2011; Nofer and Hinz 2015), movie sales (Liu et al. 2016; Choudhery and Leung 2017), disease outbreak (Woo et al. 2016; Byrd et al. 2016) and election results (Franch 2013; Tsakalidis et al. 2015; Cameron et al. 2016), just to name a few. There has also been interesting studies showing that with even scant amount of information from individual users on the social network, it is possible to perform micro-level predictions such as detecting psychopathy (Wald et al. 2012; Preotiuc-Pietro et al. 2015), anxiety and depression (DeChoudhury et al. 2013; Coppersmith et al. 2015).

One of the main focal research areas on social network data is to derive such *actionable insights* in order to facilitate decision making. To this end, detecting and monitoring events on the social network has been of interest as trends and events on social networks have shown to reflect many of the characteristics of the events of the real world (Unankard et al. 2015; Paltoglou 2016). Most of the techniques that focus on the detection of trending topics and events on social networks are based primarily on the hypothesis that trending social topics or events can be characterized by the change in observation frequency. In other words, if a certain topic is frequently mentioned in a certain time period, while the topic had been dormant in the past time periods, it can be considered to be a reflection of an event happening in the real world or a trending issue. Therefore, the success of such *event detection* models relies on the global monitoring of social content across, at least, a subsection of the broad social network.

More recently, researchers have become interested in studying a more nuanced form of events that are specific to a given user, known as *personal life events*. Personal life events, also known as life events, relate to an event or incident that happened for an individual social network user that was then publicly reported by this user on the social network. Life events can include getting engaged, moving to a new house, buying a new phone, and graduating from college, among others. If detected, personal life events can provide a good basis for making recommendations to users. For instance, it would be possible to recommend new phone products on the market to a user who has reported a personal life event related to breaking his/her phone. As another example, it would be possible to recommend relevant career opportunities for a user who is engaged with the life event of changing jobs.

The detection of life events has a different set of challenges compared to the detection of trending topics and global events as these events are only mentioned locally by the user him/herself and at most reacted to by some close social connections. Therefore, methods that rely on changes in content frequency would not be suitable for this purpose. For instance, a user would tweet *'I just got married thanks monbebes for tuning into our wedding ceremony'*; reporting that she/he has gotten married. This tweet was liked three

times and re-tweeted only once; therefore, it is quite difficult to determine its significance and importance. Furthermore, social network users use special jargon to report life events that are quite hard to pick out and identify. As an example, a Twitter user posts *'i don't miss being alone!'* and posts a picture of his engagement party. It is difficult to determine this life event given the brevity and rather cryptic nature of the message. Moreover, there are many messages posted on Twitter that are similar to life events but are in fact information sharing or advertising content. A user posting *'That I'll get hoodwinked into a 2nd marriage.'* is not reporting a life event but rather expressing his/her opinion about a topic that has a similar representation to a life event. Finally, the high *class imbalance* of reported personal life events compared to non-life event content makes the detection of life event messages a difficult task. Existing work (Dickinson et al. 2015; Cavalin et al. 2014; Choudhury and Alani 2014a; Li et al. 2014) already extensively explore various manually curated features such as syntactic, semantic, sentiment as well as behavioral features to build classification models that can determine whether one or a collection of tweets are discussing a personal life event.

The objective of this paper is to move beyond the state of the art by attempting to predict *if and what* personal life event a given user will discuss in a future time interval by solely relying on and processing the user's historical tweets. In other words, in contrast to earlier work, this paper presents several methods for anticipating whether a given user will report on or mention a certain personal life event in her future tweets by considering previous tweets. For instance, by considering a past tweet from a Twitter user, such as *'I do not understand why planes don't board from the back first. That would save so much time and frustration. Is that not logical?!'*, our work would determine that the user will be reporting on a *Travel* related life event in the future; however, without having access to the future tweet that mentions *'Left Chicago today & moved into new NYC apartment'*. From an objectives perspective, the main differentiating aspect of our work from the literature is that earlier work always benefit from the tweet that has reported the personal life event as one of the inputs to their classification model; however, our work performs predictive analytics based on past tweets assuming that the tweet that will be reporting the personal life event has not yet been posted by the user.

In this paper, we view the problem of predicting a user's future life event as a sequence generation process where the objective is to process a sequence of past tweets and generate a future sequence that would determine the future personal life event of the user through our proposed variants of recurrent neural networks. Succinctly, the contributions of this paper are as follows:

- We propose two variants of a classification architecture based on recurrent neural networks that consider a sequence of input life events and a sequence of tweets, respectively and directly generate an output label that represents a future life event;
- In addition, we introduce a sequence generation architecture based on recurrent neural networks that considers past tweets and their life event labels *in tandem* to generate an output sequence resembling a future tweet, which would then be used to determine the future life event;
- Our proposed work is evaluated on a gold standard dataset that is systematically collected from Twitter and has real-world characteristics such as having a high class imbalance. We further discuss examples of where our methods fail to identify future life events and elaborate on the reasons for them.

The rest of this paper is organized as follows: In the next section, we classify the related work in three subsections covering pertinent work in event prediction, life event detection and deep learning techniques applied on social content. In Section 3, a high-level overview

of our proposed approach is presented, which is then followed by an in-depth presentation of the details of the work in Section 4. The details of our experiments, gold standard dataset, measurement metrics, baselines and our findings are presented in Section 5. The next section is dedicated to error analysis and the paper is finally concluded in Section 7.

2 Related work

The prior related work to this paper can be classified broadly into three categories, namely event prediction, personal life event detection, and deep learning methods for social content. We present work in these areas in the following subsections.

2.1 Event prediction

Users' behavior on social networks is often a reflection of the events and emotions that they experience in the real world. As such researchers have been interested to use social content in order to model user behavior and make prediction on that basis. A wide range of prediction models based on Twitter content have already been developed that span several domains including traffic (Ni et al. 2014), election (Franch 2013; Tsakalidis et al. 2015; Cameron et al. 2016), healthcare (Eichstaedt et al. 2015; Woo et al. 2016; Byrd et al. 2016), and the stock market (Nguyen et al. 2015; Nguyen and Shirai 2015; Mao et al. 2012; Bollen et al. 2011; Nofer and Hinz 2015). For instance, Tsakalidis et al. (2015) relied on Twitter content to predict the outcome of elections in the EU (Germany, Netherlands and Greece). Their model included eleven features based on Twitter, such as the number of times that different parties were mentioned on Twitter. In addition, they used sentiment analysis to assign a sentiment value to each tweet. They found that the use of Twitter features leads to statistically significant better results compared to just using poll information. Cameron et al. (2016) found that the size of candidates' network, such as the number of followers on Twitter and friends on Facebook, is not a good predictor of election results.

Mao et al. (2012) studied whether the daily number of tweets can predict the S&P 500 stock indicators while the work in Bollen et al. (2011) employed sentiment features for predicting stock market trends. Nguyen et al. (2015) proposed a model for predicting the stock market by capturing company and topic specific sentiments. They proposed a novel feature called the topic-sentiment which represents the sentiments towards specific company topics. Nofer and Hinz (2015) considered the number of Twitter followers in their analysis in addition to sentiments because they hypothesized that the importance of every tweet depends on the number of users recognizing the original message.

Liu et al. (2016) used bag of words, mentions, presence of a URL, emoticons, tweet length and trigger word features to predict box-office revenues for movies prior to their release. Choudhery and Leung (2017) presented a social data mining system that takes into account the number of tweets per day, as well as percentages of positive and negative tweets (based on sentiment analysis on those tweets) to build a polynomial regression model to predict the expected box office revenue. Based on the features of the social graph and social text content, Lassen et al. (2014) developed a linear regression model that transforms iPhone-related tweets into an accurate prediction of the quarterly iPhone sales with an average error close to the established prediction models from investment banks. Radosavljevic et al. (2014) developed a model called *Goalr* that predicts the World Cup outcomes and the relative strength of each country based solely on Tumblr blog posts. They used both team

mentions and player mentions in their analysis. In all above methods, feature extraction from social content is a necessary task for building the prediction model.

Byrd et al. (2016) collected a set of tweets based on influenza related keywords, classified them based on their sentiment characteristics and identified Twitter users who are affected by influenza in selected cities. Eichstaedt et al. (2015) used language expressed on Twitter to show that community-level psychological features correlate with age-adjusted mortality from atherosclerotic heart disease (AHD). They identified that the language pattern showing negative emotions are risk factors for AHD whereas as positive emotions act as a protective factor against AHD.

2.2 Personal life event detection

Most of the earlier work that have focused on the detection of life events from Twitter employ either an individual tweet (DiEugenio et al. 2013; Dickinson et al. 2015; Choudhury and Alani 2014b; Cavalin et al. 2014) or a stream of conversations (Cavalin et al. 2015; Moyano et al. 2015). One of the widely used approaches for detecting life events is based on the analysis of individual tweets with the objective to process the content of the given tweet using natural language processing (NLP) techniques to explore whether a mention of a certain life event can be detected or not. However, this task can be very challenging due to noisy, ambiguous and the short length of tweets. The use of conversations; however, can improve the performance of life event detection methods. Briefly, conversations can not only help identify events with a higher *precision*, but can also be a way to infer additional information that might not have been present in the original tweet. To this end, Choudhury and Alani (2014a) used the collection of a user's tweets within a specific time interval in order to detect whether a life event has been reported by the user in that time or not. However, in their work, they do not detect the type of the life event and only resort to detecting whether a life event has been mentioned.

