BUILDING A TIME SERIES ACTION NETWORK FOR EARTHQUAKE DISASTER

The-Minh Nguyen, Takahiro Kawamura, Yasuyuki Tahara and Akihiko Ohsuga
The University of Electro-Communications' Graduate School of Information Systems
1-5-1, Chofugaoka, Chofu-shi, Tokyo, Japan

Keywords: Evacuation-rescue, Twitter, Action network, Action-based collaborative filtering.

Abstract: Since there is 87% of chance of an approximately 8.0-magnitude earthquake occurring in the Tokai region of Japan within the next 30 years; we are trying to help computers to recommend suitable action patterns for the victims if this massive earthquake happens. For example, the computer will recommend "*what should do to go to a safe place*", "*how to come back home*", etc. To realize this goal, it is necessary to have *a collective intelligence of action patterns, which relate to the earthquake*. It is also important to let the computers make a recommendation *in time*, especially in this kind of emergency situation. This means these action patterns should to be collected in real-time. Additionally, to help the computers understand the meaning of these action patterns, we should build the collective intelligence based on web ontology language (OWL). However, the manual construction of the collective intelligence will take a large cost, and it is difficult in the emergency situation. Therefore, in this paper, we first design a time series action network. We then introduce a novel approach, which can automatically collects the action patterns from Twitter for the action network in real-time. Finally, we propose a novel action-based collaborative filtering, which predicts missing activity data, to complement this action network.

1 INTRODUCTION

The ability of computers to recommend suitable action patterns based on users' behaviors is now an important issue in context-aware computing (Matsuo et al., 2007), ubiquitous computing (Poslad, 2009), and can be applied to assist people in disaster areas. When the massive Tohoku earthquake and Fukushima nuclear disaster occurred in March 2011, many people felt panic, and did not know "what should do", "where was the available evacuation center", "how to come back home", etc. The Japanese government said that there is 87% of chance of an approximately 8.0-magnitude earthquake occurring in the Tokai region within the next 30 years (Nikkei, 2011). In this case, temporary homeless such as people unable to return home is expected to reach to an amount of 6.5 million (Nakabayashi, 2006). Therefore, we need an approach to help the computers to provide suitable action patterns for the disaster victims.

To help the computers provide suitable action patterns, it is necessary to have *a collective intelligence of action patterns*. Additionally, we need to understand *how to collect activity data*, *how to express or define each activity*. During the massive Tohoku earthquake, while landlines and mobile phones got stuck, Twitter were used to exchange information. Not only individuals but also the Fire and Disaster Management Agency, Universities and local governments used Twitter to provide information about evacuation, traffic, damaged area, etc. On 11 March, the number of tweets from Japan dramatically increased to about 33 million (Biglobe, 2011), 1.8 times higher than the average figure. Therefore, we can say that Twitter is becoming the *sensor* of the real world. In other words, we can collect activity data which relate to the earthquake from Twitter.

In this paper, we define an activity by five attributes namely *actor*, *act*, *object*, *time* and *location*. And an *action* consists of a combination of *act* with *object*. For example, in the sentence "Tanaka is now taking refuge at Akihabara", *actor*, *act*, *time* and *location* are "Tanaka", "take refuge", "now", "Akihabara" respectively. Since, the number of tweets is large, it is not practical to manually collect these attributes. Additionally, sentences retrieved from Twitter which are more complex than other text media, are often structurally varying, syntactically incorrect, and have

Nguyen T., Kawamura T., Tahara Y. and Ohsuga A..
BUILDING A TIME SERIES ACTION NETWORK FOR EARTHQUAKE DISASTER.
DOI: 10.5220/0003741701000108
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 100-108
ISBN: 978-989-8425-95-9

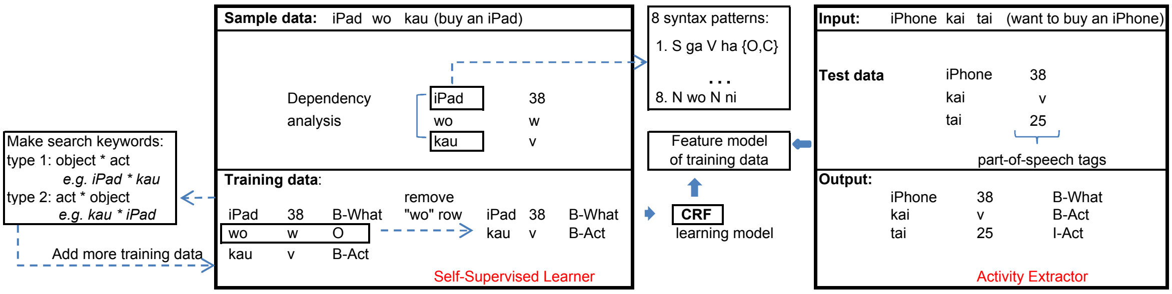

Figure 1: By using deep linguistic parser and 8 syntax patterns, the Learner automatically makes training data. Based on these training data, the Extractor automatically extracts activity attributes in each sentence retrieved from Twitter.

