A System for Collecting Tweets Using Event-Based Structuring of Web Contents

Norifumi Hirata[1], Shun Shiramatsu[2], Tadachika Ozono[3], Toramatsu Shintani[4]

Dept. of Computer Science and Engineering, Graduate School of Engineering Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, Japan

[1]nori@toralab.org; [2]siramatu@toralab.org; [3]ozono@toralab.org; [4]tora@toralab.org

***Abstract*- We introduce a system that aims to improve understanding of an event that appears on news and microblogs. In order to identify an event that a user is interested in, a user gives a system a URL of browsing news page. A system has two kinds of agents. One is a news agent to extract relevant news articles and the other is a microblog agent to extract related microblogs. A news agent extracts news articles that represent the same event as a browsing news article. A microblog agent extracts related microblogs from news articles. A microblog agent uses news URL references and a similarity between a news article and a microblog. As the result of experiment, tweets that are collected using our proposed method were 300 times more than tweets using a primitive method. We confirmed that a user could obtain more related tweets using our proposed system.**

Keywords- News Article; Microblog; Event

I. INTRODUCTION

The vast amount of contents is posted to news sites and microblogging services every day. It is important to collect, classify and use the opinions for developing social or public services. To classify the opinions, a system needs to collect enough opinions. This paper introduces a system to collect opinions that have a same target, e.g., a news event. In particular, our proposed system focuses on Web news articles and tweets. In our previous work [1], we have proposed a method to classify news articles based on events. Almost all news articles and some tweets represent events. Almost all news articles represent recent occurrences or incidents. News articles are important information source on the Web. News articles affect some tweets. Because some twitter users know recent occurrences or incidents by reading news articles, our proposed system collects tweets that are affected by news articles.

2.5 billion tweets are posted in a day[1]. It is difficult to collect all tweets related to a specific event. If we use a retrieval using hash tags or keywords, we need to use appropriate queries. News articles and microblog posts have links to other contents. Even if contents of posts are related, the contents do not always have links. News articles tend to have links to other news articles on a same news site. If a user creates a microblog post by a post button on a news site, the microblog post has a link to a news article. However, collected tweets are limited when we retrieve using a URL of a news article. The target of this work is to collect enough tweets and to keep a precision.

To collect many related tweets, we use an event-based structure that is generated by news articles and tweets. In this paper, a related tweet means a tweet related to a particular event. Real time is a feature of Twitter. To keep the real time, a system must not take time to calculate a relation between an event and a tweet. Our proposed system creates a search query based on a similarity. Our proposed system can collect many related tweets and to keep a precision to use the query. It is difficult to collect all opinions on the Web. However, microblogging services often have search APIs. We can collect related opinions using existing search APIs and retrieval systems.

II. RELATIONSHIP BETWEEN NEW ARTICLES AND TWEETS

A. Collecting Tweets

It is important to collect and propose tweets related to news articles. For example, news readers of Yahoo news[2] can post opinions and emotional to news articles. News articles have share buttons[3]. Tweet button provides the search result of a news URL. Users can know and share other users' opinions. However, the collectable tweets depend on a published time of news article, an attention degree, the number of users of a news site and settings of share buttons.

Users can retrieve tweets that have a URL of a browsing news article when users click a tweet button. Users can know other users' reactions and opinions. When retrieved tweets are few, users might think that more related tweets exist because the result is retrieved using a URL of a browsing news article. It is important to collect sufficient tweets to share other users' opinions when the tweets that have a URL of a browsing news article are few.

[1] Web2.0 Summit 2011
[2] http://dailynews.yahoo.co.jp/
[3] tweet button (http://twitter.com/goodies/tweetbutton),
Like button http://developers.facebook.com/docs/reference/lugins/ like), +1 button (http://www.google.com/intl/ja/+1/button)

DOI:10.5963/IJCSAI0302003

Idea Box and Data Box in Open Government Lab [2] are Japanese government system to collect opinions. Citizens can post ideas and opinions. e-Participation platform Open-Opinion (O2) [3] collects related tweets and news articles for discussion among citizens. O2 uses Linked Open Data for a transparency. These systems aim to e-Participation. Our proposed system can apply to systems for e-Participation because our proposed system can collect tweets related to events that represent local concerns.