Regardless of whether one tweet or a collection of tweets are selected, the primary focus of the work in the literature has been to extract strong discriminative features from tweets in order to identify life events. One of the most common features is the bag of words feature that models tweets as a vector whose entries are nonzero if the corresponding terms appear in the tweet. DiEugenio et al. (2013) and Dickinson et al. (2015) have explored various features and found that n-grams (an extension of the bag of words approach) show the highest performance. However, this method suffers from the *curse of dimensionality* when the vocabulary size is large. This is particularly the case for Twitter content due to the large number of slangs, acronyms, abbreviations, and misspellings that are predominantly observed in tweets. Furthermore, the temporal ordering of words and the semantic and syntactic features of the text, e.g., named entities and part of speech tags, are overlooked in this approach. Several researchers have explored the impact of linguistic and structural features such as part of speech tagging (DiEugenio et al. 2013; Li et al. 2014) and reported that this feature is not sufficient alone because few users actually observe grammatical structure on Twitter given the informal communication style.

Another group of features that have shown to be useful are based on the so called named entity vectors, which attempt to extract information by answering the 4W questions: who, what, when, and where. To this end, named entity extraction (Dickinson et al. 2015) and semantic role labeling (DiEugenio et al. 2013) techniques have been used to generate *semantic* features. Furthermore, given the fact that personal life events can impact a user's feelings and emotions such as becoming glad, upset, and restless, among others, the use of sentiment

features have also been considered (Choudhury and Alani 2014b) and shown to be strong indicative features for personal life event detection. From a different perspective, attention features (Choudhury and Alani 2014a) define how the content posted by a user are noticed by other users and are measured in terms of replies and retweets, and reflect how many times the user is addressed/talked about by other users within a given time interval. The motivation for using these 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 (Dickinson et al. 2015; Choudhury and Alani 2014a, b). Dickinson et al. (2015) refer to attention features as *interaction* features and rather than just considering the number of retweets, favorites or replies, they consider who are the users performing these actions and their interaction patterns towards the poster of the tweet. They found almost no effect for the interaction features for the purpose of life event detection when applied on Twitter content and compared to other features such as n-grams, sentiments, and emojis.

Finally, while many of the introduced features have good performance on the precision metric, they do not perform too well with respect to *recall*. To address this issue, the work in (Khodabakhsh et al. 2017) adopts a word vector representation of tweets that is obtained from neural word embedding techniques. These neural word embedding based features incorporate both syntactic and semantic aspects of the content and therefore show to be effective features for identifying personal life events from the perspective of both recall and precision.

2.3 Deep learning on social content

Unlike work on life event detection and event prediction on Twitter, our work in this paper focuses on predicting future life events prior to them being reported by the user. We employ recurrent neural networks (RNN) for this purpose.

In tasks that involve sequential inputs, such as speech and language, recurrent neural networks process each element of the input sequence in the order observed in the input and codify the observed past information in the sequence as a vector in the hidden states, which can be considered to be a form of a feature. RNNs are in essence a folded feed-forward network. RNNs are suitable for predicting the next characters in text Sutskever et al. (2011), the next word in a sequence (Mikolov et al. 2013b) or more complex tasks such as translation (Sutskever et al. 2014; Cho et al. 2014). After reading a sentence word by word, an encoder network is used to learn the thought expressed in that sentence in its final hidden state, referred to as the *thought vector*. The final hidden state is then used as the initial hidden state of a jointly trained decoder that would produce a probability distribution for the first word in the output sequence. Once a particular first word is chosen from this distribution and provided as input to the decoder, it will then output a probability distribution for the second word of the output sequence and so on until a termination symbol is observed.

In theory, RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps. Long Short-Term Memory (LSTM) was introduced to overcome this problem by augmenting the network with a memory cell which remembers inputs for a long time (Hochreiter and Schmidhuber 1997). LSTM networks or related forms of gated units are also currently used for encoder and decoder networks that perform well in machine translation (Sutskever et al. 2014; Cho et al. 2014), sentiment analysis (Tang et al. 2015; Glorot et al. 2011), text classification (Zhang et al. 2015; Lee and Dernoncourt 2016; Kim 2014), generation of sequence models (Graves 2012a).

Beyond RNNs, other deep learning methods have also shown promising results on various tasks on Twitter content. Tang et al. (2014) extended the C&W model and developed three neural networks to learn sentiment-specific word embeddings from Twitter content. The tweets containing positive and negative emotions were used as training data. Severyn and Moschitti (2015) used a three-step process to train their deep learning model for sentiment analysis of tweets at both message and phrase levels. Word embeddings were initialized using a neural language model which is trained on a large unsupervised collection of tweets, and a convolutional neural network is used to further refine the embeddings on a large distant supervised corpus. Limsopatham and Collier (2016) presented a model based on bidirectional LSTMs to automatically learn orthographic features without requiring feature engineering for named entity recognition in Twitter messages. Their model consists of three main components: (1) orthographic sentence generator, (2) word representations as input vectors, and (3) a bidirectional LSTM. Li et al. (2016) proposed an attention-based LSTM model, which incorporates topic modeling into the LSTM architecture through an attention mechanism in order to recommend hashtags for tweets.

A deep learning framework for rumor debunking on Twitter and Weibo is introduced in Ma et al. (2016). The framework learns RNN (tanh, GRU, LSTM) models by utilizing the variation of aggregated information across different time intervals related to each event. Lin et al. (2014) believed that individuals' psychological stress is recognizable via social media. They defined two sets of attributes: 1) low-level content attributes from a single tweet, including text, images and social interactions, and 2) user-scope statistical attributes such as behavioral attributes, social engagement and linguistic style attributes from users' weekly tweet postings. To combine content attributes with statistical attributes, they designed a convolutional neural network (CNN) with cross autoencoders to generate user-scope content attributes from low-level content attributes. Finally, they proposed a deep neural network (DNN) model to incorporate the two types of user-scope attributes to detect users' psychological stress.

3 Approach overview

In our work, we propose three variations of recurrent encoder-decoders for predicting future personal life events based on user's past tweets. We employ Sequence-to-Sequence (Seq2Seq) models to perform *(i)* direct life event label prediction; and, *(ii)* indirect life event prediction through the generation of future tweets. In the first approach, we predict future life event labels directly based on a *many-to-one* Seq2Seq architecture. The output label of the sequence to sequence model is the future life event that the user might report on her Twitter timeline. We present two variations of the many-to-one Seq2Seq architecture. In the first variation, a set of tweets are directly fed into the encoding stage and predicted life event label is produced in the decoding stage. The second variation; however, works with a collection of past life event labels from that user to predict the future life event. Therefore, the input to the decoder stage is a sequence of past personal life events reported by the user and the output is a predicted personal life event in the future. Now, given the second variation requires a set of input life events, we generate such labels by using state of the art methods that classify a given tweet into certain life event classes. As such, a set of past tweets of the user are passed through such classifier that would produce one life event label per tweet. These life event labels are then passed onto the many-to-one Seq2Seq model as input to the encoder stage.

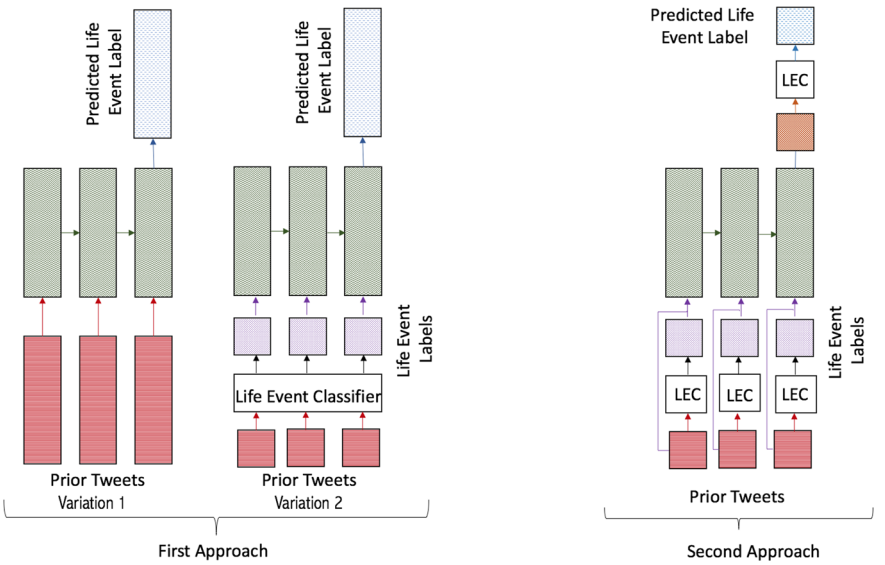

Fig. 1 Overview of the proposed Seq2Seq models for life event prediction (LEC stands for Life Event Classifier)

In the second approach, we employ a *many-to-many* Seq2Seq model that operates over the combination of the inputs of the two earlier variations of the first approach, namely the tweets and the past personal life events, in the encoding stage. The model then generates a word sequence in the decoding stage that estimates the content of a future tweet. The idea of the second approach is that it might be possible to predict a user's future tweet based on the personal life events they have reported in the past as well as the tweets they have posted. We generate the user's future tweet through a many-to-many recurrent model, the output of which is then classified into one of the personal life events using state of the art life event classification methods. Figure 1 shows the graphical representation of the two proposed approaches for life event prediction.