many user-defined new words. Thus, there are lots of challenges to extract activities in these sentences (Nguyen et al., 2011). Previous works (Fukazawa and Ota, 2009; Nilanjan et al., 2009) which are based on the co-occurrence of *act* and *object*, do not depend on the retrieved sentences syntax. However, this approach *can not extract infrequent activities*, and have to prepare a list of *act* and *object* before extracting. There are some other works (Perkowitz et al., 2004; Kawamura et al., 2009; Kurashima et al., 2009) have tried to extract human activities from web and weblogs. These works have some limitations, such as high setup costs because of requiring ontology for each domain (Kawamura et al., 2009). Due to the difficulty of creating suitable patterns, these works (Perkowitz et al., 2004; Kurashima et al., 2009) are limited on the types of sentences that can be handled, and insufficiently consider interdependency among attributes.

In emergency situations, it is important to let the computers make a recommendation *in time*. This means that the activity attributes should to be collected, and to be represented in real-time. However, there is a high possibility that activity data on Twitter are discontinuous data. Thus, we need to solve the problem of missing activity data. Additionally, to help the computer understand the meaning of the data, we should build the collective intelligence based on OWL. In this paper, we first design a time series action network to represent human activities. An then, we propose a novel approach, which can automatically collects the activity attributes from Twitter for the action network in real-time. Finally, we propose a novel action-based collaborative filtering, which predicts missing activity data, to complement the action network. The main contributions of our work are summarized as follows:

- It has successfully designed the time series action network based on OWL.
- It can automatically make semantic data for the action network.
- It can predict missing activity data to complement the action network.
- By using the action network, the computers can recommend suitable action patterns for the disaster victims.

The remainder of this paper is organized as follows. Section 2 explains how our approach automatically extract human activity from Twitter. In section 3, we design the time series action network, and then explain how to make the semantic data. Section 4 explains how to predict missing activity data. Section 5 reports our experimental results, and explains how to apply the action network. Section 6 considers related work. Section 7 consists of conclusions and some discussions of future work.

2 MINING HUMAN ACTIVITY FROM TWITTER

Our key ideas for extracting activity attributes in each sentence retrieved from Twitter, are summarized as follows:

- We represent each activity attribute by its label. Thus, activity extraction can be treated as a sequence labeling problem.
- We deploy *self-supervised learning*, and use CRF (linear-chain conditional random field) as a learning model. We firstly make training data of parsable activity sentences. Secondly, we use Google Blog search to add more training data. Finally, we use these training data to deal with more complex sentences.
- Since sentences retrieved from Twitter often contain noise data, we remove these noise data before testing. Additionally, to avoid error when testing, we convert complex sentences to simpler sentences by simplifying noun phrases and verb phrases.

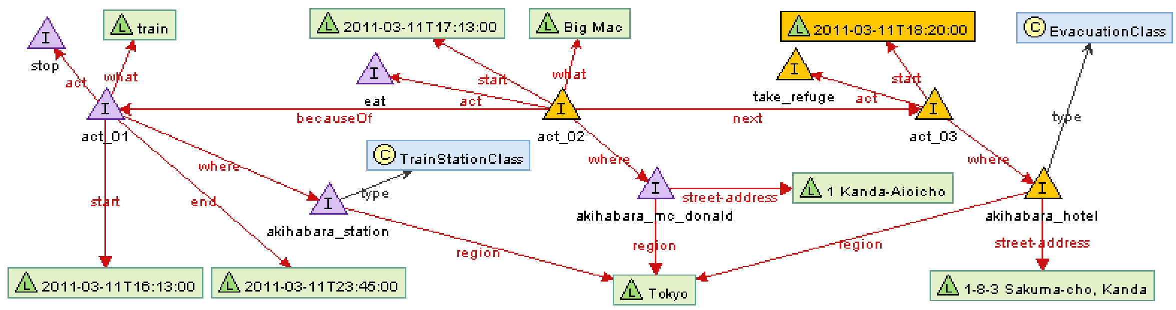

Figure 2: An excerpt from time series action network.

- We consider not only time stamp of tweets, but also time expression (now, this evening, etc) to decide time of activities in these tweets.

As shown in Figure 1, our proposed architecture consists of two modules: *Self-Supervised Learner* and *Activity Extractor*. Firstly, the Learner uses 8 basic Japanese syntax patterns to select parsable activity sentences. And then, it uses deep linguistic parser to extract activity attributes. Secondly, it uses extracted *act* and *object* to create search keywords for Google Blog search API. Based on these keywords, it collects new activity sentences that contains trustworthy attributes. Thirdly, the Learner combines extracted results to automatically make training data. Finally, it uses CRF and a feature template file to make the feature model of these training data. The Extractor does *not* deploy deep linguistic parser, it bases on the feature model to predict attributes in each sentence retrieved from Twitter.