B. Structuring of News Articles and Tweets

To collect tweets related to a particular event, we use an event-based structure of news articles and tweets. In the Topic Detection and Tracking (TDT) [4, 5] project, an event is a unique occurrence at a point in time. In this paper, an event is a target of a news article or a tweet. To identify a user's interesting event, our proposed system uses a browsing news article. A user inputs a URL of a browsing news article. Our proposed system collects related tweets using an identified event.

Fig. 1 shows structuring of news articles and tweets based on an event E. Tweets are classified in 4 classes. When a user browses a news article P, the system can retrieve tweets using a URL of P. However, the collected tweets using only one URL have a high precision and a low recall. If the system obtains a news article Q that targets a same event E, the system can retrieve more related tweets. We refer to tweets retrieved using a URL of P as Tweet Type A. We refer to tweets retrieved using a URL of Q as Tweet Type C. The proposed system collect Tweet Types B and D. Tweet Types B and D do not have a URL of news articles. To collect Tweet Types B and D, the system needs to calculate a similarity between a news article and a tweet.

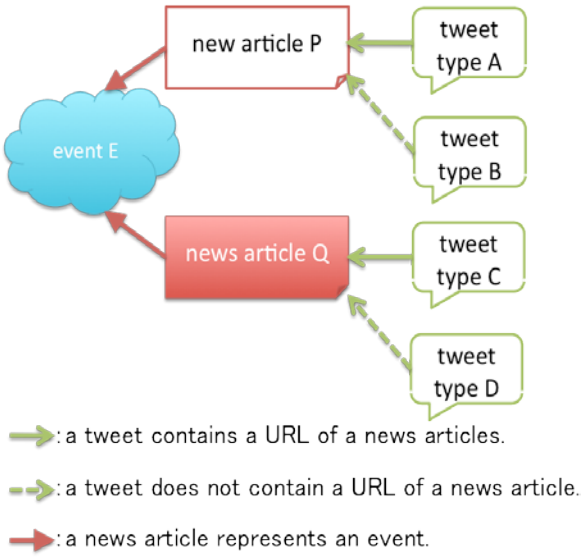

Fig. 1 Event-based structuring of news articles and tweets

Other proposed method [7] uses query expansion using information of a Twitter user. Wikipedia and WordNet are used to collect Web contents [8, 9]. Our proposed system is similar to methods that use related news articles for query expansion. Wikipedia and WordNet are comparatively static information. The system deals with new events. The system uses news articles because they have novel information about news events.

C. Collecting Related Tweets Using URL of a News Article

Twitter Web Analytics[4] and bitly[5] provide an URL access analysis on Twitter. We can analyse traffic from URL links on Twitter.

We analysed a relation between the number of tweets that have a news URL and tweets time from a news published time. The target news articles are 83 news articles and they are published on the top page of Yahoo! Japan[6] in August 3rd, 2012. We collected tweets that have URL of the 83 news articles using Twitter Search API[7] for 1 week.

Fig. 2 shows the result of relation between the number of tweets that have a news URL and tweets time from a news published time. The result means the number of tweets stops increasing in about 24 hours in this case. The average number of tweets is about 170. However, more related tweets exist because 2.5 billion tweets are posted in a day.

[4] https://dev.twitter.com/blog/introducing-twitter-web-analytics
[5] http://bitly.com/
[6] http://www.yahoo.co.jp/
[7] https://dev.twitter.com/docs/using-search

DOI:10.5963/IJCSAI0302003

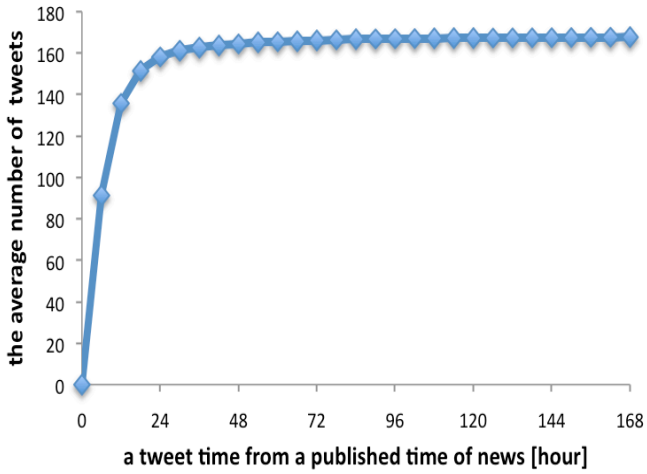

Fig. 2 Tweeting time and the average number of tweets that have news URLs

D. *Collecting Tweets Using Hash Tags*

It is difficult to find a hash tag to represent an event. Twitter users generate hash tags after an occurrence of an event. Each small event does not always have a hash tag. Twitter users tend to generate many hash tags when unexpected events occur. Fig. 3 shows hash tags about typhoon No.15 in Aichi in September 2011. Many hash tags are generated. A hash tag #nagoya is used with other hash tags. A combination of hash tags is important to collect tweets using hash tags.

