Personal Knowledge Base Construction from Text-based Lifelogs

An-Zi Yen[1], Hen-Hsen Huang[23], Hsin-Hsi Chen[13]
[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
[2]Department of Computer Science, National Chengchi University, Taipei, Taiwan
[3]MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
azyen@nlg.csie.ntu.edu.tw, hhhuang@nccu.edu.tw, hhchen@ntu.edu.tw

ABSTRACT

Previous work on lifelogging focuses on life event extraction from image, audio, and video data via wearable sensors. In contrast to wearing an extra camera to record daily life, people are used to log their life on social media platforms. In this paper, we aim to extract life events from textual data shared on Twitter and construct personal knowledge bases of individuals. The issues to be tackled include (1) not all text descriptions are related to life events, (2) life events in a text description can be expressed explicitly or implicitly, (3) the predicates in the implicit events are often absent, and (4) the mapping from natural language predicates to knowledge base relations may be ambiguous. A joint learning approach is proposed to detect life events in tweets and extract event components including subjects, predicates, objects, and time expressions. Finally, the extracted information is transformed to knowledge base facts. The evaluation is performed on a collection of lifelogs from 18 Twitter users. Experimental results show our proposed system is effective in life event extraction, and the constructed personal knowledge bases are expected to be useful to memory recall applications.

CCS CONCEPTS

• Computing methodologies → Information extraction • Information systems → Personalization

KEYWORDS

Lifelogging, Life event detection, Personal knowledge base construction, Social media

ACM Reference format:

An-Zi Yen, Hen-Hsen Huang, Hsin-Hsi Chen. 2019. Personal Knowledge Base Construction from Text-based Lifelogs. In *SIGIR '19: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331209

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07...$15.00
https://doi.org/10.1145/3331184.3331209

1. INTRODUCTION

Life event extraction from social media data provides complementary information for individuals. For example, in the tweet (T1), there is a life event of the user who reads the book entitled "Miracles of the Namiya General Store" and enjoys it. The enriched repository personal information is useful for memory recall and supports living assistance.

(T1) 東野圭吾的《解憂雜貨店》真好看 (Higashino Keigo's "Miracles of the Namiya General Store" is really nice.)

Several researches have been done for life event detection from social media [4, 8, 22, 25, 39]. However, most of them focus on the detection of major life events such as marriage, job promotions, exam, and graduation. General life events such as dining, visiting a local place, and having a talk with friends, which consist of important information for recall and retrieval, remain to be solved.

This work presents a comprehensive investigation on the topic of life event extraction on text-based lifelogs. In this paper, tweets, the short messages published and shared on the social media platform Twitter, are adopted as the source of lifelogs. We propose a system to detect whether a life event exists in a tweet, and extract the possible life events in the quadruple form (subject, predicate, object, time). Besides, we further transform natural language (NL) predicates into knowledge base (KB) relations such as Perception_active, Motion, and Presence selected from Chinese FrameNet [39]. Timestamped subject-relation-object facts form a personal knowledge base over timelines.

The key challenge in event detection from social media data is that the user-generated text is often brief and informal-written. Life events may not always be explicitly expressed. An explicit life event contains the exact information about "Who did What to Whom Where When and How". Thus words related to subject, predicate, object, and time in a tweet can be directly extracted to compose a quadruple describing the life event. On the other hand, there is no clear expression in an implicit life event, so that it is more challenging to identify the components in the quadruple.

Tweet (T2) contains two explicit life events, which can be represented in the quadruples (I, went to, KFC, noon) and (I, ate, hamburger, noon). The NL predicates "went to" and "ate" can be further transformed to the KB relations Self_motion and Ingestion, respectively. Finally, the facts Self_motion(I, KFC, noon) and

Ingestion(I, hamburger, noon) will be stored in a personal knowledge base for subsequent applications.

(T2) I went to KFC, and ate a hamburger for lunch.

(T3) iPhone X!

Tweet (T3) is an instance of implicit life event, in which no explicit predicate is mentioned. The subject is the author, the object is iPhone X, and the time is the timestamp of the tweet. However, the predicates can be "bought" or "released", which are the potential actions to the cell phone. In other words, this short description may present either the user bought a new cell phone or the iPhone X was just released. The former denotes a life event of the user, while the latter does not. In our system, the life event extraction is performed in two stages. Given a tweet, the first stage detects the implicit and the explicit life events and suggests the predicate of each life event. We formulate the related subtasks in the first stage in a novel neural network and train the subtasks in a fashion of joint learning. The second stage extracts the subject, object, and time expression for each predicate and generates the results in quadruples. The semantic role of each element participating a life event is further labeled with a semantic parser based on Chinese FrameNet. Finally, the outcomes of our system are ready to insert into a personal knowledge base.

Various approaches to question answering over knowledge base have been explored in recent years. Thus, storing life events in a knowledge base can benefit from the progress of previous research such as complex question answering. For example, the user is able to query a series of life events across timeline. Fig. 1 shows the timeline of a user who rented a bicycle and rode a bicycle from Guandu to Keelung.

Figure 1: A fragment timeline of lifelogs.

The user might query "Where did I eat before riding a bicycle to Tianshan Farm?" or "Where did I leave for the Keelung Miaokou?". Directly retrieving the tweet may not provide the correct answer to this kind of questions. Question answering over knowledge base is a fast-growing area so that more novel applications based on life events are expected to be proposed in the future.

The contributions of this paper are threefold. (1) This work introduces a new task that addresses textual lifelog mining on the real world social media data. (2) We propose a comprehensive system for the extraction. Related subtasks in the extraction workflow are formulated and modeled with a joint learning approach. Experimental results show promising performances are achieved in all the subtasks. (3) We demonstrate how to construct a personal knowledge base for general life events, providing complementary information for recall and retrieval.

The rest of this paper is organized as follows. In Section 2, we survey the related work about lifelogging and event extraction. Section 3 introduces the lifelog corpus used in experiments. The construction and the statistics of the corpus are described. Section 4 presents our system for life event extraction and personal knowledge base construction. Experimental results are shown and discussed in Section 5. Detailed analysis of experimental results is discussed in Section 6. Finally, Section 7 concludes the remarks.