3 BUILDING TIME SERIES ACTION NETWORK

3.1 Definition of Time Series Action Network

Time series action network (TiSAN) is a collective intelligence of human activities while earthquake occurs. As shown in Figure 2, TiSAN is expressed as a directed graph whose nodes are concepts of activity attributes, and whose edges are relations between these concepts.

3.2 Designing Time Series Action Network

It is important to help the computers understand the meaning of data, thus we design TiSAN based on OWL (Web Ontology Language). Since N3 (Notation 3) (W3C, 2006) is a compact and readable alternative to RDF's XML syntax, we use N3 to describe TiSAN.

```
@prefix   geo:    <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix   tl:     <http://purl.org/NET/c4dm/timeline.owl#> .
@prefix   vcard:  <http://www.w3.org/2006/vcard/ns#> .
```

Figure 3: TiSAN inherits Geo, Time line and vCards.

To easily link to external resource, TiSAN inherits Geo (Geo, 2003), Time line (Raimond and Abdallah, 2007), and vCards (Halpin et al., 2010) ontologies (Figure 3). Geo (Geo, 2003) is used for representing latitude and longitude of a location. Time line (Raimond and Abdallah, 2007) is used for representing time. And, vCards (Halpin et al., 2010) is used for representing an address of a location.

```
### Definition of activity class
:ActionClass          a                  owl:Class ;
                      rdfs:subClassOf    owl:Thing .

### Definition of act, where, and what classes
:ActClass             a                  owl:Class ;
                      rdfs:subClassOf    owl:Thing .
:WhereClass           a                  owl:Class ;
                      rdfs:subClassOf    owl:Thing .
:WhatClass            a                  owl:Class ;
                      rdfs:subClassOf    owl:Thing .

### Sub-class of WhereClass
:ShopClass            a owl:Class ;
                      rdfs:subClassOf    :WhereClass .
:RestaurantClass      a                  owl:Class ;
                      rdfs:subClassOf    :WhereClass .
:TrainStationClass    a                  owl:Class ;
                      rdf:subClassOf     :WhereClass .
:EvacuationClass      a                  owl:Class ;
                      rdf:subClassOf     :WhereClass .
```

Figure 4: Classes of TiSAN.

Figure 4 shows classes of TiSAN. ActionClass, ActClass, WhereClass, and WhatClass are classes of activity, act, location, object respectively. Shop, restaurant, train station, and evacuation center are important locations, so we create classes for them.

As shown in Figure 5, TiSAN has five properties: act, what, where, next, and becauseOf, which corre-

```
### Definition of properties
:act          a              owl:ObjectProperty ;
              rdfs:label     "act" ;
              rdfs:domain    :ActionClass ;
              rdfs:range     :ActClass .

:what         a              owl:ObjectProperty ;
              rdfs:label     "what" ;
              rdfs:domain    :ActionClass ;
              rdfs:range     :WhatClass .

:where        a              owl:ObjectProperty ;
              rdfs:label     "where" ;
              rdfs:domain    :ActionClass ;
              rdfs:range     :WhereClass .

### Definition of relations
:next         a              owl:ObjectProperty ;
              rdfs:label     "next" ;
              rdfs:domain    :ActionClass ;
              rdfs:range     :ActionClass .

:becauseOf    a              owl:ObjectProperty ;
              rdfs:label     "becauseOf" ;
              rdfs:domain    :ActionClass ;
              rdfs:range     :ActionClass .
```

Figure 5: Properties of TiSAN.

spond to activity attributes, and relations between activities.

```
:stop                a                     :ActClass ;
                     rdfs:label            "stop"@en .

:akihabara_station   a                     :TrainStationClass ;
                     rdfs:label            "Akihabara station"@en ;
                     vcard:region          "Tokyo"@en ;
                     vcard:locality        "Chiyoda-ku"@en ;
                     vcard:street-address  "1-17-6 Sotokanda"@en ;
                     geo:lat               35.69858 ;
                     geo:long              139.773108 .

:act_01              a                     :ActionClass ;
                     :act                  :stop ;
                     :what                 "train"@en ;
                     :where                :akihabara_station ;
                     tl:start   2011-03-11T16:13:00^^xsd:dateTime ;
                     tl:end     2011-03-11T23:45:00^^xsd:dateTime .
```

Figure 6: An example of TiSAN data.

Based on the above classes, properties, and inherited ontologies, we can describe data of TiSAN. For example, Figure 6 represents the activity in the sentence “The train has stopped at Akihabara Station at 16:13:00”.