#nagoya (city name)	#名古屋市 (Nagoya city)	#saigai (disaster)
#災害情報 (information of disaster)	#suigai (flood disaster)	#hinan (escape)
#庄内川 (river name)	#kasugai (city name)	#高蔵寺 (station name)
#天白川 (river name)	#天白川 (river name)	#台風 (typhoon)
#愛知豪雨 (heavy rain in Aichi)	#台風 15 号 (typhoon no. 15)	

Fig. 3 Examples of Hash tags about typhoon no.15 in Aichi Prefecture

Summarization [10, 11] and burst detection[8] are related to microblog analysis. To detect a burst in a specific genre, a detection system needs to qualify target tweets. Other works qualify target tweets using hash tags or keywords.

III. METHOD FOR COLLECTING RELATED TWEETS USING NEWS ARTICLES

A. *Outline of Similarity between a News Article and a Tweet*

Our proposed system collects related tweets based on the structure of Fig. 1. The system extracts related news articles from a browsing news article. Fig. 4 shows the outline of similarity between a news article and a tweet.

Fig. 4 shows a similarity of a case that an input news article is a_{input} and related news articles are a_1 and a_2. $sim_a(a_{input}, a_i)$ means a similarity between input news article a_{input} and related news article a_i. $sim_m(a_i, m)$ means a similarity between a news article a_i and a tweet m. A similarity $sim_{input}(a_{input}, m)$ between sim_{input} and m is a max value of $sim_a * sim_m$. If a similarity sim_{input} uses a sum or an average of $sim_a * sim_m$, sim_{input} depends on the number of related news articles. We believe that a similarity should not depend on an attention degree or an event scale.

B. *Method for Collecting Related News Article*

After a user inputs a news article, the system extracts related news articles that represent a same event. The system uses a title and body text of news articles to calculate a similarity. The system uses Japanese language morphological analysis software called MeCab [12] for evaluating parts of speech.

Equation (1) shows a similarity between news article a_1 and a_2.

[8] buzztter(http://buzztter.com/), kizasi.jp (http://kizasi.jp/)

DOI:10.5963/IJCSAI0302003

$$sim_a(a_1, a_2) = \begin{cases} \cos(\vec{a}_1, \vec{a}_2), if \left| t(a_1) - t(a_2) \right| < \theta_{ta} \\ 0, \qquad otherwise \end{cases} \tag{1}$$

$$\cos(\vec{a}_1, \vec{a}_2) = \frac{\vec{a}_1 \cdot \vec{a}_2}{\left|\vec{a}_1\right\|\vec{a}_2|} \tag{2}$$

t_a means a published time of news article a. θ_{ta} means a time restriction. $\cos(\vec{a}_1, \vec{a}_2)$ is a cosine similarity between a_1 and a_2 using tf-idf. $\vec{a}_1$ is a tf-idf vector of a news article. The system extracts news articles when the similarity $sim_a(a_{input}, a_i)$ exceeds θ_a.

C. Method for Collecting Related Tweets

When a tweet refers to URL of a news article, the post is related to the news articles. When a tweet does not refer to the URL, the system evaluates a similarity between a news article and a tweet. A similarity sim_m between an article a and a tweet m is calculated as follows:

$$sim_m(a, m) = \begin{cases} 1, & if \ \left| terms(m) \cap \{url(a)\} \right| > 0 \\ f(a, m), & else \ if \ \left| t(m) - t(a_{input}) \right| < \theta_{tm} \\ 0, & otherwise \end{cases} \tag{3}$$

$$f(a, m) = \frac{\sum_{w_{sub} \in features(a) \cap teams(m)} tf \cdot idf_a(w_{sub})}{\sum_{w \in features(a)} tf \cdot idf_a(w)} \tag{4}$$