4 Proposed approach

We provide the details of our two proposed approaches for life event prediction based on a sequence-to-sequence architecture. The Seq2Seq architecture is trained for classification in the first approach and for sequence generation in the second.

4.1 Many-to-one classification architecture

In the first approach, a Seq2Seq architecture is used to directly generate a personal life event label for a user given her past shared content. The most intuitive approach for predicting future behavior based on social content is to take the tweets that the user has posted into account to predict what the user will post in the future. For instance, a person who is getting married is likely to post tweets talking about preparing for the wedding day or a person changing jobs might post about how she is looking for a new job in the days or months

leading to her finding the job. Therefore, our main hypothesis is that a user's past tweets can be seen as potential indicators for a future life event.

In the *first variation* of the many-to-one Seq2Seq architecture, a sequence of tweets from the timeline of a user is taken into consideration to predict the future personal life event that the user will mention in her timeline. To this end, we need to work with an input sequence consisting of l tweets of user u, denoted as $tw_u = [tw_u^1, tw_u^2, ..., tw_u^l]$. In order for the sequence tw_u to be used as input, a representation form for each tweet needs to be adopted. One representation model for a tweet is to represent it as a bag of words, where each tweet is represented as a vector with v dimensions where v is the number of unique words in the tweet corpus. This method suffers from the curse of dimensionality, which is aggravated within the context of Twitter given the wide range of slangs, acronyms and abbreviations. As an alternative, Vosoughi et al. (2016) proposed a model for generating general-purpose vector representation of tweets. The model learns tweet embeddings using a character-level CNN-LSTM encoder-decoder. Dhingra et al. (2016) also adopted a similar strategy and proposed a Bi-directional Gated Recurrent Unit (GRU) neural network for finding vector space representations of tweets by learning complex, non-local dependencies in character sequences. We adopt a similar strategy to Dhingra et al. (2016) and compute a vector representation for each tweet to be used as input. Based on the vector representations of the past tweets and an observed personal life event reported by the user at time interval l, denoted by φ_u^l, the objective of the model is to maximize the selection probability of φ_u^l given tw_u:

$$\arg\max_u P(\varphi_u^l | tw_u^1, tw_u^2, ..., tw_u^{l-1}) \tag{1}$$

4.1.1 Encoding

In the proposed architecture, the encoder converts the input tweet vectors into a set of high dimensional hidden representations $h = [h_1, ...h_{l-1}]$ where h_i is the hidden state of the Seq2Seq architecture at time i. In other words, at time i, the encoder reads tw_u^i and updates its hidden state h_i as follows:

$$h_i = RNN(h_{i-1}, tw_u^i) \tag{2}$$

where $h_0 = 0$ and the RNN applied in the encoder is often either a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) or Gated Recurrent Units (GRU) (Cho et al. 2014), which are both capable of learning long-term dependencies and do not suffer from the vanishing gradient problem. Similar to our proposed interval encoding mechanism shown in (2), there have been other work that employ comparable approaches. For instance, Tang et al. (2015) used an LSTM to produce sentence vectors, which are then applied within the context of a bi-directional gated recurrent neural network to compose the sentence vectors into document vectors. Likewise, Yang et al. (2016) proposed a hierarchical attention network for document classification, which employs GRU for encoding both words and sentences. Similarly in our encoders, we use GRUs primarily because they have similar performance to LSTMs but are more computationally efficient.

4.1.2 Decoding

In our architecture, the decoder is a single-layer GRU that is responsible for generating the output predicted life event based on the encoded input in the form of a vector in the embedding space. Given the output sequence only consists of one token in our first approach, i.e.,

the predicted personal life event, the decoder would predict φ_u^l based on h_{l-1}:

$$P(\varphi_u^l = le|tw_u^1, tw_u^2, ..., tw_u^{l-1}) = P(\varphi_u^l = le|h_{l-1}) = softmax(d^1, \varphi_u^l = le) \quad (3)$$

where le is some life event such as changing jobs or getting married and $d^1 \in R^d$ is the hidden state of the GRU decoder:

$$d^1 = GRU(d^0, \varphi_u^1) \quad (4)$$

It should be noted that d^0 is initialized based on the hidden state of the interval encoding h_{l-1} based on (2).

4.1.3 Model training

In order to train the model, we maximize (1) whose parameters are learned by maximizing the log-likelihood of the condition. Back propagation through time (BPTT) (Werbos 1990) is used for computing the gradient of the objective function. The overall scheme for the first approach is depicted in Fig. 2, where the encoder computes the hidden representation for all φ_u^is, where $l > i \geq 1$ through GRU. This hidden representation is then used to compute the probability of φ_u^l.

The *second variation* of the same many-to-one Seq2Seq architecture, as shown in Fig. 2b, is based on a more dense input representation where the input to the encoder is the sequence of personal life events that have been reported by the user in the past. As discussed in the related works, there are already work in the literature that can classify a tweet into one of several classes of life events depending on the content of that tweet. For instance, such methods are able to label a tweet such as `'Just booked my flight to Cali, a vacation sounds nice'` as one that is related to *travel* and another tweet such as `'So excited to start my new job i've practically bought an entirely new wardrobe'` as a tweet that reports on the *job change* life event. Based on such personal life event classifiers, we are able to produce a sequence of life

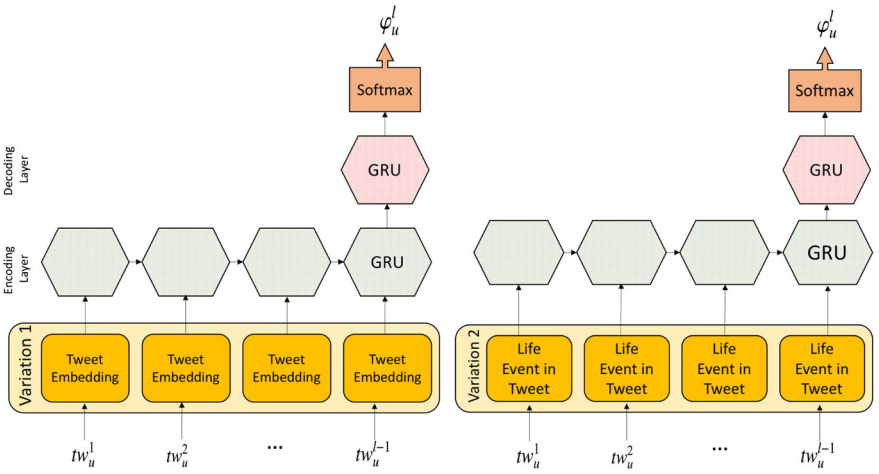

Fig. 2 The architecture for the many-to-one Seq2Seq model for both variations: **a** based on past posted tweets, and **b** based on past observed life events

events, including the `none` event, which refers to cases when no life events are mentioned in the tweet, for each user. In our work, we adopt the method proposed in Khodabakhsh et al. (2017) for detecting personal life events. This method proposes to collect a set of words to *discriminatively* express each life event. The word vector representation of these words are then computed using word embedding methods, i.e., the Skipgram model (Mikolov et al. 2013a). Based on the vector representation of the life event words, it is possible to calculate the distance between each life event and a given tweet, also represented as the vector of its constituent words using the *word mover's distance* measure (Kusner et al. 2015). Now, the advantage of this model is that it not only identifies the life event mentioned in a tweet, but also provides a vector representation for each of the life events based on the centroid of its constituting discriminative words. As such, the corresponding vector representation to each life event is used as input to the encoder in the Seq2Seq architecture when that life event has been detected in the previous tweets. In summary, we label a tweet with a life event based on the closeness of its vector representation to the vector representation of the life events and chose the vector representation of the selected life event as the representation of the tweet to be input into the encoder of the Seq2Seq model. The rest of the details of the architecture are similar to the first variation.