3.3 Creating Semantic Data

Figure 7 explains the method of creating semantic data for TiSAN. Firstly, we use #jishin (#earthquake) tag which relates to earthquake to collect activity sentences from Twitter. Secondly, we use our proposed method in Section 2 to extract activity attributes, and relationships between activities. Finally, we convert the extracted data to RDF/N3 to make semantic data for TiSAN.

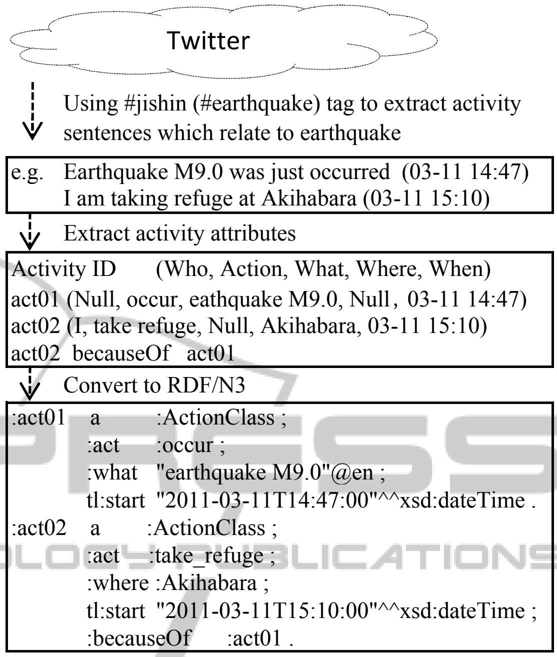

Figure 7: Method of creating semantic data for TiSAN.

4 COMPLEMENT TIME SERIES ACTION NETWORK

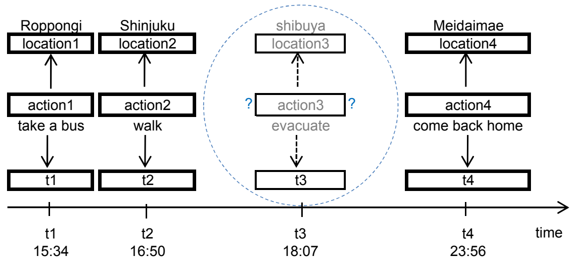

Figure 8: User did not post his activity on Twitter at 18:07.

It is important to let the computers make a recommendation in time, especially in emergency situations such as earthquake disaster. To do this, the activity mining process should be done in real-time. In other words, the action network need to contain real-time action patterns. However, there is a high possibility that the activity data on Twitter are not complete. For example, Figure 8 shows that the active user did not post his activity on Twitter at 18:07. Therefore, as shown in Figure 9, the action network lacked the activity at 18:07.

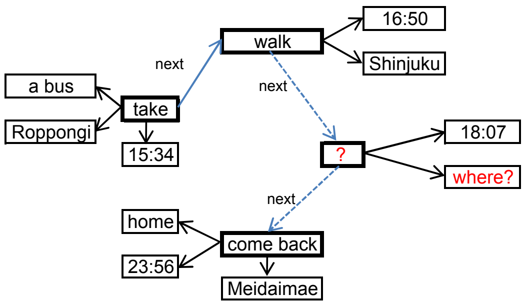

Figure 9: The network lacked the activity at 18:07.

From the above reason, to complement the action network, we need an approach to predict missing activities. As shown in Figure 10, given the active user u_a and time t as input, this approach need to know:

1. What did u_a do at time t?
2. Where was u_a at time t?

We will explain how to predict the action and the location of the active user at the time t below.

Input	u_a	active user
	t	time t
Output	$action_t$	What did u_a do at t ?
	$location_t$	Where is u_a at t ?

Figure 10: Predict missing activity of u_a at t .

4.1 Approach for Predicting Missing Activity

Let $Can_{act} = \{act_1, act_2, .., act_i, ...\}$ is the set of candidate actions of the active user u_a at time t. Detecting the action of u_a at time t can be considered as choosing the action in Can_{act}, which has the most highest possibility of occurrence. Therefore, we need to calculate possibility of u_a did act_i at time t ($P_{u_a \to act_i}$). Based on the following ideas, we calculate $P_{u_a \to act_i}$.

- It is high possibility that similar users have similar actions.
- In emergency situations, users' actions strongly depend on theirs time and locations. For example, users could not take a train while train systems are stopped. And, it is high possibility that users will take a bus if they are in bus station.

Thus, to calculate $P_{u_a \to act_i}$ we need to calculate similarity between two users ($S(u_j, u_a)$), and possibility of candidate action act_i ($P(act_i)$).

4.2 Similarity between Two Users

Based on the following ideas, we calculate similarity between two users in emergency situations.