When a tweet m refers to the URL of news article a, a similarity $sim_m(a, m)$ is 1. $terms(m)$ is words in m and $url(a)$ is a URL of a. When a time distance between a tweet and a news article is within a threshold θ_{tm}, a similarity $sim_m(a, m)$ is $f(a,m)$. $t(m)$ is a posted time of m. Time distance is one of important elements because a time and a tweet of a content are closely [13]. In the other case, a similarity $sim_m(a, m)$ is 0. $f(a,m)$ is a rate of news specific words $features(a)$ in tweet text. $features(a)$ is a set of words w that $tfidf_a(w)$ values exceed a threshold. Parts of speech of w in $features(a)$ must be a noun or a verb.

A similarity between an event and a tweet is the maximum value as shown below:

$$sim_{input}(a_{input}, m) = \quad \max_{a \in A} \{ sim_a(a_{input}, a) \cdot sim_m(a, m) \} \tag{5}$$

Where A is a set of news articles related to event e that is extracted from a_{input}. If a similarity $sim_{input}(a_{input}, m)$ exceeds a threshold θ_m, the system assumes that the tweet m relates to a_{input}. When the tweet m is extracted from a news article that has a low similarity sim_a, a condition of a similarity sim_{input} to exceed a threshold θ_m is hard. Other paper [14] has proposed a method that uses tf-idf in tweets. However, the maximum length of a tweet is 140 characters. Our proposed system evaluates a relation using appearances of terms in a tweet. A recent study [15] showed that 85% tweets do not contain duplicative terms.

D. Collecting Tweets Using Retrieval

Our proposed system cannot evaluate all tweets on the Web. As at December 2012, we can collect about 500 thousands Japanese tweets in a day using Twitter Streaming API[9]. The system must not take time to evaluate a relation between an event and a tweet.

The system can retrieve tweets using search queries from news articles. The condition that $sim_m(e,m)$ exceeds θ_m is follows:

$$S(a_{input}, a) = \Big\{ Sub \mid Sub \subseteq features(a) \\ sim_a(a_{input}, a) \cdot \frac{\sum_{w_{sub} \in Sub} tf \cdot idf_a(w_{sub})}{\sum_{w \in features(a)} tf \cdot idf_a(w)} > \theta_m \Big\} \tag{6}$$

Where Sub is a subset of $features(a)$. $tfidf_a(w)$ is a tf-idf value in a news article a. If a tweet contains all elements of Sub in Equation (6), the tweet exceeds threshold θ_m. The system can collect tweets that exceed θ_m using retrieval. A method to make a query is follow:

1. join each word w_{sub} in Sub by "and"

[9] https://dev.twitter.com/docs/streaming-api

DOI:10.5963/IJCSAI0302003

2. join each text that made in step (1) by "or"
3. clear duplications

A format of a search query is a disjunctive normal form. For example, When *Sub* is *{{t1, t2},{t1, t2, t3}}*, a search query in Step (2) is "*(t1 and t2) or (t1 and t2 and t3)*". In Step (3), the search query is "*t1 and t2*" because "*t1 and t2*" and "*t1 and t2 and t3*" make a same result. Finally, the system calculates $S(a_{input},a)$ for all related news articles *a*, and join and clear duplications.

In summary, an algorithm of related tweets extraction is shown in Fig. 5. In $GetQuery(news_{input}, news, \theta_m)$ at Line 8, the system calculates a query text according to Equation (6). *GetQuery* does not need tweet data. In $Retrieval(Post, Query, \theta_m)$ at Line 11, the system retrieves related tweets from Twitter Search API or collected tweets. We assume that our proposed system covers Japanese language. If we use our proposed system for the other languages, we need to confirm effects.

```
input: news_input is a input article
       News is a set of related news articles
       Posts is a set of tweets
output: a set of related tweets

01:procedure RetrievePosts(news_input, News, Posts)
02:begin
03:   M ← {}; // a set of related tweets
04:   Query ← {}; // a set of related queries
05:   foreach news in News do
06:     if sim_a(news_input, news) > θ_a then
07:       // GetQuery: get S(news_input, news)
08:       Query ← Query ∪ GetQuery(news_input, news, θ_m);
09:     end do
10:   end if
11:   M ← Retrieve(Posts, Query, θ_tm);
12:   return M;
13:end.
```

Fig. 5 An algorithm for collecting related tweets using retrieval

IV. SYSTEM STRUCTURE AND INTERFACE

A. System Structure

Fig. 6 shows a structure of our proposed system. First, a user inputs a browsing news article to the system. In the system, a news agent extracts related news articles from collected news articles. From the related articles, a microblog agent makes a search query and retrieves tweets related to an event. Finally, the system presents a user related tweets.