2. RELATED WORK

In recent years, the visual lifelogs, captured through the devices such as Sensecam [16] and Narrative [19], have been investigated in a variety of applications including aiding human memory recall [14], healthcare [20, 29, 34], diet monitoring [28, 33], and self-reflection [9, 15].

This paper focuses on the extraction of text-based life events from social media. Social media platforms like Facebook and Twitter provide the service for people to log their life events. Previous work has addressed the public event extraction from social media data, including the extraction of disaster outbreak [37], elections [35], news events [18], and music events [27].

In contrast to the extraction of large-scale, public events from social media, the detection of personal life events has also been explored. However, previous work focuses on the detection of major life events such as marriage and graduation. The work of Li et al. [25] collects the tweets replied with congratulations or condolences speech acts, including the phrases "Congrats", "Sorry to hear that", and "Awesome", and proposes a model to identify major life events. The work of Li and Cardie [22] proposes a model to extract major life events like job promotions, and generate timeline for individuals based on tweets.

The work of Choudhury and Alani [4] classifies 11 major life events, including marriage, job promotions, and so on. They train a classifier by using activity and attention features. The work of Dickinson et al. [8] transforms the representation of tweets as syntactic and semantic graphs and identifies the life event such as getting married, and death of a parent. The work of Sanagavarapu et al. [38] is aimed at predicting whether some people participate in an event and identifying when an event happens.

Gurrin et al. [11] release personal lifelog data for the NTCIR12-Lifelog task which are logged by wearable camera. They employed this test collection to search and retrieve personal specific moment from lifelogs, and to explore knowledge mining and gain insights into the lifelogger's daily life activities. The ImageCLEF2018lifeLog dataset [7] consists of 50 days of image data from a lifelogger employed in NTCIR-13 for analyzing the lifelog data and summarizing certain life events for a lifelogger, and retrieving a number of specific moments in a lifelog, such as the moments of shopping in a wine store.

Event detection is a challenging task in information extraction. Nguyen and Grishman [32] utilize convolutional neural networks and to label entity type of each token in the sentence for event

Table 1. Examples of the two types of tweets.

Tweet	Containing Life Event	Life Event Quadruples (Subject, Predicate, Object, Time)	Explicitness
好天氣 (The weather is good.)	Without life event	X	X
出去看一頁台北 (Go out to watch Yi Ye Taipei.)	With life event	(使用者[User], 出去[go out], X, timestamp)	Explicit
		(使用者[User], 看[watch], 一頁台北[Yi Ye Taipei], timestamp)	Explicit
米，貢丸湯，午餐 (Rice, meat ball soup, lunch.)	With life event	(使用者[User], 吃[eat], 米[rice], timestamp)	Implicit
		(使用者[User], 吃[eat], 貢丸湯[meat ball soup], timestamp)	Implicit
		(使用者[User], 吃[eat], 午餐[lunch], timestamp)	Implicit
新店瓦城用餐中 (Eating in Xindian Wa Cheng.)	With life event	(使用者[User], 在[at], 新店瓦城[Xindian Wa Cheng], timestamp)	Implicit
		(使用者[User], 用餐[eat], X, timestamp)	Explicit

detection. Chen et al. [5] present a dynamic multi-pooling convolutional neural networks to extract event mentions, triggers, and arguments. Nguyen et al. [31] propose a joint framework with bidirectional recurrent neural networks to jointly label event triggers and argument roles.

In contrast to the previous works, which deal with public event or major life event detection, our work focuses on general life events. We propose a system that detects and extracts general life events from tweets, and further construct personal knowledge base for individuals. The personal knowledge base can be merged with large scale structured KBs such as Freebase [2] and DBpedia [1], so that the personal life events are connected with world knowledge. The memory recall service can be implemented on the basis of personal and world knowledge, and the QA systems over knowledge base [17, 40].

A relation in KB may be expressed by different predicates in NL statements. An NL predicate may be mapped to different KB relations. The work of Lin et al. [23] addresses the vocabulary gap between NL and KB and proposes a word embedding approach to deal with the gap. Considering the fact that different users have different ways of expressing life events, we introduce users' metadata into our system for predicting KB relations.

3. LIFELOG CORPUS CONSTRUCTION

To the best of our knowledge, no public lifelog corpus is available for text-based general life event extraction. In this paper, we construct a corpus based on tweets. For training and testing our two-stage system, three levels of annotations have been done.

The annotator is asked to annotate the life event in the FrameNet ontology. For each tweet, an annotator labels the following information. (1) Whether the Twitter user describes one or more personal life events in this tweet. (2) For the tweet with life event(s), the annotator specifies the subject, predicate, object, and time of each life event. The explicitness of the life event, i.e., explicit or implicit, is also labeled. Here subject, predicate, and object describe the Twitter user did what to whom, and time indicates when the life event happened. In this lifelog corpus, most of the subjects are the Twitter users, and most of times are the timestamps of the tweets. (3) The annotators consult Chinese FrameNet [39] and select a suitable frame name for each predicate and label semantic roles following the definition of Chinese FrameNet. For instance, the predicates "download", "get", and "receive" suggest the KB relation "Getting". Finally, we collect 25,344 Chinese tweets from 18 selected users.

With the above annotation, a tweet is classified into two types: the tweet with life event(s) and that without any life events. Table 1 shows examples for each of the two types. A tweet without any life event means it does not log a life event. For instance, the tweet mentioning a world event or an opinion of public issue belongs to this type. A tweet with life event(s) might contain more than one life event. Each life event is further classified into two types according to explicitness: explicit and implicit. An explicit life event includes a predicate that denotes the relation between the subject and the object. For example, the second tweet in Table 1, the user explicitly expresses the life events by predicates 出去 (go out) and 看 (watch). An implicit life event is expressed without a predicate in the text. Taking the third tweet in Table 1 as an example, it represents the user ate rice and meat ball soup for lunch. No predicate like eat is overtly used to express what the user did.