- It is high possibility that as same as user u_a, similar users also did before action ($Did(a_{before})$) and after action ($Did(a_{after})$) of the candidate action act_iD
- If users had the same goal (e.g. wanted to evacuate in Shinjuku), then they had same action patterns ($SameTarget(a_t, l_t)$).
- It is high possibility that user did the same actions if they were in the same location ($SameLocation(l)$).

Therefore, the similarity between user u_j and user u_a will be calculated as Equation 1.

$$\begin{aligned} S(u_j, u_a) = & \beta Did\left(\{a_{before}, l_{before}\}, \{a_{after}, l_{after}\}\right) \\ & + \gamma SameTarget(a_t, l_t) \\ & + (1 - \beta - \gamma) SameLocation(l) \end{aligned} \tag{1}$$

Where:

- Parameters β, γ satisfy $0 \leqq \beta, \gamma, \beta + \gamma \leqq 1$. These parameters depend on each particular problem.
- If u_j did action act_i in location l, then $Did(act_i, l) = 1$, otherwise $Did(act_i, l) = 0$.
- If u_j and user u_a has the same goal (want to do action a_t in target location l_t), then $SameTarget(a_t, l_t) = 1$, otherwise $SameTarget(a_t, l_t) = 0$.
- If u_j and user u_a were in the same location l at the time t, then $SameLocation(l) = 1$, otherwise $SameLocation(l) = 0$.

4.3 Possibility of Action

In real-world, an action depend on location, time and its before-after actions. Therefore, possibility of the candidate action act_i at the time t can be calculated as Equation 2.

$$\begin{aligned} P(act_i) = & \rho_a \left\{F(a_{before} \to act_i) + F(act_i \to a_{after})\right\} \\ & + \rho_t F(act_i, t) + (1 - \rho_a - \rho_t) F(act_i, l) \end{aligned} \tag{2}$$

Where:

- Parameters ρ_a, ρ_t satisfy $0 \leqq \rho_a, \rho_t, \rho_a + \rho_t \leqq 1$.
- $F(a_{before} \to act_i)$ is frequency of $a_{before} \to act_i$ (transition from a_{before} to act_i).
- $F(act_i \to a_{after})$ is frequency of $act_i \to a_{after}$ (transition from act_i to a_{after}).

- $F(act_i, t)$ is frequency of act_i at time t.
- $F(act_i, l)$ is frequency of act_i in location l.

4.4 Predicting Missing Action

Combination of Equation 1 and Equation 2, we can calculate $P_{u_a \to act_i}$ as Equation 3.

$$P_{u_a \to act_i} = \alpha \left(\frac{\sum_{j=1,L} \omega(u_j, acti) * S(u_j, u_a)}{L} \right) + (1-\alpha) P(act_i) \quad (3)$$

Where:

- L is number of all users similar to u_a.
- $\omega(u_j, acti)$ is a weighting factor. If user u_j did $acti$, then $\omega(u_j, acti) = 1$, otherwise $\omega(u_j, acti) = 0$.
- Parameters α satisfies $0 \leqq \alpha \leqq 1$. It depends on each particular problem.

5 EVALUATION

In this section, we first evaluate our activity extraction approach. Secondly, we use SPARQL (SPARQL Protocol and RDF Query Language) to evaluate our time series action network. Then, we evaluate our proposed approach which complements missing activities. Finally, we discuss the usefulness of the action network.

We had collected 416,463 tweets which related to the massive Tohoku earthquake. And then, to create data-set for the evaluations, we selected tweets which were posted by users in Tokyo from 2011/03/11 to 2011/03/12.

5.1 Activity Extraction

Generative learning and discriminative learning are the two main machine-learning approaches. While generative learning is well-known by hidden Markov model (HMM), discriminative learning is famous for maximum entropy Markov model (MEMM), support vector machine (SVM), and conditional random field (CRF). Previous works (Sha and Pereira, 2003; McCallum and Li, 2003; Kudo et al., 2004) have shown that CRF outperforms both MEMM and HMM on sequence labeling task. Therefore, we focused on comparing between CRF and SVM. Basically, SVM is a binary classifier, thus we must extend SVM to multi-class classifier (Multi-SVM) in order to extract activity attributes (actor, act, object, time, and location). The evaluation results are shown in Table 1. These results have shown that: with every activity attribute, CRF outperforms Multi-SVM in both precision and recall. In other words, we can see that CRF is a good choice for our task, activity extraction.

Table 2 shows the comparison results of our approach with baseline method, and Nguyen et al. (Nguyen et al., 2011). Based on the results, we can see that the baseline has high precision but low recall. The reason is that sentences retrieved from twitter are often diversified, complex, syntactically wrong. Nguyen et al. also used self-supervised learning and CRF, but it could not handle complex sentences.