4.2 Many-to-many generation architecture

The main idea of the second approach is to predict a future personal life event by trying to predict the content of a user's future tweet. Similar to Sordoni et al. (2015), we hypothesize that there are two information pieces that can assist with the prediction of a future tweet, namely the past tweets and the user's past tweet topics. In the context of our work and given we are interested in personal life events, we customize these two information pieces as (i) user's past tweets, and (ii) user's past reported life events; therefore, we exploit users' tweets and the life events that they reported as prior knowledge for estimating the next tweet. In order to incorporate these information pieces into our model, we need to first create a uniform representation as the two information sources are not compatible because past tweets are a set of words while the reported life events are merely a label. We address this issue by converting the reported life event labels into a set of representative words for the life event. The representative words used here are those derived based on Khodabakhsh et al. (2017) and used in the second variation of the first approach. Based on these words, depending on the life event that a tweet is reporting, it is augmented with a set of words representing that life event. It should be noted that the reported life event for each tweet is automatically determined and those tweets that are determined not to be reporting any personal life events are not augmented with any additional words. Now, based on the vector representation of the augmented tweets as well as an augmented future tweet, i.e. the sequence of words in a future tweet plus the representative words of the life event that tweet is representing, the objective is to maximize the generation probability of the augmented future tweet as follows:

$$\arg\max_{u} P\left(aug\left(tw_u^l, \varphi_u^l\right) | aug\left(tw_u^1, \varphi_u^1\right), aug\left(tw_u^2, \varphi_u^2\right), ..., aug\left(tw_u^{l-1}, \varphi_u^{l-1}\right)\right) \quad (5)$$

where tw_u^l is the user's l^{th} tweet, and φ_u^l is the reported life event in that tweet. Also, $aug()$ is a function that augments the words within the input tweet with the relevant representative life event words. This function returns a vector whose elements are the tweet words and life event words sequentially. In other words, the words in the tweet are concatenated with the representative words of the life event, forming one sequence of words consisting of the tweet words first followed by life event words. The overall architecture of the Seq2Seq

model for the second approach is shown in Fig. 3. This many-to-many Seq2Seq architecture predicts the next tweet given the tweets already posted by the user. The history of past posted tweets is considered as a sequence at two levels: a sequence of words for each tweet and a sequence of tweets. The many-to-many Seq2Seq architecture models this hierarchy of sequences using two RNNs: one at the word level and one at the tweet level. In the sequence of tweets, the first encoder RNN maps each tweet to a tweet vector. The tweet vector is the hidden state obtained after the last token of the tweet has been processed. The higher-level encoder RNN keeps track of past tweets by processing each tweet vector iteratively. After processing a tweet, the hidden state of the second encoder RNN represents a summary of the sequences up to and including the tweet, which is used to predict the next tweet. The next tweet prediction is performed by means of a decoder RNN, which takes the hidden state of the second encoder RNN and produces a probability distribution over the tokens in the next tweet. The decoder RNN is similar to an RNN language model but with the important difference that the prediction is conditioned on the hidden state of the second encoder RNN. It can be seen as a tweet generation module.

4.2.1 Encoding

The encoding process of our architecture consists of two encoding processes: (1) encoding of the augmented tweets, and (2) the encoding of the sequence of augmented tweets. In the first encoding step, we employ a recurrent neural network architecture for encoding the augmented tweets, which produces a fixed-length vector, after reading the last word of an augmented tweet. The RNN processes the tweet words and the life event words sequentially

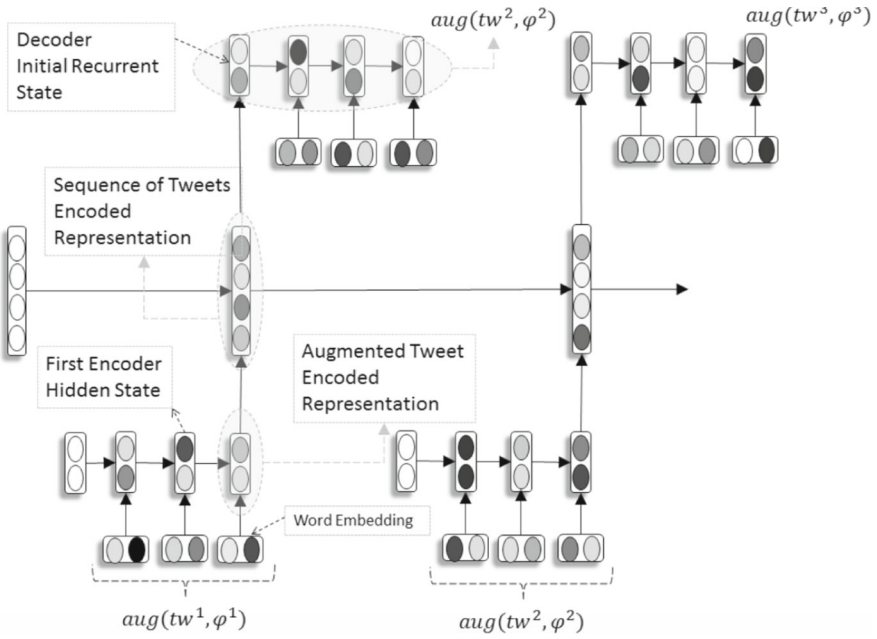

Fig. 3 The architecture for the many-to-many Seq2Seq model for augmented tweet prediction (the different color circles point to the fact that vector elements can have different values)

from the augmented tweet and updates its hidden state as follows:

$$h_i^1 = RNN_E\left(h_{i-1}^1, aug\left(tw_u^k, \varphi_u^k\right)[i]\right) \tag{6}$$

The hidden state $\left(h_i^1\right)$ is expressed with a superscript to denote that it is related to the first encoding step and $aug(tw_u^k, \varphi_u^k)[i]$ is the i^{th} word of the augmented tweet tw_u^k. It is important to mention that our work has resemblance to the work by Semeniuta et al. (2017), which is based on the the variational autoencoder framework for generating tweets. Its scoring components are a convolutional encoder and a deconvolutional decoder combined with an LSTM recurrent layer. Similarly, Oak et al. (2016) also employed LSTMs at both character and word levels to generate realistic-looking tweets with the same statistical properties as the actual data. Furthermore, Shang et al. (2015) have introduced a Neural Responding Machine (NRM), based on the general encoder-decoder framework, in order to generate response for Weibo (a popular Twitter-like microblogging service in China) conversations. They used GRU for both the encoder and decoder. Also, Serban et al. (2017) proposed the multi-resolution recurrent neural network for generatively modeling sequential data at multiple levels of abstraction. They used GRUs and applied it to dialog response generation for Twitter conversations. One of the shortcomings of GRUs is that they are only able to make use of previous context and have restrictions as the future input information cannot be reached from the current state. Bidirectional RNNs (BRNNs) (Schuster and Paliwal 1997) address this issue by processing data in both directions with two separate hidden layers, which are stacked on top of each other. In our work, we use Bidirectional GRUs, to represent RNN_E, which compute a *forward* hidden layer by iterating through the input from beginning to end and a *backward* hidden layer by iterating through the input sequence from the end to the beginning. These two hidden layers are concatenated into a single vector, denoted by h_i^1. Therefore, h_i^1 represents the encoded version of the augmented tweet up to word i.

The second encoding step is focused on reading the encoded version of the augmented tweets sequentially and encoding this sequence into a fixed-length vector. The RNN processes the encoded representation of each augmented tweet and updates the encoding as follows:

$$h_i^2 = RNN_S\left(h_{i-1}^2, h_i^1\right) \tag{7}$$

Likewise, RNN_S is modeled as a GRU.

4.2.2 Decoding

In this architecture, the decoder is another GRU, which is trained to generate the next augmented tweet given the previous augmented tweets. In other words, the GRU decoder processes the hidden state of the encoded augmented tweets as input and produces a probability distribution over the tokens in the next time interval as output and is computed as follows:

$$P(aug(tw_u^l, \varphi_u^l) | \underbrace{aug(tw_u^1, \varphi_u^1), ..., aug(tw_u^{l-1}, \varphi_u^{l-1})}_{\Psi}) = \prod_{i=1}^{N} P(w_i | w_1, .., w_{i-1}, \Psi) \tag{8}$$

where is N is the length of tw_u^l and w_i is its ith word. The probability of word w_i given previous words in the tweet and the previous augmented tweets can be estimated as follows:

$$\prod_{i=1}^{N} P(w_i = v|w_1, .., w_{i-1}, \Psi) = softmax(d^{l-1}, w_i = O_v) \tag{9}$$

where O_v is the encoded embedding of word v and $d^{l-1} \in R^d$ is the hidden state of the RNN decoder:

$$d^j = RNN_D(d^{j-1}, w_j) \tag{10}$$

Here, RNN_D is a GRU. It should be noted that we use beam search (Graves 2012b) for in the decoding process to approximate the most probable augmented tweet given the past augmented tweets.

4.2.3 Model training

Let us assume that a time interval I consists of l augmented tweets and that a generative model parameterizes a probability distribution P with parameters θ over the set of possible intervals of arbitrary length. The probability of an interval I can be defined as follows:

$$\begin{aligned} P_\theta(aug(tw_u^1, \varphi_u^1), ..., aug(tw_u^l, \varphi_u^l)) &= \prod_{t=1}^{l} P_\theta(aug(tw_u^t, \varphi_u^t)|\underbrace{aug(tw_u^1, \varphi_u^1), .., aug(tw_u^{t-1}, \varphi_u^{t-1})}_{\Upsilon}) \\ &= \prod_{t=1}^{l}\prod_{i=1}^{N_t} P(w_i^t|w_1^t, .., w_{i-1}^t, \Upsilon) \end{aligned} \tag{11}$$

where N_t is the length of the tth augmented tweet and w_i^t is its ith word.