As a result, the numbers of the tweets with and without life event are 16,429 and 8,915, respectively. The number of explicit and implicit life events labeled by annotators are 12,064 and 3,461, respectively. To examine the annotation quality, 100 reference tweets are selected and carefully annotated by a supervisor annotator, and these tweets are included in all annotators' batches. Thus, the same 100 reference tweets will be labeled by all annotators.

We measure the agreement of each annotator with the supervisor annotator by using the Cohen's kappa and F-score. The average agreement of the types of tweets, subjects, predicates, objects, times, KB relations, and semantic roles are show in Table 2. The agreement on type of tweets is substantial, and all the agreements on the other components measured in F-scores are higher than 0.7.

Table 2. Average agreement of annotations.

Components	Metric	Value
Type of tweets	Cohen's kappa	0.6341
KB relations	F-score	0.7481
Subjects	F-score	0.8490
Predicates	F-score	0.7817
Objects	F-score	0.7231
Times	F-score	0.8236
Role of subjects	F-score	0.8490
Role of objects	F-score	0.7167
Role of times	F-score	0.7993

Finally, 137 unique frame names are selected and regarded as KB relations. Table 3 shows the top 10 frequent KB relations and their related predicates.

Table 3. Most frequent KB relations and their related predicates.

KB relations	Related Predicates	Frequency
Perception_active	看 (see), 聽 (listen)	1991
Presence	在 (in), @ (at)	1373
Using	用 (use), 整 (use), 拿 (take)	1263
Motion	到 (go to), 去 (go to), 回(back)	1027
Ingestion	吃 (eat), 喝(drink), 咬(bite)	779
Telling	說 (tell), 講(tell), 告訴(tell)	686
Sending	貼 (post),傳 (send), 寄 (send)	482
Commerce_buy	買 (buy), 下 (bid), 訂購 (order)	478
Creat_representation	拍照 (take phtot)	367
Getting	下(download), 收到(receive)	316

We observe that users usually log their life events about what they saw, what they heard, and where they were. Especially, Twitter users often use the symbol "@" to denote the meaning of "at". Note that some NL predicates are associated with more than one KB relation. For example, the predicate "下" with the meanings of "bid" or "download" in informal writing can be mapped to two KB relations "Commerce_buy" and "Getting".

4. METHODOLOGY

In this paper, we aim to extract life events from tweets and represent them in the format of frame semantics and transform NL into knowledge base facts. A number of tasks have to be done to achieve our goal.

4.1 System Overview

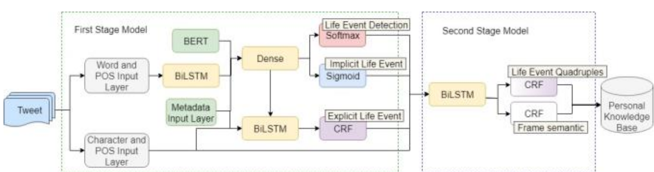

Figure 2: Overview of our personal knowledge base construction system

Figure 2 shows an overview of our system. Overall, the workflow is divided into two stages. The first stage includes three subtasks. The first subtask is aimed at deciding if the tweet contains life events.

For the tweet with life event(s), the second subtask recognizes all the predicates that trigger explicit life events, and the third subtask recognizes the implicit life events and predicts a suitable predicate for every implicit life event. The outcomes of the first stage are $n+m$ predicates, indicating n explicit and m implicit life events, respectively. We propose a joint learning approach to these three tasks since they are highly related. The following three subtasks are trained simultaneously, and parts of their layers are shared for achieving better generalization.

- Life event detection: this subtask identifies whether the tweet contains a life event. The tweets without any life event will be filtered out. This subtask is regarded as a problem of binary classification.
- Explicit life event recognition: this subtask extracts all the predicates that trigger explicit life events. Every predicate is further mapped to a KB relation. We formulate this subtask to be a sequence labeling problem with the BIO scheme. For each word in the tweet, the model will label it with one of the three labels: Begin, Inside, and Outside. The Begin denotes the first word of a predicate, and the following Insides denote the rest of the words of the predicate. A word is labeled as Outside if it is not a part of any predicate.
- Implicit life event recognition: this subtask identifies all implicit life events in the tweet and looks up a KB relation for each implicit life event. This subtask is regarded as a problem of multi-label classification.

In the second stage, the system extracts subject, object, and time in tweet for each predicate. Considering that the frame semantics is related to quadruples generation, the system also parses a tweet with frame semantics to obtain the semantic roles, which can be mapped to subject, object, and time. Frame semantics parsing provides additional quadruples to add into the knowledge base for better coverage. The second stage results $(n+m) \times j$ quadruples, where j is the number of objects extracted by the system. The two subtasks in the second stage are described as follows.

- Life event quadruple generation: this subtask identifies the subject, object, and time expression for each predicate, representing a life event in the tweet. Obviously, this is a sequence labeling problem. Given the predicate, the model will label the spans of subject, object, and the time expression. We extend the BIO scheme to seven labels: B-Subject, I-Subject, B-Object, I-Object, B-Time, I-Time, and Outside.
- Frame semantics parsing: this subtask fulfills semantic roles according to the definition of Chinese FrameNet. This is another sequence labeling. Given the frame, the model will label the spans of frame elements. Each frame contains different frame elements. For instance, the frame elements in Presence are entity, location, time, and so on.

As a result, the system generates life event quadruples and transforms them to the facts for storing in the personal knowledge base for the Twitter user.

In the architecture of our joint learning approach, the input layer and the sentence representation are shared among subtasks, and each subtask has a private task-specific output network for its goal. For the subtask of life event detection, the softmax layer is used as the output layer. For the subtask of implicit life event recognition, the sigmoid layer is used. For those subtasks regarded as sequence labeling, we combine the bidirectional LSTM (BiLSTM) and the conditional random field (CRF) model.