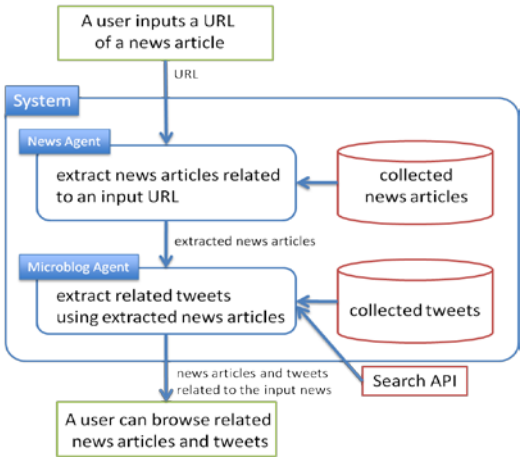

Fig. 6 A system structure for collecting related tweets

The system collects news articles from news sites. The system collects tweets using Twitter Streaming API. Tweets that are collected using Twitter Streaming API are a part of all tweets. The system collects tweets using Twitter Search API when an input news article is closely. We can retrieve tweets until about 1 week ago using Twitter Search API.

B. Interface of the System

Fig. 7 shows an interface of our proposed system for Web browsers. The left side is related tweets. The right shows related news articles. The top is a title of an input news article.

DOI:10.5963/IJCSAI0302003

Fig. 7 An interface for related tweets and news articles.

The proposed system can collect many related tweets. Users cannot browse all related tweets at a glance. User can select a priority because tweets that a user needs depend on each user.

Twitter has many features [16] such as a length of a tweet, posting time and a twitter user's profile. In our proposed system, users can select a tweet time and a similarity between an input news article as priorities.

V. EXPERIMENT

A. Experiment Environment

We collected related tweets from news articles. We used news articles on asahi.com[10], MSN Sankei News[11], YOMIURI ONLINE[12]. A time restriction θ_{tm} about tweets is 24 hours because the number of tweets stops increasing in about 24 hours in Section II.C. A time restriction θ_{ta} about news articles is also 24 hours. *idf* are calculated from news articles within a time restriction θ_{ta}.

We evaluated two types of a precision as follows:

$$pre_1(A_{input}) = \frac{\sum_{a \in A_{input}} |related(a) \cap collected(a)|}{\sum_{a \in A_{input}} |collected(a)|} \tag{7}$$

$$pre_2(A_{input}) = \frac{1}{A_{input}} \sum_{a \in A_{input}} \frac{|related(a) \cap collected(a)|}{|collected(a)|} \tag{8}$$

Where *related(a)* is a set of related tweets. *collected(a)* is a set of collected tweets from news article *a*. Equation (7) evaluates collected tweets equally. *collected(a)* depends on input news article *a*. Our proposed system is used when a user browses a news article. Equation (8) evaluates browsing news articles equally.

B. Preliminary Experiment

We compared collected tweets using feature words from only titles and feature words from titles and first sentences of body texts. Leads of news text tend to appear specific words [17]. Other researcher's method [9] uses titles and first sentences of body texts. We collected related tweets using search query. Collected tweets are Tweets Types B and D in Fig. 1. A set of input news article A_{input} is published between February 2nd 2012 and February 4th 2012. A_{input}, has 10 news articles. Two methods are as follows:

type B+D (title): use titles

[10] http://www.asahi.com/
[11] http://sankei.jp.msn.com/
[12] http://www.yomiuri.co.jp/

DOI:10.5963/IJCSAI0302003

type B+D (title+text): use titles and first sentences

We evaluated whether collected tweets are related to input news articles by hands. Table I shows sample evaluations. Sample tweets in Table I are collected using the news article about the Osaka Metropolis plan and Sakai city. Tweet m_B does not relate Osaka Metropolis plan and Sakai city. It is difficult to evaluate whether tweet m_C relates to the input news article. *related(a)* contains tweets such as m_B.