The parameters θ can be learned by maximizing the log-likelihood of an interval I. We employ back propagation through time (BPTT) (Werbos 1990) for computing the gradient of the objective function as defined in (11). Based this equation, it is possible to generate a future augmented tweet based on its relevance to the past augmented tweets. In our work, the recurrent neural networks RNN_E, RNN_S and RNN_D are GRUs whose parameters are learnt in this way. In light of the fact that the sequence of words produced by the decoder in the output is intended to represent a user's future tweet, we employ it to predict the next life event of the user. We use the same method presented in Khodabakhsh et al. (2017), denoted as Life Event Classifier (LEC) in Fig. 1, to classify the generated sequence into one of the life events. The determined life event class for the generated tweet would represent the predicted life event in this model.

5 Experiments

We evaluate the different variations of our sequence to sequence model, and compare it against the state of the art baseline methods. The objective of our experiments is to measure the effectiveness of our proposed work for predicting future personal life events solely based on the previous tweets of the user.

Table 1 Specification of the dataset with the distribution of life events

Life event	Number of tweets	Percentage
Broken device	5,768	8.69%
Device upgrade	2,822	4.25%
Moving	4,001	6.03%
New job	4,001	6.03%
Travel	3,480	5.24%
Wedding	4,015	6.05%
Negative samples (no life event)	42,275	63.7%

5.1 Experimental setup

The primary dataset used in our experiments is composed of 66,362 labeled tweets collected and annotated manually by twenty experts.[1] The labeled tweets cover six personal life events as shown in Table 1. The dataset also consists of tweets that are not about any life events, which are used as negative samples. As shown in Table 1, 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 dataset, over 63% of the tweets are related to no life events.

For each of the labeled tweets in the dataset, we also extracted tweets by the same user from one week prior to the given tweet through the Twitter RESTful API in order to have access to historical tweets. We pre-processed the dataset as follows. First, we removed URLs and reposting marks such as 'RT' and '//'. Then, we filtered stop words. Finally, we used the Stanford CoreNLP tool set for lemmatizing all the tweets.

In all models, back propagation through time Werbos (1990) and Adam (Kingma and Ba 2015) are used for computing gradient of the objective function and optimization, respectively. All our models were trained on NVIDIA Tesla K80 GPU provided through SHARCNET.[2] Our implementation is done using the open-source Python library Theano (Bastien et al. 2012). The hidden state dimension for the encoders and decoders is set to 512 in both variations of the first approach. Also the size of the embeddings is set to 100. The training stops if the likelihood of the validation set does not improve for 10 consecutive iterations (early stop). The learning rate was set to 0.001. The dimension of hidden state of the RNN in the second approach is set to 1,000. We initialize the word vectors using a pre-trained word embedding with size 400 obtained by executing the Skip-gram methods (Mikolov et al. 2013a) on a 10 million English tweet corpus collected through the Twitter RESTful API. This model is trained with a learning rate of 0.0001 and with mini-batch containing 25 training samples. The size of beam is 1. In the test phase, we consider variant lengths of 4, 6 and 8 because the average number of words in a tweet is close to 6 based on an experiment that we did on our tweet corpus.

5.2 Baselines

While there are no prior work on predicting future personal life events, as mentioned in the related work, there are work that determine whether a life event is mentioned in a tweet

[1] Available upon request

[2] https://www.sharcnet.ca/

or a collection of tweets. One of the most related works in the literature is proposed by Li et al. (2014), which is a pipelined system for detecting major life events. This work first identifies the life event category the tweet is referring to and subsequently identifies whether the tweets is a self report of the life event or it is referring to the life event for a third party. As suggested by the authors, in order to replicate the baseline, we extract features including bag of words, named entity mentions, dictionary and window for tweet modeling. Bag of words and NER features are the sequence of words in the tweet and named entities, respectively. A dictionary of the top-40 words for each life event category is built (automatically inferred by a topic model). The dictionary feature value is the term's probability generated by the corresponding event. If a dictionary term exists in the tweet, left and right context words within a window of 3 words and their part-of-speech tags are extracted as the window feature. Named entity tags are assigned using Ritter et al.'s Twitter NER system (Ritter et al. 2011) and Part-of-Speech tags are assigned based on Twitter POS package (Owoputi et al. 2013). As suggested by the authors (Li et al. 2014), we train a 7-class maximum entropy classifier with all these features as the experiments by Li et al. reported this classifier to have the best performance. The other work that we use as baseline is the life event detection method proposed by Choudhury and Alani (2014b). In their work, the authors propose to extract a set of features including hashtags, emoticons, named entities, and sentiments to build classifiers that can find life event mentions. We use these features as well to build a second baseline using a maximum entropy model.

Given the objective of our evaluation is to investigate the performance of the proposed method for life event prediction, we use the standard information retrieval metrics including *precision*, how many of the life events detected by our method were correct, *recall*, how many life events in the ground truth set were retrieved by our method and F-Score, which is the harmonic mean of precision and recall.

5.3 Results

In order to perform the evaluations, we split the life event dataset introduced in Table 1 into three parts, namely training, validation and testing, each of which contained 80%, 10%, and 10% of the original dataset, respectively. The results of the experiments are shown in Table 2 for both variations of the first approach (classification architecture) as well as the second approach (generation architecture). The results are compared against the strongest baselines available in the literature reported by Li et al. (2014) and Choudhury and Alani (2014b). Given our objective is to predict the personal life event of a future time interval, we executed Li et al.'s method using both the last tweet prior to the future tweet as well as the past one week of tweets leading to the future tweet of interest. However, given the nature of the features by Choudhury et al., we report the findings based on building classifiers for each individual feature as well as a classifier where all features are included using the Maximum Entropy classifier. The results are reported in Table 2. As seen in the table, the precision of both variations of the baseline by Li et al is quite poor and in fact close to random when keeping in mind that we are performing a 7-class prediction task and therefore a random classifier would perform at a 14% precision rate compared to the current precision of the baseline which is 13.3%. It should be noted that the work by Li et al is considered to be a strong baseline as it performs at precision rates of $\sim$ 75% when the life event is mentioned in the tweet that is being classified; however, as observed in our results, it is not able to show competitive performance for performing future life event prediction. Among other conclusions, this significant difference in classification precision shows that

Table 2 The performance of the proposed classification and generation methods

Method		Precision	Recall	F-Score
Classification architecture	Variation 1	0.234	0.211	0.222
	Variation 2	0.272	0.328	0.297
	Sequence Length			
Generation architecture	100	0.289	0.363	0.322
	8	0.357	0.358	0.358
	6	0.377	0.359	0.368
	4	0.394	0.369	0.381
Baseline by Li et al. (2014)	Latest Tweet	0.13	0.361	0.191
	All Tweets from Past Week	0.133	0.364	0.195
Baseline by Choudhury and Alani (2014b)				
based on maximum entropy classification	Hashtags	0.129	0.359	0.19
	Emoticons	0.13	0.36	0.191
	Named Entities	0.13	0.36	0.191
	Sentiments	0.133	0.365	0.195
	All Features	0.19	0.358	0.248

the task of predicting the personal life event of a future task is quite difficult and cannot be performed using existing life event classification techniques. Furthermore, we observe that the work by Choudhury and Alani (2014b) shows quite similar performance to the work by Li et al. (2014) when individual features are used to train the classifier. The performance of these classifiers are weaker than an average baseline, which would have a precision of 14%. However, the performance of the classifier based on all of the features is better than the other baselines in terms of precision at 0.19. However, this performance is still weaker than all variations of our generation architecture and weaker than the second variation of the classification architecture.

Now, in terms of the comparative performance of the two variations of the first approach, which is primarily based on a many-to-one Seq2Seq classification architecture, one can see that the second variation shows stronger performance. While the first variation works directly with the tweets, the second variation only works based on a set of life event labels that have been observed in the past. In other words, the input sequence to the second variation is a sequence of personal life events that the user has tweeted about (including a *none* label). Our observation has been that reducing symbol space of the Seq2Seq model has improved the performance of the model in terms of both precision and recall. The main reason for this could be that the first variation needs to model any sequence, which could consist of many different tokens spanning possibly beyond the number of English words, as tweets could include abbreviations and slang words, and therefore, creating an accurate representation for such a large symbol space would be challenging. In contrast the input sequence space of the second variation consists of only seven symbols each representing one of the life events or the *none* life event label.

While the second variation shows better performance compared to the first variation, it is important to point out that it is limited to the performance of the underlying tweet life event classification algorithm. In other words, given the input to this model is a sequence

of life event labels, and tweets themselves do not have the life event labels with them and hence the life event label for each tweet is automatically determined using a classification technique, the performance of the second variation is dependent on the performance of the employed life event classifier. Therefore, it is possible to get better performance using the second variation if more accurate life event classifications per tweet were available.