5.2 Time Series Action Network

```
SELECT   DISTINCT   ?location_name   ?street_address   ?end_time
WHERE {?action   :act   :open .
     ?action   tl:start   ?start_time .
     ?action   tl:end   ?end_time .
     ?action   :where   ?location .
     ?location   rdf:type   :EvacuationClass .
     ?location   rdfs:label   ?location_name .
     ?location   vcard:locality   "Chiyoda-ku"@en .
     ?location   vcard:street-address   ?street_address .

     FILTER(?start_time <= "2011-03-11T17:00:00"^^xsd:dateTime &&
          ?end_time >= "2011-03-11T17:00:00"^^xsd:dateTime &&
          lang(?street_address) = "en" &&
          lang(?location_name) = "en"
     )
}
```

Figure 11: Look up an available Evaluation center.

location_name	"Akihabara Washington Hotel"@en
street_addresss	"1-8-3 Sakuma-cho, Kanda"@en
start_time	2011-03-11T16:00:00
end_time	2011-03-12T09:00:00

Figure 12: Opening evacuation center.

We used SPARQL to make RDF queries to evaluate our time series action network. For example, Figure 11 shows the query that look up an available Evaluation center based on the current time (2011-03-11T17:00:00), and the current location (Chiyoda-ku) of the victims. The result of this query is shown in Figure 12. Therefore, we can say that our action network which is working properly with RDF queries.

5.3 Missing Activity Prediction

To evaluate our proposed approach, we first created correct action data of 3,900 Twitter users in Tokyo, after the massive earthquake occurred. Secondly, we repeated 10 times of the following experiment.

Table 1: Comparison of CRF with Multi-SVM in precision, recall, and F-measure.

@	Learning Model	Activity	Actor	Act	Object	Time	Location
Precision	Multi-SVM	66.15%	77.22%	90.02%	74.05%	73.51%	75.20%
	CRF	**73.21%**	**82.25%**	**97.11%**	**81.23%**	**80.04%**	**82.11%**
Recall	Multi-SVM	60.03%	72.03%	85.31%	70.02%	71.78%	72.15%
	CRF	**66.54%**	**80.11%**	**93.18%**	**76.57%**	**79.75%**	**81.02%**
F-measure	Multi-SVM	62.94%	74.53%	87.60%	71.98%	72.63%	73.64%
	CRF	**69.72%**	**81.17%**	**95.10%**	**78.83%**	**79.89%**	**81.56%**

Table 2: Comparison of our approach with baseline, and Nguyen et al, (2011).

@	Method	Activity	Actor	Act	Object	Time	Location
Precision	Baseline	81.17%	86.32%	98.13%	84.14%	87.96%	88.25%
	Nguyen et al.	57.89%	72.79%	82.98%	67.01%	76.40%	80.20%
	Our approach	**73.21%**	**82.25%**	**97.11%**	**81.23%**	**80.04%**	**82.11%**
Recall	Baseline	23.86%	26.38%	28.87%	24.77%	26.20%	26.02%
	Nguyen et al.	51.13%	69.13%	90.23%	62.11%	73.51%	77.67%
	Our approach	**66.54%**	**80.11%**	**93.18%**	**76.57%**	**79.75%**	**81.02%**
F-measure	Baseline	36.88%	40.41%	44.61%	38.27%	40.37%	40.19%
	Nguyen et al.	54.30%	70.91%	86.45%	64.47%	74.93%	78.91%
	Our approach	**69.72%**	**81.17%**	**95.10%**	**78.83%**	**79.89%**	**81.56%**

1. Randomly select 39 users as the active users.
2. Randomly delete activity data of these active users.
3. Let the active users' names and time of deleted activities as input data, using our proposed approach in section 4 to determine whether the deleted activity data is reproduced or not.

The average results are shown in Table 3. From these results, we can say that our approach can reproduce 69.23% of missing actions, 76.92% of missing locations, and 43.59% of missing activities (both of action and location).

Table 3: Recall of Deleted Activity Data.

Action	Location	Both of Action and Location
69.23%	76.92%	43.59%

5.4 Application of Action Network

Today, many companies and research centers are trying to build user activity model, and to predict users' behaviors in real-world. For example, NTT Docomo (NTTDocomo, 2009) is trying to predict users' destinations, and then provide shop information around these destinations. KDDI research center (KDDI, 2009) is trying to collect users' daily activity data on their mobiles, and then provide suitable information for these users. Addition to these application, our work is applicable to many fields and business models, such as context-aware computing (Matsuo et al., 2007), ubiquitous computing (Poslad, 2009), behavioral targeting, rescue-evacuation.

If data on Twitter is real-time data, then we can say that TiSAN reflects real-world activities in real-time. By using SPARQL (SPARQL Protocol and RDF Query Language), computers can understand situations of trains, evacuation centers, shops..etc. Therefore, we can use TiSAN to find a safe place for disaster victims.

