Predicting Personal Life Events from Streaming Social Content

Maryam Khodabakhsh
Ferdowsi University of Mashhad
maryamkhodabakhsh@stu.mail.ac.ir

Hossein Fani
University of New Brunswick
hfani@unb.ca

Fattane Zarrinkalam
Ryerson University
fzarrinkalam@ryerson.ca

Ebrahim Bagheri
Ryerson University
bagheri@ryerson.ca

ABSTRACT

Researchers have shown that it is possible to identify reported instances of personal life events from users' social content, e.g., tweets. This is known as *personal life event detection*. In this paper, we take a step forward and explore the possibility of predicting users' *next* personal life event based solely on the their historically reported personal life events, a task which we refer to as *personal life event prediction*. We present a framework for modeling streaming social content for the purpose of personal life event prediction and describe how various instantiations of the framework can be developed to build a life event prediction model. In our extensive experiments, we find that (i) historical personal life events of a user have strong predictive power for determining the user's future life event; (ii) the consideration of *sequence* in historically reported personal life events shows inferior performance compared to models that do not consider sequence, and (iii) the number of historical life events and the length of the past time intervals that are taken into account for making life event predictions can impact prediction performance whereby more recent life events show more relevance for the prediction of future life events.

ACM Reference Format:
Maryam Khodabakhsh, Hossein Fani, Fattane Zarrinkalam, and Ebrahim Bagheri. 2018. Predicting Personal Life Events from Streaming Social Content. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269313

1 INTRODUCTION

Online social networks, such as Twitter, have become one of the mainstream medium for communication and social interaction. As such, the trajectory of a user's personal life events over time, such as graduation, getting married, moving into a new apartment and going on honeymoon, might be observable in one's social timeline as the user moves through different life stages.

While a large body of research has been focused on multimedia content such as images and videos for automatically retelling series of personal life events [9], little work has been done on streaming textual content in online social networks for the purpose of identifying personal life events. The task of *personal life event detection* from user generated textual content in social networks is challenging due to the short, informal and noisy characteristics of social posts, e.g., tweets. Further and more importantly, it suffers from the *sparsity* problem as only few users share their personal life events in publicly accessible platforms such as Twitter. More recently, there have been a few works that have focused on the identification of such personal life events from users' social posts [3, 7, 8]. The objective of these works is to determine whether a given social post, such as a tweet, is describing some personal life event. This can be viewed as a *multi-class* classification task, which is quite difficult to train primarily because of two reasons: 1) there is a *high class imbalance* with regards to life events where a high portion of user's tweets is not about personal life events, as such, the training of the classification model becomes quite difficult; and, 2) there are many cases where the mentioned life event in a social post is not personal, e.g., reporting on a friend's wedding. Such cases require the identifying *self-reporting* social content. Despite these challenges, works in life event detection have effectively used various types of features ranging from user interaction, syntactic, semantic and neural embeddings to train classifiers with reasonable performance.

A logical next step to the work on personal life event detection is future personal life event *prediction* for users based on their historical social content. In other words, while existing work attempt to classify an existing social content into one of the personal life event classes, personal life event prediction would aim to identify the user's next personal life event in a future time interval. The objective of this paper is twofold and as follows: (i) to establish a multidimensional framework for exploring how personal life events can be predicted based on streaming social content, and (ii) to systematically compare the performance of the various dimensions of our proposed framework and discuss how the observed performance points to interesting findings for the personal life event prediction task. We present our observations in this paper by answering three main research questions: RQ1) Whether the sparse historical social content of a given user that report on personal life events in the past has the predictive power to indicate future personal life events; RQ2) Whether the consideration of the sequence of personal life events increases the predictive power of the personal life event prediction task or alternatively, an unordered model, which does not take sequence into account, has a higher predictive power. RQ3) Would the length of the considered time interval or the number of past personal life events of the user be a key factor in the performance of a life event prediction model.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3269313

Table 1: The overview of our proposed framework.