4.2 Multitask Learning

Multitask learning (MTL) [3] has been shown to be effective in learning better representations in various NLP tasks [26, 36]. The basic idea of MTL is that training the model for multiple related tasks simultaneously enables the model to learn a more generalized representation and reduces the issue of overfitting. Given the 3 related tasks in the first stage and the 2 related tasks in the second stage, we define φ and τ as cost functions, respectively. We use cross entropy as the cost function of classification task in our model. For the sequence labeling, we exploit the negative log likelihood objective as cost function. The global cost function is the weighted sum of the cost of each task:

$$\varphi = w_T \times \left(-\sum_{i}^{n}\sum_{c=1}^{C} y_{i,c} * log(\hat{y}_{i,c}) \right) + w_E \times (-log(\bar{y}_E|X))) + w_I \times \left(-\sum_{i}^{n}\sum_{f=1}^{F} y_{i,f} \times log(\hat{y}_{i,f}) \right) \quad (1)$$

$$\tau = w_Q \times (-log(\bar{y}_Q|X))) + w_s * (-log(\bar{y}_S|X))) \quad (2)$$

where w_T, w_E, w_I, w_Q, and w_s denote the weights for event type identification, explicit life evnet extraction, and implicit life event recognition, life event quadruple generation, and frame semantics parsing. After tuning the weights by validation data, we set w_T, w_E, and w_I to 0.4, 0.3, and 0.3, respectively. w_Q, and w_s are set to 0.6 and 0.4, respectively. n is the number of inputs. $C = 2$ is the number of classes in event type identification. $F = 137$ is the number of KB relations in our dataset. y denotes labels, and $\hat{y}$ denotes the prediction probabilities of our model. $\bar{y}$ is the annotated label sequences, and X is input sequences.

4.3 Conditional Random Field

A condition random field [24] focuses on sequence labeling in which the states of neighboring tags are taken into account instead of modeling tagging decisions at each time step.

Let $X = x_1, x_2, \dots, x_n$ be a sequence of words and $Y = y_1, y_2, \dots, y_n$ be the corresponding sequence of labels, the conditional probability of a linear chain CRF is defined as follows:

$$P_\lambda(Y|X) = \frac{1}{z_{\lambda(x)}} exp\left(\sum_{t=1}^{n}\sum_{k} \lambda_k f_k\,(y_{t-1}, y_t, X, t) \right) \quad (3)$$

where $z_{\lambda(x)}$ is the per-input normalization, $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$ trainable parameters associated with feature functions $f = \{f_1, f_2, \dots, f_k\}$, and t denotes time step. The most probable label sequence for an input X is got by computing:

$$\hat{y} = arg\,\max_{y} P_\lambda(y|X) \quad (4)$$

The decoding process can be efficiently computed by using the Viterbi algorithm.

4.4 Bidirectional LSTM-CRF Network

LSTM [13] is a kind of recurrent neural network (RNN) that is usually applied on sequential data. Different from LSTM, bidirectional LSTM [12] can also generate a representation of the right context $\overleftarrow{ht}$ by the forward and the backward LSTM layers. The benefit of the bidirectional LSTM is the additional information from the reversed sequence for a given time step to do sequence labeling task.

Our final sequence labeling model combines the BiLSTM and the CRF models. The BiLSTM-CRF network can derive past and future features from inputs efficiently from BiLSTM layer, and CRF layer predicts an optimal sequence of labels by using features extracted from BiLSTM layer.

For a sequence of predictions y, Lample and Ballesteros [21] define the score:

$$s(X, y) = \sum_{i=0}^{n-1} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \quad (5)$$

where P is the matrix of scores output by the bidirectional LSTM network, $P_{i,j}$ corresponds to the score of the j^{th} tag of the i^{th} word in a sentence. A is a transition score matrix in which $A_{i,j}$ denotes the transition score of tag i to tag j. Then, we can use the dynamic programming to compute $A_{i,j}$ and predict optimal sequence of labels.

4.5 Feature Representations

The input of our model is either in the word level or in the character level. Besides the embedding of the word or the character, we further enrich the input features with linguistic information such as part-of-speech (POS) tags. POS tagging is performed on the tweet with the Stanford POS tagger [30], and the one-hot representation of each POS tag is concatenated with the embedding(s) of the corresponding word or the corresponding characters.

We also consider the metadata of the tweet as features. For capturing a user's habits, such as what days the user is used to riding a bicycle and what time the user goes to school. We input user account to indicate who posted the tweet and the timestamp to indicate when the tweet was posted.

Furthermore, we explore the latest pre-trained sentence representation, Bidirectional Encoder Representations from Transformers (BERT) [6], in this work. The final hidden state output from the BERT is taken as features. We fine-tune BERT with life event detection and extract the output of final hidden state.

5. EXPERIMENTS

We first evaluate the performance of the two stages individually. The results of the first stage are shown in Sections 5.1, 5.2, 5.3 and 5.4. And the result of the second stage is shown in Sections 5.5. We first evaluate the models in these two stages independently. In Section 5.6, the end-to-end performance of our system is evaluated.

For each user, we sort her/his tweets by timestamps and use the first two-thirds of tweets for training and the rest of tweets for testing. The tweets in the last one-third of training set are further held out as validation data. As a result, the sizes of the training data, validation data, and test data are 11,260, 5,631, and 8,453 tweets, respectively. In the test set, there are 5,478 tweets without life event and 2,975 tweets with at least one life event. In the experiments, F-score is the main metric for performance evaluation.

5.1 Life Event Detection

In this section, we evaluate the performances of the models for the three subtasks in the first stage, i.e., life event detection, explicit life event recognition, and implicit life event recognition. We compare our joint learning models under different settings with the baseline models that are individually trained for the three subtasks. Table 4 reports the results of life event detection with Accuracy (A), Precision (P), Recall (R), and F-score (F1), respectively. The first four rows show the performances of the baseline models, and the following rows show the performances of our joint learning models.

In the subtask of life event detection, the joint learning models are generally superior to the baseline models. The best setting is MTL-LSTM-CRF with all features, which achieves an F-score of 93.96% and significantly outperforms all the baseline models with p<0.001 using the McNemar's test.

Table 4. Performance of life event detection.