In the second approach, which unlike the first approach, is based on a sequence generation architecture, an augmented tweet is predicted, which is assumed to be the rough estimated representation of a future personal life event-related tweet by the user. As seen in Table 2, we have tested the second approach on varying lengths of the output sequence. The output sequences that were generated had a length of 4, 6, 8, and 100 tokens. We were interested to see whether both smaller and larger sequence lengths had any significant impact on the performance of the prediction model. The first important observation is that the predictions made based on the second architecture, regardless of the length of the output sequence, have both higher precision and recall rates compared to the classification architecture. This can be attributed to the fact that the generation architecture employs an augmented tweet as input, which consists of the tweet itself and a specific set of words for the related life event. This way, the set of words related to the life event drives the encoding and decoding process and hence leads to better prediction performance. It should be noted that more so than the second variation of the classification approach, the performance of the generation architecture is even more dependent on the performance of the employed life event classifier as it is used for both generating the augmented tweets as well as classifying the generated output sequence.

From the perspective of the generated sequence length, it can be observed that while it does not significantly impact the performance of the recall metric, it does have a substantial impact on precision. Longer generated sequences have a lower precision and hence less effective for predicting a future personal life event. This can be potentially explained at least by two reasons. First, a study in 2012[3] showed that while tweets can be as long as 140 characters, in practice they are 40 characters long on average. This is approximately 4-6 words in length per tweet. This is also in line with our own observation in our Twitter gold standard dataset. Therefore, generating output sequences of the same length as the average tweet would prevent the generation of possibly irrelevant tokens in the output. Second, given the output sequence generated at the decoder relies on *beam* search, the longer the sequence is, the more likely it will be that the generated sequence would have lower cohesion (Graves 2012b). We observed that the output sequence of length four provides the best performance in terms of both precision and recall.

5.4 Bootstrap aggregating

Several researchers have already shown that ensemble meta-algorithms can improve the performance of recurrent neural networks. More specifically, the bootstrap aggregating (*bagging*) method can lead to improved classification performance without the need to change the model being trained and by only learning ensembles over different subsets of the training data. Bagging operates by generating n training subsets by uniformly sampling data points with replacement from the original training set, which would then be used for

[3] https://goo.gl/ohBcD8

training *n* similar models trained on different data subsets. Breiman (1996) has shown that bagging can improve the performance of *unstable* methods such as neural networks that have been used in this paper. On this basis, we applied the bagging approach on both the baselines and the various proposed architectures in this paper.

To apply the bagging approach, we trained several models with different number of epoches ranging from 200 to 400 with an interval of 50 epoches in between while also systematically applying the training dataset splits. Given the results in Table 2 showed that the generation architecture shows its best performance with a sequence length of 4, we used this sequence length in the models trained in the bagging approach, as well. The multiple predicted labels were gathered and used to get an aggregated predicted label based on a *plurality vote*. It should be noted that we experimented with both weighted and unweighted bagging methods and the difference was not noticeable to the third decimal point. The results of the bagging approach are reported in Table 3.

There are two main observations from the obtained results. The first is that all models have been improved as a result of the bagging process on both precision and f-score measures. This is most noticeable on the generation architecture where the precision improved as a result of bagging from 0.394 to 0.519, which is an improvement of 31.7%. The second observation is that the bagging process does not have a significant impact on recall as the improvements are only marginal. This observation can be broadened by comparing the recall rates across the different methods as well as before and after the bagging process. In all cases, while the recall rate of the generation model both after and before bagging outperforms the other methods but still the differences are not substantial. This shows that the models have very similar *retrieval* capacity with recall values in the range of 0.33 to 0.373, while the generation architecture has a strong *classification* ability shown by the precision of 0.519.

It should be noted that the results of the life event prediction should be understood within the context of the work on personal life prediction and the extremely difficult, highly noisy and class imbalanced dataset that it offers. We would like to point out that the strongest

Table 3 The performance of the proposed classification and generation methods after bagging

Method		Precision	Recall	F-Score
Classification architecture	Variation 1	0.262	0.330	0.292
	Variation 2	0.278	0.338	0.305
	Sequence Length			
Generation architecture	4	0.519	0.373	0.434
Baseline by Li et al. (2014)	Latest Tweet	0.247	0.264	0.255
	All Tweets from Past Week	0.278	0.305	0.291
Baseline by Choudhury and Alani (2014b)				
based on maximum entropy classification	Hashtags	0.129	0.359	0.19
	Emoticons	0.129	0.359	0.191
	Named Entities	0.129	0.359	0.191
	Sentiments	0.223	0.256	0.195
	All Features	0.19	0.358	0.248

baseline proposed by Li et al. (2014) offers an f-score of 0.54 when working directly on the tweet of interest and performing a tweet classification task. In contrast, we attempt to predict the future life event without having access or working with the content of the tweet of interest and solely rely on past tweets to predict the future life event and achieve an f-score of 0.434, which is meaningful and strong when compared to the results obtained in Li et al. (2014).

In the next section, we will present tweets from our dataset for both correctly labeled and incorrectly labeled instances and discuss the predominant reasons for the incorrect classifications.

6 Error analysis and discussion

Beyond the information retrieval metrics that report the performance of the proposed methods, it is also important to understand the cases when the models succeed in identifying the correct personal life event as well as when and why the models do not correctly identify the life events. To this end, we have reviewed the result of the better performing generation architecture. We found that the model performs very well for cases when a logical progression between the tweets of the same user can be observed. In other words, for those twitter users who consistently and continuously discuss their personal life events on Twitter, the model is able to show good performance. This is primarily due to the fact that the Seq2Seq model builds an internal representation by encoding a sequence of tweets and their life event labels (*augmented tweets*), which is then used to predict a future tweet. The more cohesive the set of tweets in this sequence are, the more consistent the encoded internal representation would be and hence the generated sequence through decoding is more likely to represent the future tweet and its life event. However, if the past sequence of tweets has arbitrary order, then the internal representation would be convoluted based on the inputs from the unrelated tweets in the sequence.

An example of a case where the model is successful in predicting the personal life event is when a user reports *'Smashed my iPhone, okay fairs'* and subsequently posts *'first one I've ever smashed!!! Dan do apple fix your phone or give you a replacement??'*. For such a consistent sequence of tweets, the model generates the sequence: `'note drop phone break'`, which is easily labeled as the 'Broken Device' label. The first two rows of Table 4 provides two more examples of how coherent flows of tweets lead to correct classification.

Now in terms of the incorrectly predicted personal life event labels, we identified three primary reasons for the incorrect predictions. The first and primary cause (Error Type I) for the errors was related to the cases when the user was reporting a life event related to another person. In such cases, it was possible that the reported life event would be determined to be a personal life event and hence labeled incorrectly. For instance, for a user who tweeted *'OMG in shock with the news I just received. You never know when your time is up'* and subsequently *'Mom going house hunting in Florida'*, the method incorrectly predicted the 'Move' life event having mistakenly labeled the second tweet as a *move* related tweet and hence using the incorrect life event in the augmented tweet and hence negatively influencing the sequence generation process. It should be noted that the performance of the Life Event Classifier (Khodabakhsh et al. 2017) used in our experiments leads to such errors. While this model does implicitly consider self-reports in the process, it does not explicitly do so from a linguistics perspective. There are other techniques in the literature (Li et al. 2014) that have a pipeline that determines whether the user who is reporting the life event is in fact directly involved in the life event or not, e.g., through checking whether the subject of

Table 4 Sample tweets and the predicted labels

	Past tweets		Actual label	Predicted label
Correct predictions	Smashed my iPhone, okay fairs	First one I've ever smashed!!! Dan do apple fix your phone or give you a replacement??	Broken Phone	Broken Phone
	you'll be back! Hopefully!,Xx	This house hunting is getting to me now meh!	Move	Move
Error type I	Got to bring this beautiful girl to my sister's wedding last night [url]	Just one more big shoutout to Kevin and Debby finally getting married this weekend! Glad to join	No Life Event	Wedding
	My Friend From Bible School Is Pregnant	She Got Married 6 Months Ago and Now Shes Preggers	No Life Event	Wedding
Error type II	OMG my grandma lol [url] g grandma lol	I've always been the person to think getting married young was stupid but oh god I wouldn't mind marrying you right now	No Life Event	Wedding
	@user You can find rent, floor plans and all other info on the property here: [url]	@user You should move here, Raleigh is a great place! Will you be looking for an apartment if you move?	No Life Event	Move
Error type III	@user @user No appointments at passport office so I have to pay 200 for an external companies help	@user @user I'm not due a smashed phone for another 2 weeks.	Broken Phone	Travel
	Exploring the trails and the falls. #WebstersFalls #DiscoverOntario #ExploreCanada #hamont [url]	Finally working on my i-to-i TEFL eResume! Looking forward to starting the job search and beginning a new adventure, and chapter in my life!	New Job	Travel

the tweet is a *first person singular*, but given those models do not have 100% precision, still such instances can happen.