		Temporal	Sliding
Sequence of Events (SoE)		TSoE	SSoE
Bag of Events (BoE)	Disjoint	DTBoE	SDBoE
	Stacked	STBoE	SSBoE

2 PROPOSED FRAMEWORK

Our work in this paper systematically explores whether past observations of personal life events for a specific user can serve as discriminatory features for predicting the user's future personal life event. As such, we rely on existing work in the literature to label historical tweets of a user with appropriate life events. More concretely, given the set of all tweets $\mathbb{M}$, posted by users $\mathbb{U}$ up until time period T, and the set of predefined personal life events $\mathbb{E}$, we require a mapping function $f : \mathbb{M} \longrightarrow \{\mathbb{E}\} \cup \{\lambda\}$ such that $f(m_t^u)$ is the identified personal life event in tweet $m_t^u \in \mathbb{M}$ posted by user $u \in \mathbb{U}$ at time $t \leq$ T. In addition, the case of the λ (nil) life event needs to be considered for those tweets that do not report on any personal life events. There have already been work that provide reasonable estimations of f. Our work is not dependent on any specific life event detection method. We opted to use the personal life event detection method proposed in [7] to learn f. However, any other method such as [3, 8] could also be used.

Once the personal life event of each tweet of all users up to time period T is determined, we create a stream $s_u = [(e, t)_{1:|s_u|}]$ of length $|s_u|$ for user u including all of her personal life events $e \in \mathbb{E}$ such that $(e, t)_i$ is the i^{th} event for the user reported at time t_i. As such, the objective of our work can be formally defined as predicting the next personal life event for user u given s_u.

Our proposed framework, shown in Table 1, systematically explores alternative ways in which s_u can be analyzed for predicting future life events. Within this framework and as shown in the columns of the table, we propose two *sequence selection* strategies, namely *temporal* and *sliding window* strategies. Within the temporal strategy, a fixed number of past time intervals are considered and life events reported by the user in those time intervals are taken into consideration. On the contrary, the sliding window strategy considers a fixed number of past life events reported by the user regardless of how long it took for the user to post the life events.

Orthogonal to how past life events are selected, the rows of the framework show how the selected sequence of life events can be represented. We propose that *life event sequence representation* can be done either as a strictly sequenced set of events where the order of the observed life events is important (Sequence of Events) or as a set of unordered life events where the order of observation of the life events is of no significance (Bag of Events).

Based on these two dimensions, i.e., the framework rows, *sequence representation*, and the framework columns, *sequence selection*, we can define various models as mentioned by the acronyms in the table. For instance, the SSoE model refers to the case when a fixed number of historical life events for the user are strictly taken into account in the same order as they were observed. We further formalize the details of the framework below.

2.1 Sequence Representation

2.1.1 Sequence of Events. When considering past life events with the strict order they were observed, the problem of personal life event prediction can be viewed as an instance of the sequence prediction problem. Therefore, given a subsequence $[(e, t)_{\text{I:J}}]$ from s_u, the personal life events stream of a user u, we aim at predicting the next personal life event $e_{\text{J}+1}$ where $e \in \mathbb{E}$ and $t_{\text{J}} < t_{\text{J}+1}$. When I=1 and J=$|s_u|$, we make predictions based on all the previously seen personal life events in the user's life event stream.

One of the most popular sequence prediction models is prediction by partial matching (ppm) [1], which relies on the Markov property. While this approach has inspired other work such as all-k-order-Markov (akom) [10], it falls short in cases where the Markovian assumption does not hold [5]. Further, ppm and akom do not use all the elements of the training sequence except for the last k elements, called the order of the Markov model, to perform predictions and, hence, their accuracy could be impacted. A possible solution to this problem could be to increase the order of the Markov model, which in turns leads to increased time complexity and makes the methods impractical [2]. Gueniche et al. [5] proposed the compact prediction tree (cpt) method that employs all of the information in the training sequence and makes a prediction by measuring the similarity between subsequences. Despite outperforming ppm and akom in accuracy, the time complexity of cpt remains higher. More recently, the same authors have proposed an update to their previous work, cpt+ [4], which improves the time and space complexity of cpt, yet does not forgo accuracy. Alternatively, long short-term memory (lstm) [6] is an effective recurrent neural network architecture for sequence modeling that uses a series of gates to control what information should be stored as it passes each item in a sequence. We use cpt+ and ppm as non-Markovian and Markovian sequence prediction models, respectively, and lstm as a neural-based method in our *sequence of events* modeling approaches.