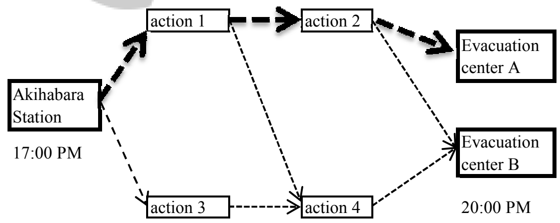

Figure 13: Using TiSAN to recommend suitable action patterns.

The computers also can recommend "what should to do" for a user, based on action patterns of the others in TiSAN. For example, as shown in Figure 13, in the three hours past (17:00 PM to 20:00 PM), most people from Akihabara station did {action 1, action 2} to reach to "evacuation center A". Therefore, the computer can recommend {action 1, action 2} and "evacuation center A" for the current user at Akihabara station.

6 RELATED WORK

There are three fields related to our research: human activity, concept network, and collaborative filtering

Table 4: Comparison of our action-based pproach with traditional CF.

Point of View		Traditional CF	Our approach (action-based CF)
Reasech goal		Recommend suitable items	Predict missing action data
Target		Items in EC sites	Users' activities
Complexity		1 variable (item)	4 variables (act, object, location, time)
Dependence	Location	NO	YES
	Transistion	Weak	Strong
	Goal concept	NO	YES (e.g. want to evaluate)
Continuity		NO	YES (need to consider executive time)

(CF). Below, we will discuss the previous researches of each field.

6.1 Human Activity

Since weblogs and Twitter are becoming sensor of real-world, previous works (Kawamura et al., 2009; Kurashima et al., 2009; Fukazawa and Ota, 2009; Nilanjan et al., 2009) have tried to extract users' activities from weblogs and Twitter. Kawamura et al. (2009) requires a product ontology and an action ontology for each domain. So, the precision of this approach depends on these ontologies. Kurashima et al. (2009) uses a deep linguistic parser to extract action and object. But, Banko and Etzioni (2008) indicated that it is not practical to deploy deep linguistic parser, because of the diversity and the size of web corpus. This approach can only handle sentences whose structure is "NP wo/ni VP". Additionally, because this approach gets date information from date of weblogs, so it is highly probable that extracted time might not be what activity sentences describe about.

Fukazawa et al. (2009) uses the pattern "Domain wo/ni VP" as the search keyword to acquire domains (e.g. movie, music, meal,.. etc) and verb phrases (e.g. watch, listen, eat,..etc), by using a search engine. And then, it selects (Domain,VP) pairs which satisfy *Score(Domain, VP)* $\geq 10^{-5}$, and treats these pairs as (object, act) pairs. Because of using the specified pattern, this approach has a low recall, and can not extract infrequent activities.

$$Score(Domain, VP) = \frac{Hits(Domain\ \ AND\ \ VP)}{Hits(Domain)Hits(VP)} \tag{4}$$

The goal of Nilanjan et al. (2009) (Nilanjan et al., 2009) is to capture a users' real-time interests in activities from Twitter. Firstly, it prepares a list of "interest-indicative words" (e.g. game, music, food,...etc), a list of "act keywords" (e.g. go, play, listening, eating,..etc), and a list of "temporal keywords" (e.g. tonight, tomorrow, weekend,..etc). Secondly, it calculates co-occurrences of (interest-indicative words, act keywords), and (interest-indicative words, temporal keywords). Finally, it selects high co-occurrences as users' interests. For example, if the word "movie" occurs along with "go" and "tomorrow" with high co-occurrences, it means the user is interested in going to a movie tomorrow. This approach has some problems such as inability of extracting actors, and infrequent activities. But it is highly probable that infrequent activities contain valuable information. Additionally, this approach can not get exact time of activities, so we can said that it can not extract users' interests in real-time.

In activity recognition, there are some works (Perkowitz et al., 2004; Vincent et al., 2009; Shiaokai et al., 2007) have used the Web to mine human activity models, and to label activity data retrieved from sensors. However, these works focused on common sense models, such as: cleaning indoor, laundry, making coffee.

6.2 Concept Network

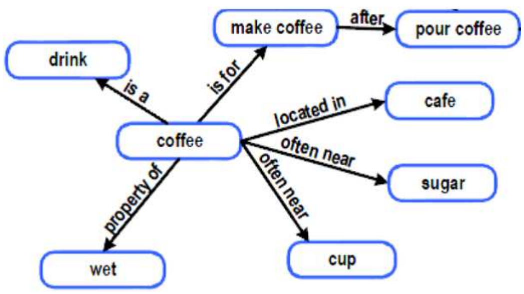

Figure 14: An excerpt from ConceptNet.

The main research of concept network is ConcepNet (MIT Media Lab) (Hugo and Push, 2004). ConceptNet is a semantic network of commonsense knowledge, based on the information in OpenMind commonsense corpus (OMCS) (MIT, 2011). As shown in Figure 14, ConceptNet is expressed as a directed graph whose nodes are concepts, and whose edges are relations between these concepts.