The second reason for the errors (Error Type II) was due to cases when a user was providing information on a topic that had similar components to a life event. For instance, for a user who tweeted *'If you was a Passport where would you hide?'* and then later tweets *'I lost my passport, got a new one. Lost new one, found old one. Can I use it?? '*, the user is in fact communicating some information about his passport to his followers. However, given the tweets have similar components to a travel tweet, the model would predict a 'Travel' life event for this user. The third reason for the errors (Error Type III) in the prediction was related to a sudden change of subject in the users tweets. In these cases, the user is discussing some topic and suddenly changes the topic and discusses a new topic. For such cases, the proposed sequence generation approach cannot build a coherent internal encoded representation and hence has difficulty in predicting the correct life event. For instance, a user has tweeted *'@user love ya mucho'* and then immediately afterwards posts *'unpacking from one trip and packing for another'*. In this case, the proposed approach having encoded the word *love* mistakenly predicts the 'Wedding' life event. As such Error Type III is essentially related to cases where the subject matter changes abruptly and hence the proposed approach in this paper is very likely (or at least expected) to predict the wrong personal life event given it cannot predict ubrupt changes. Table 4 provides further examples for these three types of systematic errors that the proposed approach suffers from.

In addition to the above three types of errors, we also observed some other cases that a human oracle might also have problem in predicting if the actual future tweet was not available. For instance, for the following two tweets: *'Man I'm mad I broke my phone. That s[..]t killing me'* and *'Cant wait until Friday to get my new phone'*, our proposed approach predicted a 'Broken Device', which is in fact correct; however, the user went on to tweet about a new phone and hence a 'Device Upgrade' would have been the more appropriate label, which was not correctly predicted by our approach. It seems that this type of error is similar to Error Type III in the sense that the user moves beyond the immediate scope of the previous tweet onto a different life event; however, it differs from Type III errors in that the transition to the new life event is logical.

Based on these three types of errors, we believe that the work in this paper can be extended in the following directions:

1. The generation architecture proposed in this paper depends on a method to accurately determine the life events of each individual tweet. While there has already been work in the literature that perform life event classification at tweet level, as used in this paper, the performance can still be improved. One of the direction of our future work will focus on learning life event representations in tandem with the Seq2Seq architecture. In other words, instead of performing a pre-processing step to extract the life events for each tweet, we will use an *autoencoder* architecture to learn dense representations for life events. This will have two advantages: 1) it will remove dependence on the life event classifier, and 2) the life event representations will be learnt based on the available training set and will hence be customized for the prediction task.
2. Based on our observation that there are many tweets that have similar components to personal life events but are not personal reports of a life event, we believe that the inclusion of a binary classifier that would determine whether each individual past tweet is a personal report or not could potentially serve as an indicative feature of whether the user will be engaging in a personal report or not. The hypothesis for our future work is that if a user has engaged in a pattern of self reports in the past tweets then it

is more likely to do so in the future as well. This binary classifier can either be used independently to act as a filter or in tandem with the recurrent neural network in the same way that the life event classifier was incorporated in our proposed approach in this paper.

3. Finally, the identified type III errors seem to be related to the fact that recurrent neural networks are more dependent on short-range dependencies as opposed to longer-range dependencies (Bengio et al. 1994). As such, in our model, the latest tweets would have the highest impact on the generated output sequence that would subsequently determine the predicted personal life event. There has been some research that already proposed *hacks* such as reversing the order of the input to bring longer-range dependencies into the short-range (Shi et al. 2016); however, a more systematic approach would be to use *attention models* (Cho et al. 2015). In our future work, we are interested in employing attention models in our work so as to determine which portions of the past tweets or which phrases therein should be considered more heavily within the recurrent neural network. This can potentially address type III errors by lowering attention on less relevant content to the personal life event prediction objective.

7 Concluding remarks

In this paper, we have addressed the problem of predicting future mentions of personal life events on social microblogging platforms, with specific attention to Twitter content. While the literature is abundant with methods that are able to identify and monitor events on social networks at a macro-scale, the detection of personal life events, which are user-specific and have an insignificant social resonance is yet to be fully explored. Within the area of personal life events, a few researchers have shown that it is possible to perform extensive feature engineering to train machine learning classifiers to classify a given tweet into a certain class of life events. Our focus in this paper has been to go beyond the state of the art and predict if and what personal life events will be mentioned in a future and yet unobserved tweet of a given user. To this end, we have proposed two main architectures based on recurrent neural networks that either directly or indirectly predict future personal life event mentions. We have shown in our experiments that this is a non-trivial task and our work is able to provide reasonable performance on a gold standard dataset despite the highly class-imbalanced nature of the personal life event data. We have further explored the errors observed in the obtained results and classified the errors into three primary error type classes, based on which, we have proposed three avenues for future work. In addition, we have only considered six life event types in our work in this paper that would only constitute a subset of all possible personal life event types that can be observed on Twitter. The main reason for this has been the high cost of curating and labeling life event related tweets from Twitter. Future work will need to focus on a larger and broader set of life event types that would show generalizability of this work beyond the six main life event types that have been covered in this paper.

References

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., Bengio, Y. (2012). Theano: new features and speed improvements. In Deep learning and unsupervised feature learning NIPS 2012 Workshop.

Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bollen, J., Mao, H., Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, *2*(1), 1–8.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Byrd, K., Mansurov, A., Baysal, O. (2016). Mining twitter data for influenza detection and surveillance. In Proceedings of the international workshop on software engineering in healthcare systems, SEHS '16, (New York, NY, USA), (pp. 43–49). ACM.

Cameron, M.P., Barrett, P., Stewardson, B. (2016). Can social media predict election results? evidence from new zealand. *Journal of Political Marketing*, *15*(4), 416–432.

Cavalin, P., Gattide Bayser, M., Pinhanez, C. (2014). Towards personalized offers by means of life event detection on social media and entity matching. In HT (Doctoral Consortium/Late-breaking Results/Workshops) (vol. 1210, 01).

Cavalin, P.R., Moyano, L.G., Miranda, P.P. (2015). A multiple classifier system for classifying life events on social media. In 2015 IEEE international conference on Data mining workshop (ICDMW) (pp. 1332–1335). IEEE,.

Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In Eighth workshop on syntax semantics and structure in statistical translation (SSST-8).

Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), (Doha, Qatar) (pp. 1724–1734). Association for Computational Linguistics.

Cho, K., Courville, A., Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, *17*(11), 1875–1886.

Choudhery, D., & Leung, C.K. (2017). Social media mining: prediction of box office revenue. In Proceedings of the 21st international database engineering & applications symposium, IDEAS 2017, (New York, NY, USA) (pp. 20–29). ACM.

Choudhury, S., & Alani, H. (2014a). Detecting presence of personal events in twitter streams. In International conference on social informatics, (pp. 157–166), Springer, Springer International Publishing.

Choudhury, S., & Alani, H. (2014b). Personal life event detection from social media. In Doctoral consortium and workshop proceedings of the 25th ACM hypertext and social media conference, (vol. 1210, 01).

Coppersmith, G., Dredze, M., Harman, C., Hollingshead, K. (2015). From adhd to sad: Analyzing the language of mental health on twitter through self-reported diagnoses. In CLPsych@ HLT-NAACL (pp. 1–10).

DeChoudhury, M., Gamon, M., Counts, S., Horvitz, E. (2013). Predicting depression via social media. *ICWSM*, *13*, 1–10.

Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., Cohen, W.W. (2016). Tweet2vec: character-based distributed representations for social media. In Proceedings of the 54th annual meeting of the association for computational linguistics (vol.2, pp. 269–274). 05.

Dickinson, T., Fernandez, M., Thomas, L.A., Mulholland, P., Briggs, P., Alani, H. (2015). Identifying prominent life events on twitter. In Proceedings of the 8th international conference on knowledge capture, k-CAP 2015, (New York, NY, USA), (pp. 4:1–4:8). ACM.

DiEugenio, B., Green, N., Subba, R. (2013). Detecting life events in feeds from twitter. In 2013 IEEE Seventh international conference on semantic computing (ICSC), (pp. 274–277). IEEE.

Eichstaedt, J.C., Schwartz, H.A., Kern, M.L., Park, G., Labarthe, D.R., Merchant, R.M., Jha, S., Agrawal, M., Dziurzynski, L.A., Sap, M., et al. (2015). Psychological language on twitter predicts county-level heart disease mortality. *Psychological Science*, *26*(2), 159–169.

Fan, W., & Gordon, M.D. (2014). The power of social media analytics. *Communications of the ACM*, *57*(6), 74–81.

Fani, H., Zarrinkalam, F., Bagheri, E., Du, W. (2016). Time-sensitive topic-based communities on twitter. In Canadian conference on artificial intelligence (pp. 192–204). Springer.

Fani, H., Bagheri, E., Du, W. (2017a). Temporally like-minded user community identification through neural embeddings. In 26th ACM international conference on information and knowledge management (CIKM).