2.1.2 Bag of Events. When the order of the past life events is not taken into account, the life event prediction problem can be formulated as an instance of the typical multi-class classification problem where the order of the input subsequence of personal life events is of no significance. We explore this through two strategies. In the first strategy, assuming that an ideal classification method is blind to the order of its input features, we simply treat subsequences of events as features. We refer to this strategy as *Disjoint* because it considers each observed instance of a life event as a feature in isolation. In contrast, in the second strategy, given a subsequence of personal life events, we create a bag of size $\mathbb{E}$ where each personal life event is represented by its total occurrence in the subsequence. We refer to this strategy as *Stacked* because the number of times each life event type is observed is counted and represented as one feature. Given the fact that the bag of events models are instances of the multi-class classification problem, we use the random forest classifier as our learning model.

2.2 Sequence Selection

2.2.1 Sliding Window Selection. In the *sliding window* strategy, we consider a fixed length of past life events in a user's personal life events regardless of their timestamp. Specifically, $\forall 1 \leq i \leq |s_u|$ and a given $(e, t)_i \in s_u = [(e, t)_{1:|s_u|}]$, we create a suffix subsequence $p_u^{w_s}(i) = [e_{i-w_s:i-1}]$ of s_u using a sliding window w_s. This subsequence is then used as the set of life events that will be taken into consideration within the prediction task where we train a predictor

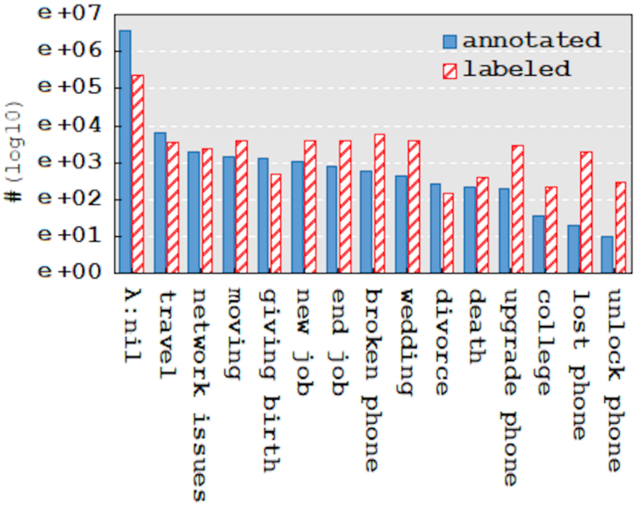

Figure 1: Distribution of personal life events by event class.

function h_s on subsequences with fixed window of length w_s on life event stream s_u for all $u \in \mathbb{U}$ such that $h_s(p_u^{w_s}(i)) \simeq e_i$.

2.2.2 *Temporal Selection.* In the *temporal* strategy, instead of a fixed window of past life events, we collect all the reported personal life events of user u in a given time frame. Formally, $\forall 1 \leq i \leq |s_u|$ and a given $(e, t)_i$ in $s_u = [(e, t)_{1:|s_u|}]$, we create a suffix subsequence $p_u^{w_t}(i)$ of s_u within a time frame w_t where $\forall (e, t) \in p_u^{w_t}(i) : 0 < |t_i - t| \leq w_t$. Given time information is taken into account in this strategy, the subsequence length might not be the same for different users as different users might post dissimilar number of life events within a given time frame. We train a predictor function h_t on subsequences with variable length of personal life events for all $u \in \mathbb{U}$ such that $h_t(p_u^{w_t}(i)) \simeq (e, t)_i$.

Having defined the dimensions of our framework, it is possible to interpret its variants, e.g., STBoE suggests to adopt a stacked bag of events model for sequence representation and a temporal model for sequence selection. We will systematically evaluate the instantiations of our framework in the subsequent section.

3 EXPERIMENTS

3.1 Datasets and Setup

For the evaluation of the personal life event prediction task, we used a publicly available dataset[1] from Twitter, consisting of 3.78 million tweets posted by 1,200 active users since the beginning of their sign up till May 31, 2015. It is worth noting that in the task of personal life event prediction, we need such a dataset in order to be able to create an ideally long sequence of personal life events for each user. We identify the personal life event of each tweet in this dataset by the personal life event detection function f to serve as a *silver standard.* We show the distribution of the personal life events in this *annotated* dataset in Figure 1. As seen, there is a high class imbalance with regards to personal life events where a high portion of user tweets is not about personal life events, which can make the prediction of the next personal life event challenging.

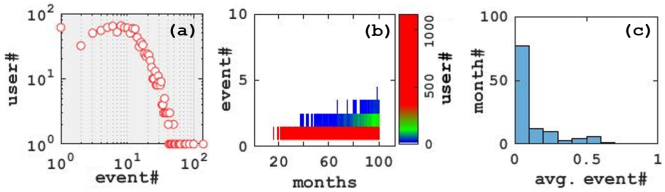

Figure 2: Distribution of users by the number of personal life events (a) overall, (b) per month, and (c) average number of events over all users per month, in the annotated dataset.