Models	A	F1	P	R
Majority	64.81%	N/A	N/A	N/A
LSTM	94.55%	92.10%	93.92%	90.35%
BiLSTM	94.56%	92.12%	**93.98%**	90.32%
MTL-LSTM	94.96%	92.94%	91.64%	94.29%
MTL-LSTM-CRF	**95.72%**	**93.96%**	93.25%	94.69%
MTL-BiLSTM	95.63%	93.91%	92.25%	**95.63%**
MTL-BiLSTM-CRF	95.30%	93.47%	91.58%	95.43%

5.2 Explicit Life Event Recognition

In our lifelog corpus, there are 137 types of KB relations to represent personal life events. One challenge of explicit life event recognition is the ambiguity of the mapping between the textual predicates and the KB relations. Table 5 shows the number of predicates that have the ambiguous problem. There are 302 predicates mapped to more than one KB relation.

Table 5:. Number of predicates mapping to one or more than one KB relation.

	Number of predicates
Only mapped to one KB relation	1,723
Mapped to more than one KB relation	302

Especially, the predicates "開" and "打" express a variety of meanings, i.e., up to 17 KB relations. For instance, the predicate "開" can be mapped to Board_vehicle, Using, and Change_operational_state. The predicate "打" can be mapped to Board_vehicle, Contacting, and Hit_target.

We present the average performances of explicit life event recognition in Table 6. The evaluation criterion is that predicate and its corresponding KB relation must be matched to the annotation exactly. The test data is the tweets with life events. In other words, we do not consider error propagation [10] from life event detection in this subsection. The pipelined system evaluation is presented in Section 5.6. The results show that most joint learning models are superior to the baseline models. The MTL-BiLSTM-CRF model achieves an F-score of 39.23%, significantly outperforming all the baseline models with p<0.001 using the McNemar's test.

Table 6. Performance of explicit event recognition.

Models	F1	P	R
LSTM	30.61%	34.43%	29.83%
LSTM-CRF	31.90%	36.16%	30.77%
BiLSTM	34.74%	38.02%	34.68%
BiLSTM-CRF	35.70%	39.34%	35.37%
MTL-LSTM	33.93%	38.88%	32.60%
MTL-LSTM-CRF	34.56%	39.18%	33.43%
MTL-BiLSTM	37.42%	41.57%	36.53%
MTL-BiLSTM-CRF	**39.23%**	**43.40%**	**38.58%**

5.3 Implicit Life Event Recognition

We report the average performances of implicit life event recognition in Table 7.

Table 7. Performance of implicit event recognition.

Models	F1	P	R
Majority	61.58%	61.58%	61.58%
LSTM	72.70%	72.48%	73.96%
BiLSTM	74.49%	74.15%	76.10%
MTL-LSTM	71.86%	71.68%	73.23%
MTL-LSTM-CRF	71.74%	71.43%	73.36%
MTL-BiLSTM	73.25%	73.48%	74.10%
MTL-BiLSTM-CRF	**80.99%**	**77.68%**	**89.25%**

Similarly, only the tweets with life events are used as test data in this subsection. In this task, the best model is MTL-BiLSTM-CRF which achieves an F-score of 80.99% and significantly outperforms all the baseline models and the other settings with p<0.001 using the McNemar's test. It represents that joint learning is capable of leveraging useful information by training multiple related tasks simultaneously and results in improvements of all the tasks.

As mentioned in Section 1, implicit life event recognition is a challenge task because implicit life events are mostly expressed in informal and incomplete short messages. However, the performances of implicit event recognition seem to be better than those of explicit event recognition by comparing Table 6 with Table 7. The reason is that the distribution of implicit life event is highly sparse. The F-score would be 61.58% for a classifier that always predicted the tweet as that without implicit life event.

Table 8. Performance of life event quadruples generation.

Models	All				Explicit life event quadruples			Implicit life event quadruples		
	Overall (F1)	Subject (F1)	Object (F1)	Time (F1)	Subject (F1)	Object (F1)	Time (F1)	Subject (F1)	Object (F1)	Time (F1)
LSTM	38.97%	55.25%	40.43%	19.65%	45.79%	38.28%	20.74%	79.15%	46.61%	14.23%
LSTM-CRF	43.90%	57.72%	42.47%	37.64%	52.36%	40.97%	38.92%	71.79%	46.80%	27.74%
BiLSTM	40.12%	53.12%	39.44%	33.53%	39.87%	35.93%	35.67%	84.07%	49.46%	18.67%
BiLSTM-CRF	54.68%	71.53%	55.95%	29.42%	64.42%	54.97%	31.38%	88.71%	58.78%	13.98%
MTL-LSTM	39.57%	51.41%	41.65%	31.61%	45.59%	41.53%	23.59%	66.11%	41.97%	23.84%
MTL-LSTM-CRF	55.79%	66.38%	55.39%	**43.85%**	58.72%	53.99%	**45.69%**	86.81%	59.43%	**31.77%**
MTL-BiLSTM	45.71%	60.92%	45.54%	36.13%	50.58%	43.83%	37.87%	85.37%	50.45%	25.36%
MTL-BiLSTM-CRF	**58.06%**	**72.92%**	**58.50%**	40.81%	**65.63%**	**57.14%**	43.38%	**90.57%**	**62.37%**	21.06%

Besides, the occurrences of the top three frequent implicit life events, Perception_active, Using, and Presence, are 59% of all implicit life events.

5.4 Life Event Quadruples Generation

With the outcomes of the first stage, this subsection presents our joint learning model on extracting factual life event quadruples from tweets for personal knowledge base construction. Table 8 shows the performances of life event quadruples generation given the tweets with life events.

The best model MTL-BiLSTM-CRF achieves an overall F-score of 58.06% when we verify the factual quadruples with annotated ground-truth, where the subject, the object and the time must be exactly matched the ground-truth. The joint learning approach improves overall performances and the CRF layer is effective in sequence labeling. In the results of implicit life event quadruples, the performances on subject are higher than those on the others because the subjects are often the Twitter user in implicit life events.