ConceptNet prepared a list of patterns in advance, and then it uses these patterns to extract concepts, and the relations between these concept. For example, given "A pen is made of plastic" as an input sentence, it uses "NP is made of NP" to get two concepts (a pen, plastic), and the relation (is made of) between these concepts. However, it is not practical to deploy this method for extract human activity from Twitter, because sentences retrieved from twitter are often diversified, complex, syntactically wrong. Additionally, ConcepNet is not designed based on OWL.

6.3 Collaborative Filtering

While traditional CF is trying to recommend suitable products on internet for users, our work is try to predict missing action data in real-world. Different with products, user action strongly depend location, time, and before-after actions. Additionally, we need to consider executive time of each action. Table 4 shows comparisons of our action-based approach with the traditional CF.

(Ma et al., 2007; Koren, 2009) are the start-of-art approaches of the traditional CF. (Ma et al., 2007) proposed a combination item-based CF and user-based CF, but it did not consider time and location. (Koren, 2009) considered time, but did not consider location.

7 CONCLUSIONS

In this paper, we have designed an time series action network. Additionally, we proposed a novel approach to automatically collect action patterns from Twitter for the action network. We also explained how to use this semantic network to assist disaster victims.

We are improving the architecture to handle more complex sentences retrieved from Twitter. We also improving the approach of predicting missing activity data to complement the action network.

REFERENCES

Biglobe (2011). http://tr.twipple.jp/info/bunseki/20110427 .html.

Fukazawa, Y. and Ota, J. (2009). Learning user's real world activity model from the web. In *IEICE SIG Notes*.

Geo (2003). http://www.w3.org/2003/01/geo/wgs84_pos.

Halpin, H., Iannella, R., Suda, B., and Walsh, N. (2010). http://www.w3.org/2006/vcard/ns.

Hugo, L. and Push, S. (2004). a pratical commonsense reasoning toolkit. *BT Technology Journal*, 22(4).

Kawamura, T., Nguyen, T.-M., and Ohsuga, A. (2009). Building of human activity correlation map from weblogs. In *Proc. ICSOFT*.

KDDI (2009). Mobile phone based Lifelog.

Koren, Y. (2009). Collaborative filtering with temporal dynamics. In *Proc. KDD*.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying conditional random fields to japanese morphologiaical analysis. In *Proc. EMNLP*, pages 230–237.

Kurashima, T., Fujimura, K., and Okuda, H. (2009). Discovering association rules on experiences from large-scale weblogs entries. In *Proc. ECIR. LNCS vol 5478. Springer*.

Ma, H., King, I., and R-Lyu, M. (2007). Effective missing data prediction for collaborative filtering. In *Proc. SIGIR*.

Matsuo, Y., Okazaki, N., Izumi, K., Nakamura, Y., Nishimura, T., and Hasida, K. (2007). Inferring long-term user properties based on users' location history. In *Proc. IJCAI*, pages 2159–2165.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.

MIT (2011). http://openmind.media.mit.edu/.

Nakabayashi, I. (2006). Development of urban disaster prevention systems in japan - from the mid-1980s. *Journal of Disaster Research*, 1.

Nguyen, T.-M., Kawamura, T., Nakagawa, H., Tahara, Y., and Ohsuga, A. (2011). Automatic extraction and evaluation of human activity using conditional random fields and self-supervised learning. *Transactions of the Japanese Society for Artificial Intelligence*, 26:166–178.

Nikkei (2011). http://e.nikkei.com/e/fr/tnks/Nni20110705D 05HH763.htm.

Nilanjan, B., Dipanjan, C., Koustuv, D., Anupam, J., Sumit, M., Seema, N., Angshu, R., and Sameer, M. (2009). User interests in social media sites: An exploration with micro-blogs. In *Proc. CIKM*.

NTTDocomo (2009). My Life Assist Service.

Perkowitz, M., Philipose, M., and Donald, K. F. J. (2004). Mining models of human activities from the web. In *Proc. WWW*.

Poslad, S. (2009). *Ubiquitous Computing Smart Devices, Environments and Interactions*. Wiley, ISBN: 978-0-470-03560-3.

Raimond, Y. and Abdallah, S. (2007). http://purl.org/NET/c4dm/timeline.owl.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proc. HLTNAACL*, pages 213–220.

Shiaokai, W., William, P., Ana-Maria, P., Tanzeem, C., and Matthai, P. (2007). Common sense based joint training of human activity recognizers. In *Proc. IJCAI*, pages 2237–2242.

Vincent, W., Derek, H., and Qiang, Y. (2009). Cross-domain activity recognition. In *Proc. Ubicomp*, pages 2159–2165.

W3C (2006). http://www.w3.org/DesignIssues/Notation3.