Furthermore, we used a dataset consisting of 10 million tweets that were randomly sampled from the Spritzer Twitter stream grab[2] to train the word embeddings required by [7]. Additionally and in collaboration with our industrial partner, a total of 260,061 tweets, not overlapping with any of the above two tweet datasets, were labeled with one of the 14 personal life events or λ(nil) by twenty human evaluators. The list of the 14 life events is shown in Figure 1 along with the distribution of personal life events in our *labeled* dataset. This dataset was used to train the personal life event detection function f proposed in [7]. The average performance of the trained model over the 14 personal life events plus λ in terms of precision, recall, and f-score was 0.382, 0.606, 0.446, respectively.

We created the personal life event stream s_u for all users in the annotated dataset and removed 77 users who had a personal life event stream length of zero from our experiments. The overall distribution of the remaining 1,133 users by number of personal life events as well as the temporal distribution of events over 113 months is shown in Figure 2. As illustrated in Figure 2.a, the majority of the users have a personal life event stream of length 16 or less. While there is growing trend in the number of personal life events as the months pass, it still remains less than 2 for the majority of users as shown in Figure 2.b. In Figure 2.c, we show that in ~ 70% of the months, the probability of reporting a personal life event is less than 0.1. This clearly shows the *sparsity* problem.

In terms of implementation details, the different variation of our framework as mentioned in Table 2 were implemented as either a sequence prediction model or a multi-class classification method. For the variants where a *sequence of events* sequence representation method is selected, i.e., TSoE and SSoE, they are implemented as sequence prediction using three different algorithms including cp+, ppm and lstm. We used the spmf pattern mining Java library for the cpt+ and ppm methods. We used Keras to implement our lstm method with two stacked hidden lstm layers of 50 memory units. The output layer consists of 14 units (the size of our predefined set of personal life event set $\mathbb{E}$) with the softmax activation function. With respect to the variants where *bag of events* sequence representation approach is adopted, i.e., DTBoE, SDBoE, STBoE and SSBoE, a multi-class classification technique is used. We trained separate random forest classifiers for each of these variants. The scikit-learn python machine learning library was used to run random forest with 30 decision trees of maximum depth 10 and employed Gini impurity as the information gain metric. All the other parameters of these methods were set to defaults. Our reported results are based on

[1] bitbucket.org/sliang1/uct-dataset/get/UCT-Dataset.zip

[2] archive.org/details/archiveteam-twitter-stream-2012-01

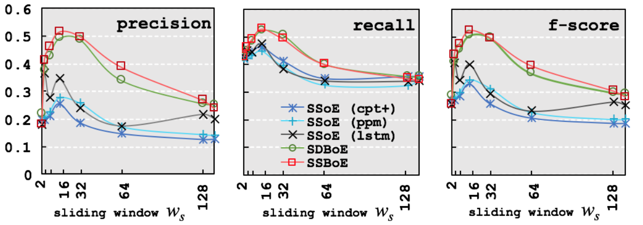

Figure 3: Comparative results of the *sliding* strategy.

ten-fold cross validation where in each fold the original distribution of the different life events in testing and training sets are preserved.

3.2 Findings

We present the performance of the different approaches in Figures 3 and 4 for different sliding and temporal window sizes. The results of the experiments allow us to answer our three research questions. Regarding RQ 1, our observations indicate that despite the sparse historical personal life events of users, it is still possible to predict future personal life events using the variants of our proposed framework. For instance, the SDBoE and DTBoE variants showed an f-score of 0.51 across 14 different classes of life events. Our observation regarding **RQ1** is that the sparse historical social content of a user that mention her personal life events in the past has a reasonable predictive power to indicate future personal life events.

To answer **RQ2**, we compare the variants of each of the sequence representation techniques of our framework. Based on the results reported in Figures 3 and 4, regardless of the sequence selection approach, we observe that none of the sequence modeling approaches, i.e., TSoE and SSoE, were able to outperform the bag of event approaches. As such and with regards to the second research question **RQ2**, we find that the consideration of *ordered sequence* in the personal life events does not improve predictive power and unordered models in the form of *bag of events* have higher predictive power. This might be partly attributed to the *sparsity* problem as the users do not follow any sequential pattern when posting about their personal life events, e.g. reporting on some life events and not on others or not respecting the actual order when reporting events.