5.5 Frame Semantics Parsing

In this subsection, we report the average performances on semantic role labeling in Table 9.

Table 9. Performance of frame semantics parsing.

Models	All			Explicit	Implicit
	F1	P	R	F1	F1
LSTM	9.07%	12.98%	7.78%	8.07%	12.11%
LSTM-CRF	11.71%	17.65%	9.40%	10.26%	16.12%
BiLSTM	31.68%	38.10%	30.50%	33.80%	25.24%
BiLSTM-CRF	35.16%	44.66%	31.61%	37.45%	28.20%
MTL-LSTM	13.05%	17.24%	11.69%	11.96%	16.38%
MTL-LSTM-CRF	22.01%	28.74%	19.36%	20.62%	26.24%
MTL-BiLSTM	31.18%	33.95%	32.68%	32.78%	26.33%
MTL-BiLSTM-CRF	**41.60%**	**53.07%**	**37.03%**	**44.55%**	**32.64%**

The best model is MTL-BiLSTM-CRF, which achieves an F-score of 41.60% significantly outperforming all the baseline models with p<0.001 using the McNemar's test. Without the CRF layer as output layer, the performances of all models degrade. Moreover, we find that the joint learning approach helps improve the performances of both life event quadruples and frame semantics parsing subtasks.

5.6 Evaluation of the Pipelined System

Finally, we evaluate the end-to-end performance of our system in the pipelined workflow. That is, the implicit/explicit life events identified in the first stage are sent to the second stage to generate the KB facts.

However, the word spans extracted by our models may not exactly match the ones labeled by annotators. For example, the word 看 (see), which is extracted by our model, is equivalent to the word 看到 (see) annotated by human annotators. Table 10 shows an example of such a case. Actually, 深圳灣公園 (Shenzhen Bay Park), which is extracted by our model, is even more informative than the word 公園 (park) annotated by human annotators.

Table 10. An example of the object predicted by our model different from the answer labeled by annotator.

Tweet	在深圳灣公園裡看看書。(I read the book in the Shenzhen Bay Park.)
Our Model	深圳灣公園 (Shenzhen Bay Park)
Annotator	公園 (Park)

Therefore, we report an alternative F-score that regards the prediction is correct if the head word is matched with the ground-truth. Table 11 shows the performances of the pipelined system. We input the outcomes of the first stage into the MTL-BiLSTM-CRF model, which achieves the highest overall F-score in the second stage. The baseline model is the single task learning model where we select the best model in each task. Specifically, we select BiLSTM for both life event detection and implicit life event recognition and select BiLSTM-CRF for explicit life event recognition. As shown in Table 11, the problem of error propagation of the baseline model is more serious than that of the joint learning model. The best model MTL-BiLSTM-CRF achieves an F-score of 15.63%.

Table 11. Performances of the pipelined system.

Task / Models	First stage								Second stage								
	Life Event Detection		Explicit Life Event Recognition			Implicit Life Event Recognition			Subject Extraction	Object Extraction	Time Extraction	Frame Semantic Parsing			Life Event Quadruples Generation		
	A	F1	F1	P	R	F1	P	R	F1	F1	F1	F1	P	R	F1	P	R
Baseline	94.56%	92.12%	27.40%	26.92%	29.09%	51.07%	48.21%	58.50%	45.40%	13.04%	27.06%	22.60%	23.93%	21.63%	10.85%	12.49%	10.45%
MTL-LSTM	94.96%	92.94%	32.87%	37.26%	31.80%	58.51%	55.95%	65.21%	55.41%	17.72%	29.21%	25.80%	31.51%	20.66%	14.04%	15.09%	13.84%
MTL-LSTM-CRF	**95.72%**	**93.96%**	32.75%	37.53%	31.47%	62.84%	62.57%	64.25%	56.14%	18.40%	29.69%	26.40%	31.94%	21.20%	14.21%	15.30%	14.02%
MTL-BiLSTM	95.63%	93.91%	36.19%	38.63%	**36.61%**	63.09%	62.98%	64.35%	59.76%	20.22%	33.19%	28.17%	35.00%	22.66%	14.94%	15.83%	14.85%
MTL-BiLSTM-CRF	95.30%	93.47%	**37.13%**	**41.09%**	36.52%	**73.18%**	**70.19%**	**80.65%**	**63.94%**	**22.00%**	**35.00%**	**29.77%**	**37.42%**	**23.91%**	**15.63%**	**16.21%**	**15.97%**

6 DISCUSSIONS

Section 6.1 analyzes the performances of the model with different feature sets (i.e., metadata and BERT). Section 6.2 analyzes the performances on the character level and on the word level features. Section 6.3 and Section 6.4 analyze the user behavior of lifelogging with the results extracted by our system, showing potential directions of the text-based lifelog mining.

6.1 Performances of Features

In this subsection, we show the performances of our model with or without metadata and BERT. Table 12 shows the performances of the best models of each task. We find that the features of metadata and BERT are effective, especially BERT. The metadata is effective for implicit life event recognition.

Table 12. Performances of the models with different features on each subtask.

Subtask	Models	Features	F1
Life Event Detection	MTL-LSTM-CRF	all	93.96%
		w/o metadata	92.57%
		w/o BERT	80.76%
Explicit Life Event Recognition	MTL-BiLSTM-CRF	all	39.23%
		w/o metadata	34.88%
		w/o BERT	31.80%
Implicit Life Event Recognition	MTL-BiLSTM-CRF	all	80.99%
		w/o metadata	65.22%
		w/o BERT	74.40%
Frame Semantic Parsing	MTL-BiLSTM-CRF	all	58.06%
		w/o metadata	48.92%
		w/o BERT	42.97%
Life Event Quadruples Generation	MTL-BiLSTM-CRF	all	41.60%
		w/o metadata	30.38%
		w/o BERT	27.71%

6.2 Comparing Performances on Character Level or Word Level Features

In this paper, we report the performance of concatenating the character level and the word level as input. Table 13 show the performances on the character level and on the word level individually in the task of life event detection.