Fani, H., Bagheri, E., Zarrinkalam, F., Zhao, X., Du, W. (2017b). Finding diachronic like-minded users. Computational Intelligence.

Franch, F. (2013). (Wisdom of the crowds) 2: 2010 uk election prediction with social media. *Journal of Information Technology & Politics*, *10*(1), 57–71.

Glorot, X., Bordes, A., Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: a deep learning approach. In Proceedings of the 28th international conference on international conference on machine learning, ICML'11, USA (pp. 513–520). Omnipress.
Graves, A. (2012a). Generating sequences with recurrent neural networks. In ICML representation learning workshop.
Graves, A. (2012b). Sequence transduction with recurrent neural networks. In International conference of machine learning (ICML) 2012 workshop on representation learning.
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.
Kaleel, S.B., & Abhari, A. (2015). Cluster-discovery of twitter messages for event detection and trending. *Journal of Computational Science*, *6*, 47–57.
Khodabakhsh, M., Kahani, M., Bagheri, E., Noorian, Z. (2017). Detecting life events from twitter based on temporal semantic features. Knowledge-based Systems Journal. second revision submitted, http://ls3.rnet.ryerson.ca/wiki/images/4/48/Life_Event_Detection.pdf.
Kim, Y. (2014). Convolutional neural networks for sentence classification. In Proceedings of the 2014 conference on empirical methods in natural language processing.
Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In the 3rd International conference for learning representations.
Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q. (2015). From word embeddings to document distances. In Proceedings of the 32Nd International conference on international conference on machine learning - Volume 37, ICML'15 (pp.957–966) JMLR.org.
Lassen, N.B., Madsen, R., Vatrapu, R. (2014). Predicting iphone sales from iphone tweets. In Proceedings of the 2014 IEEE 18th International enterprise distributed object computing conference, EDOC '14, (Washington, DC, USA) (pp. 81–90). IEEE Computer Society.
Lee, J.Y., & Dernoncourt, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguisticss.
Li, J.L., Ritter, A., Cardie, C., Hovy, E.H. (2014). Major life event extraction from twitter based on congratulations/condolences speech acts. In EMNLP.
Li, Y., Liu, T., Jiang, J., Zhang, L. (2016). Hashtag recommendation with topical attention-based lstm. In Proceedings of the 26th international conference on computational linguistics (pp. 943–952), Coling.
Limsopatham, N., & Collier, N. (2016). Bidirectional lstm for named entity recognition in twitter messages. In Proceedings of the 2nd workshop on noisy user-generated text (pp. 145–152).
Lin, H., Jia, J., Guo, Q., Xue, Y., Li, Q., Huang, J., Cai, L., Feng, L. (2014). User-level psychological stress detection from social media using deep neural network. In Proceedings of the 22Nd ACM international conference on multimedia, MM '14, New York (pp. 507–516), ACM.
Liu, T., Ding, X., Chen, Y., Chen, H., Guo, M. (2016). Predicting movie box-office revenues by exploiting large-scale social media content. *Multimedia Tools and Applications*, *75*(3), 1509–1528.
Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.-F., Cha, M. (2016). Detecting rumors from microblogs with recurrent neural networks. In Proceedings of the Twenty-Fifth international joint conference on artificial intelligence, IJCAI'16 (pp 3818–3824), AAAI Press.
Madani, A., Boussaid, O., Zegour, D.E. (2015). Real-time trending topics detection and description from twitter content. *Social Network Analysis and Mining*, *5*, 59.
Mao, Y., Wei, W., Wang, B., Liu, B. (2012). Correlating s&p 500 stocks with twitter data. In Proceedings of the first ACM international workshop on hot topics on interdisciplinary social networks research, HotSocial '12, (New York, NY, USA) (pp. 69–72). ACM.
Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013a). Efficient estimation of word representations in vector space. In Proceedings of workshop at ICLR.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th international conference on neural information processing systems, NIPS'13 USA (pp. 3111–3119). Curran Associates Inc.
Moyano, L.G., Cavalin, P.R., Miranda, P.P. (2015). Life event detection using conversations from social media. In Brazilian workshop on social network analysis and mining.
Nguyen, T.H., & Shirai, K. (2015). Topic modeling based sentiment analysis on social media for stock market prediction. In Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (Volume 1: Long Papers) (pp. 1354–1364). Association for Computational Linguistics.
Nguyen, T.H., Shirai, K., Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, *42*(24), 9603–9611.
Ni, M., He, Q., Gao, J. (2014). Using social media to predict traffic flow under special event conditions. In The 93rd annual meeting of transportation research board.

Nofer, M., & Hinz, O. (2015). Using twitter to predict the stock market. *Business & Information Systems Engineering*, *57*(4), 229–242.

Oak, M., Behera, A., Thomas, T., Alm, C.O., Prud'hommeaux, E., Homan, C., Ptucha, R.W. (2016). Generating clinically relevant texts: A case study on life-changing events. In CLPsych@ HLT-NAACL (pp. 85–94).

Owoputi, O., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In Proceedings of NAACL association for computational linguistics.

Paltoglou, G. (2016). Sentiment-based event detection in twitter. *Journal of the Association for Information Science and Technology*, *67*(7), 1576–1587.

Preotiuc-Pietro, D., Eichstaedt, J., Park, G., Sap, M., Smith, L., Tobolsky, V., Schwartz, H.A., Ungar, L. (2015). The role of personality, age and gender in tweeting about mental illnesses. In NAACL HLT (vol. 2015, pp. 21).

Radosavljevic, V., Grbovic, M., Djuric, N., Bhamidipati, N. (2014). Large-scale world cup 2014 outcome prediction based on tumblr posts. In KDD workshop on large-scale sports analytics.

Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In Proceedings of the conference on empirical methods in natural language processing (pp. 1524–1534) Association for Computational Linguistics.

Schuster, M., & Paliwal, K.K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Semeniuta, S., Severyn, A., Barth, E. (2017). A hybrid convolutional variational autoencoder for text generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, pp. 638–648.

Serban, I.V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., Courville, A.C. (2017). Multiresolution recurrent neural networks: An application to dialogue response generation. In AAAI (pp. 3288–3294).

Severyn, A., & Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, SIGIR '15, New York (pp. 959–962), ACM.

Shang, L., Lu, Z., Li, H. (2015). Neural responding machine for short-text conversation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (pp. 1577–1586). The Association for Computer Linguistics.

Shi, X., Knight, K., Yuret, D. (2016). Why neural translations are the right length. In EMNLP (pp. 2278–2282).

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., GrueSimonsen, J., Nie, J.-Y. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In Proceedings of the 24th ACM international on conference on information and knowledge management, CIKM '15, (New York, USA) (pp. 553–562), ACM.

Sutskever, I., Martens, J., Hinton, G. (2011). Generating text with recurrent neural networks. In Proceedings of the 28th international conference on international conference on machine learning, ICML'11, USA (pp. 1017–1024). Omnipress.

Sutskever, I., Vinyals, O., Le, Q.V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th international conference on neural information processing systems, NIPS'14* (pp. pp. 3104–3112). Cambridge: MIT Press.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd annual meeting of the association for computational linguistics (Volume 1: Long Papers), (Baltimore, Maryland) (pp. 1555–1565). Association for Computational Linguistics.

Tang, D., Qin, B., Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing, Lisbon (pp. 1422–1432). Association for Computational Linguistics.

Tsakalidis, A., Papadopoulos, S., Cristea, A.I., Kompatsiaris, Y. (2015). Predicting elections for multiple countries using twitter and polls. *IEEE Intelligent Systems*, *30*(2), 10–17.

Unankard, S., Li, X., Sharaf, M.A. (2015). Emerging event detection in social networks with location sensitivity. *World Wide Web*, *18*(5), 1393–1417.

Vosoughi, S., Vijayaraghavan, P., Roy, D. (2016). Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval, SIGIR '16, New York (pp. 1041–1044). ACM.

Wald, R., Khoshgoftaar, T.M., Napolitano, A., Sumner, C. (2012). Using twitter content to predict psychopathy. In 2012 11th international conference on machine learning and applications (ICMLA) (vol. 2, pp. 394–401). IEEE.

Werbos, P.J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

Woo, H., Cho, Y., Shim, E., Lee, J.-K., Lee, C.-G., Kim, S.H. (2016). Estimating influenza outbreaks using both search engine query data and social media data in south korea. *Journal of medical Internet research*, *7*, 18.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H. (2016). Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 1480–1489).

Zarrinkalam, F., Fani, H., Bagheri, E., Kahani, M., Di Nunzio, G.M. (2016). Inferring implicit topical interests on twitter. In Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Hauff, C., Silvello, G. (Eds.) *Advances in information retrieval: 38th European conference on IR research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings* (pp. 479–491). Cham: Springer International Publishing.

Zhang, X., Zhao, J., LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th international conference on neural information processing systems, NIPS'15* (pp. 649–657). Cambridge: MIT Press.

Zhao, Z., Lu, H., Cai, D., He, X., Zhuang, Y. (2016). User preference learning for online social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *28*(9), 2522–2534.