Finally, to address our third research question **RQ3** and to analyze the impact of the number of past life events or the length of the past time intervals on prediction performance, we evaluated the various variants based on differing temporal and sliding window sizes. In the sliding window strategy, as illustrated in Figure 3, all the baselines show more or less similar behaviour. They reach their maximum performance in $w_s = 16$ and decline gradually as the w_s increases. This explains that a future personal life event is more influenced by its more recent events rather than all the preceding ones. Likewise in the temporal strategy, as illustrated in Figure 4, we observe that for most of the methods, recent months are contributing more to the prediction performance. In the context of **RQ3**, these findings show that the length of the past time interval and the number of past life events do impact the performance of the prediction model and indicate that more recent life events have a higher predictive power for performing life event prediction.

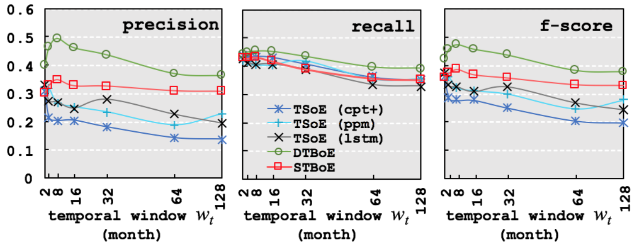

Figure 4: Comparative results of the *temporal* strategy.

4 CONCLUDING REMARKS

This paper focuses on the prediction of a user's future life event based on the historically reported life events of that user. We have presented a framework to show how streaming social content can be modeled in different ways for predicting future life events. In addition, the paper reports on strong baseline implementations of the variants of the proposed framework and reports its findings through three research questions. Summarily, we found that it is possible to predict the next personal life event of a user given her past historical life events with an f-score of over 50% on a 14-class personal life event dataset; therefore, showing that historical life events do have strong predictive power for determining future life events. We also found that models that consider the sequence of historical life events do not show competitive performance to the models that overlook sequence. The difference between the best performing sequence of events and bag of life events models is significant with a difference of ~10% on f-score. This could be partially explained by the fact that life events are *sparse* on a user's timeline and considering sequence would further limit the generalizability of the observed content and hence lead to a poorer performance. Finally, our results show that prediction performance is highly influenced by the length of past historical events that are considered. This can be due to the staleness of older life event information.

REFERENCES

[1] John G. Cleary and Ian H. Witten. 1984. Data Compression Using Adaptive Coding and Partial String Matching. *IEEE Trans Comm* 32, 4 (1984), 396–402.

[2] Mukund Deshpande and George Karypis. 2004. Selective Markov models for predicting Web page accesses. *ACM Trans. Internet Techn.* 4, 2 (2004), 163–184.

[3] Thomas Dickinson, Miriam Fernández, Lisa A. Thomas, Paul Mulholland, Pam Briggs, and Harith Alani. 2015. Identifying Prominent Life Events on Twitter. In *K-CAP*. 4:1–4:8.

[4] Ted Gueniche, Philippe Fournier-Viger, Rajeev Raman, and Vincent S. Tseng. 2015. CPT+: Decreasing the Time/Space Complexity of the Compact Prediction Tree. In *PAKDD 2015*. 625–636.

[5] Ted Gueniche, Philippe Fournier-Viger, and Vincent S. Tseng. 2013. Compact Prediction Tree: A Lossless Model for Accurate Sequence Prediction. In *ADMA 2013*. 177–188.

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[7] Maryam Khodabakhsh, Mohsen Kahani, Ebrahim Bagheri, and Zeinab Noorian. 2018. Detecting life events from twitter based on temporal semantic features. *Knowl.-Based Syst.* 148 (2018), 1–16.

[8] Jiwei Li, Alan Ritter, Claire Cardie, and Eduard H. Hovy. 2014. Major Life Event Extraction from Twitter based on Congratulations/Condolences Speech Acts. In *EMNLP 2014*. 1997–2007.

[9] Vasileios Mezaris, Ansgar Scherp, Ramesh Jain, and Mohan S. Kankanhalli. 2014. Real-life events in multimedia: detection, representation, retrieval, and applications. *Multimedia Tools and Applications* 70, 1 (2014), 1–6.

[10] James E. Pitkow and Peter Pirolli. 1999. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. In *USITS'99*.