Table 13. Performance of concatenating character level or word level in life event detection.

Models		A	F1	P	R
LSTM	Both	94.55%	92.10%	93.92%	90.35%
	Word level	89.84%	84.97%	88.61%	81.61%
	Char level	82.14%	77.14%	70.17%	85.65%

The performance of the LSTM with the word level features as input is better than that with the character level features because the word level features contain more semantic information. However, the word level features may contain wrong Chinese word segmentation results due to the informal writing in tweets. Therefore, input with both level features improves the performance.

6.3 Explicit and Implicit Life Event Analysis

We list the top 20 frequent explicit life events in Fig. 3 and the top 20 frequent implicit life events in Fig. 4. Comparing the frequencies between the explicit and the implicit life events, users often express what they see and where they are in both explicit and implicit ways. However, users usually express the life events about where they *go* explicitly. In addition, when users want to express what they are doing by using something, they tend to give comments on the thing they are using. However, users might query what they used instead of the comments. This shows a semantic gap between the text-based log and the query.

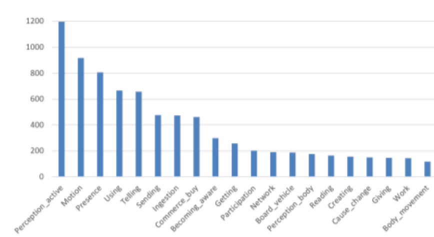

Figure 3: Top 20 frequent explicit life events

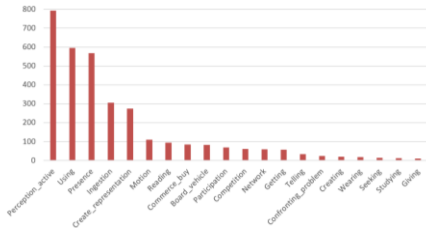

Figure 4: Top 20 frequent implicit life events

6.4 Relation between Personal Life Event and Time

In this subsection, we investigate the relation between life events and times. We show the frequencies of seven important life events, including Presence, Motion, Ingestion, Commerce_buy, Participation, Broad_vehicle, Work, and Network, of the 18 users at the time intervals of daily and hourly. Fig. 5 shows the frequencies of the life events in each day. The bar denotes frequency. The life events of Presence, Participation, and Ingestion happen on holiday more frequently, while the frequencies of Work and Network on Sunday are less than on the other days. This result reflects that people often do not work on Sunday and they might tend to go out with friends and go to restaurant rather than to surf on internet at home. Fig. 6 shows the frequencies of life events in each hour. We observe that the life event Ingestion frequently happens on at the 12 o’clock and the period of 18 o’clock to 23 o’clock. It might represent two timestamps are the time of lunch, dinner and midnight snack. Besides, people like to mention their work in the morning.

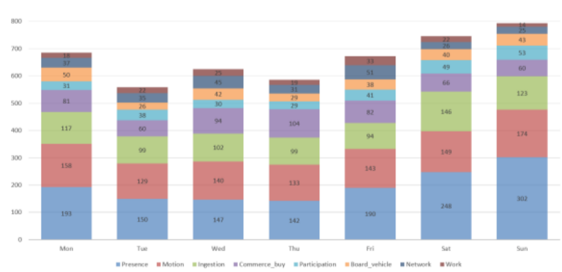

Figure 5: Frequencies of seven important life events in each day

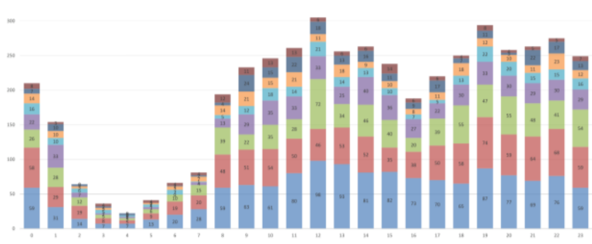

Figure 6: Frequencies of seven important life events in each hour

Furthermore, we investigate the number of tweets posted in each hour, which is shown in Fig. 7. The red line denotes the frequency of the tweet without life event, and the blue line denotes the frequency of the tweet with life event. We notice that the users use Twitter actively at 10 o’clock to 13 o’clock, and 22 o’clock to 24 o’clock. While people seldom post a tweet during the period of 1 o’clock to 6 o’clock.

Interestingly, the number of the tweets with life event and the number of the tweets without life event are almost equivalent in this time period. It might represent that users prefer to tell their life events if they post the tweet during the period between 1 o’clock and 7 o’clock.

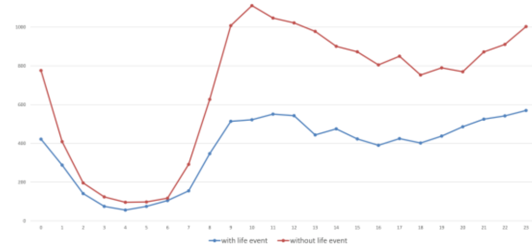

Figure 7: The number of tweets posted in each hour

7 CONCLUSIONS

Lifelogging attracts much attention in recent years. Different from previous work, this paper addresses the topic of personal knowledge base construction on text-based social media lifelogs. We propose a complete system that identifies life events, extracts event components, and generates KB facts. Both implicit and explicit cases of life events are considered. We represent the extracted life events in the form of the quadruple (subject, predicate, object, time), which is compatible with most modern knowledge bases.

In the two stages of our system, dedicated models are proposed based on sophisticated technology such as joint learning. Furthermore, to predict the potential action in an implicit life event, we investigate the influence of different input features. The results show that a combination of features of BERT and user metadata improves the performance, especially metadata. Besides, combining word level and character level features as input helps to learn a better representation on informal text that achieves a better performance on life event detection.

We do not only evaluate our model for each subtask individually, but also conduct an end-to-end experiment with the system in the pipelined workflow. Experimental results show the effectiveness of each model in our system, and confirm the quality of the generated KB facts. The KB constructed by our system is accessible to provide complementary information for a variety of applications such as memory recall and living assistance.

ACKNOWLEDGMENTS

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-106-2923-E-002-012-MY3, MOST-107-2634-F-002-011-, and MOST-108-2634-F-002-008-.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International the Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, 722–735.

[2] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250.

[3] Richard Caruana. 1993. Multitask learning: a knowledge based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*.

[4] Smitashree Choudhury and Harith Alani. 2015. Detecting Presence of Personal Events in Twitter Streams. In *Proceedings of the International Conference on Social Informatics*, 157-166.

[5] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jian Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In Proceedings of ACL, 167-176.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[7] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Mathias Lux, and Cathal Gurrin. 2018. Overview of ImageCLEFlifelog 2018: Daily Living Understanding and Lifelog Moment Retrieval. In: *CLEF 2018 Working Notes. CEUR Workshop Proceedings*, CEURWS.org <http://ceur-ws.org>, Avignon, France.

[8] Thomas Dickinson, Miriam Fernández, Lisa A. Thomas, Paul Mulholland, Pamela Briggs, and Harith Alani. 2016. Identifying Important Life Events from Twitter Using Semantic and Syntactic Patterns. In *Proceedings of the 15th International Conference WWW*, 143-150.

[9] Rowanne Fleck and Geraldine Fitzpatrick. 2009. Teachers' and tutors' social reflection around SenseCam images. In *International Journal of Human-Computer Studies*, 67(12): 1024-1036.

[10] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. 2006. Solving the problem of cascading errors: approximate Bayesian inference for linguistic annotation pipelines. In *Proceedings of EMNLP*, 618-626.

[11] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. 2016. Overview of ntcir lifelog task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, NTCIR-12. National Center of Sciences.

[12] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of ICASSP 2013*, 6645–6649.

[13] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6), 602-610.

[14] Eben Harrell. 2010. Remains of the day: can a new device help amnesia patients outsource memory? *Time Magazine*, 46-51.

[15] Richard Harper, Dave W. Randall, Nicola Smyth, Cara Evans, L. Heledd, Ronald E. Moore. 2008. The Past is a Different Place: They Do Things Differently There. In *Proceedings of the 7th ACM conference on Designing interactive systems*, 271-280.

[16] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Woodberry. 2006. SenseCam: A retrospective memory aid. In *Proceedings of International Conference on Ubiquitous Computing*, 177-193.

[17] Sen Hu, Lei Zou, and Xinbo Zhang. 2018. A State-transition Framework to Answer Complex Questions over Knowledge Base. In *Proceedings of EMNLP*, 2098-2108.

[18] Alan Jackoway, Hanan Samet, and Jagan Sankaranarayanan. 2012. Identification of live news events using twitter. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, 25-32.

[19] Martin Källström. 2013. Lifelogging camera: the narrative clip. Retrieved from http://getnarrative.com/

[20] Jacqueline Kerr, Simon J. Marshall, Suneeta Godbole, Jacqueline Chen, Amanda Legge, Aiden R. Doherty, Paul Richard Kelly, Melody Oliver, Hannah Badland, Charlie Foster. 2013. Using the SenseCam to Improve Classifications of Sedentary Behaviour in Free-Living Settings. In *American journal of preventive medicine*, 44(3): 290-296.

[21] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, 260–270.

[22] Jiwei Li and Claire Cardie. 2014, Timeline Generation: Tracking individuals on Twitter. In *Proceedings of the 23rd international conference on World wide web*, 643-652. DOI: https://doi.org/10.1145/2566486.2567969

[23] Chin-Ho Lin, Hen-Hsen Huang, and Hsin-Hsi Chen. 2018. Learning to Map Natural Language Statements into Knowledge Base Representations for Knowledge Base Construction". In *Proceedings of LREC*.

[24] John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[25] Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. 2014. Major Life Event Extraction from Twitter based on Congratulations/Condolences Speech Acts". In *Proceedings of the EMNLP*, 1997-2007.

[26] Davis Liang and Yan Shu. 2017. Deep Automated Multi-task Learning. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, 55-60.

[27] Xueliang Liu, Raphaël Troncy, and Benoit Huet. 2011. Using social media to identify events. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media*, 3-8.

[28] Takuya Maekawa. 2013. A sensor device for automatic food lifelogging that is embedded in home ceiling light: a preliminary investigation. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, 405-407.

[29] Rajesh Elara Mohan, Lee Hyowon, K. S. Jaichandar, and Carlos Acosta Antonio Calderon. 2012. LifeVision: Integrating heart rate sensing in lifelogging camera for accurate risk diagnosis for the elderly. In *Proceedings of the 6th International Conference on Rehabilitation Engineering & Assistive Technology*, 35.

[30] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014, The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 55-60.

[31] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL*, 300-309.

[32] Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 365-371.

[33] Mitsuo Nohara, Nobuhide Kotsuka, Masayuki Hashimoto, and Hiroki Horiuchi. 2010. A study on food-log application to a medical-care consult via telecommunications. In *Proceedings of the 16th International Conference on Virtual Systems and Multimedia*, 88-91.

[34] Gillian O'Loughlin, Sarah Jane Cullen, Adrian McGoldrick, Siobhán O'Connor, Richard J Blain, Shane O'Malley, and Giles D. Warrington. 2013, Using a Wearable Camera to Increase the Accuracy of Dietary Analysis. In *American journal of preventive medicine*, 44(3): 290-296.

[35] Juan José Soler, Fernando Cuartero, and Manuel Roblizo. 2012, Twitter as a tool for predicting elections results. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, 1194-1200.

[36] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of ACL*, 231-235.

[37] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, 851-860.

[38] Krishna Chaitanya Sanagavarapu, Alakananda Vempala, and Eduardo Blanco. 2017. Determining Whether and When People Participate in the Events They Tweet About. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. ACL*, 641-646.

[39] Tsung-Han Yang, Hen-Hsen Huang, An-Zi Yen, and Hsin-Hsi Chen. 2018, Transfer of Frames from English FrameNet to Construct Chinese FrameNet: A Bilingual Corpus-Based Approach. In *Proceedings of LREC*.

[40] Xuchen Yao, and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*, 956-966.