TABLE I SAMPLE TWEETS THAT ARE COLLECTED USING THE NEWS ARTICLE ABOUT THE OSAKA METROPOLIS PLAN AND SAKAI CITY

tweet	a part of tweet text	a relation between a tweet and a news article
m_A	Sakai city is not positive to join the Osaka Metropolis plan...	related tweet
m_B	...need to fossil fuel substitute. ``In Osaka, signatures exceed a required number...	unrelated tweet
m_C	...talked about a tax payment, the pension issue, the Osaka Metropolis plan and Kasumigaseki...	difficult to evaluate

Table II shows the results of collecting related tweets. The average number of related news articles is 2.2 articles for collecting related tweets.

TABLE II PRECISIONS AND THE AVERAGE NUMBER OF COLLECTED TWEETS

method	$pre_1(A_{input1})$ [%]	$pre_2(A_{input1})$ [%]	the average number of collected tweets
type B+D (title)	89.2	90.1	13.9
type B+D (title+text)	86.4	87.7	14.7

The collected tweets are about a set of input news articles A_{input1}.

The difference of the results between type B+D(title) and type B+D(title+text) was small. If news titles have a few specific words, Method Type B+D (title+text) can collect more than Method Type B+D (title). Method Type B+D (title) could obtain the similar result because a number of news articles were used.

C. Result of Collecting Related Tweets

We evaluated the number of collected tweets and precisions. Collected tweets were classified Tweets Types A, B, C and D. We used Twitter Search API when the system collected Tweets Type A and Type C because there were some cases that collected tweets were zero.

A set of input news article A_{input2} was published between December 1st 2012 and December 2nd 2012. A_{input2} had 5 news articles. Each news article had more than one related news article. Each news article had a few Type A tweets. The average number of tweets that had URLs of input news articles was 11.4. The threshold θ_a was 0.5 and the θ_a was 0.4.

We evaluated precisions of the collected tweets by three persons. $PRE_1(A_{input2})$ and $PRE_2(A_{input2})$ are the average of $pre_1(A_{input2})$ and $pre_2(A_{input2})$. Table III shows the average number of collected tweets and precisions. Type A+C means a sum of the Tweet Type A and Type C. Type B+D is a sum of the Tweet Type B and Type D.

TABLE III THE COLLECTED TWEETS ABOUT A SET OF INPUT NEWS ARTICLES A_{INPUT2}

tweet type	$PRE_1(A_{input2})$ [%]	$PRE_2(A_{input2})$ [%]	the average number of collected tweets
type A	100.0	100.0	11.4
type A+C	100.0	100.0	153.8
type B	96.6	96.4	27.6
type B+D	90.1	83.8	38.2

Each news article has a few tweets that have news URLs.

D. Discussion

The number of collected tweets using a single news article was less than using a number of news articles. The number of collected tweets using URLs of news articles was less than using search queries. The results show that many related tweets that do not have URLs of news articles exist. If the number of tweets using Streaming API is 1% of all tweets, Type B+D tweets might exist 300 times (=100*38.2*0.901/11.4) compared to Type A tweets. If tweets that have a news URL are few, the system could collect 300 times tweets.

DOI:10.5963/IJCSAI0302003

Type A+C tweets are about 13.5 times more than Type A tweets. Type B+D tweets are about 1.38 times more than Type B tweets. The effect of using Type D tweets seems small. However, the Type D tweets are collected using a number of news articles. Therefore Type D tweets can keep precisions.

The collected tweets using news URLs are 100.00%. We evaluate that Type A tweets are related because Type A tweets have URLs of input news. We evaluate that Type C tweets are related because extracted news articles are related to input news articles. The precisions $PRE_1(A_{input2})$ and $PRE_2(A_{input2})$ of Type B+D tweets are less than the precisions of Type B tweets. To keep precisions, it is necessary to collect Type D tweets. Collecting tweets are failed when a search query that is made of a title of an input news article is not appropriate. Our proposed system can avoid a critical failure to use a number of news articles.

E. Effect of Time Restriction

The number of tweets stops increasing in about 24 hours in Section II.C. A published time of a news article and a tweeting time about the article have a closely relation. We confirmed effects of a time restriction θ_{tm} that is a parameter to extract tweets. An experiment environment is same as the preliminary experiment in Section V.B.

Fig. 8 shows precision, recall and f-measure when a time restriction θ_{tm} is changed. F-measure is a weighted harmonic mean of precision and recall. Recalls in Fig. 8 is based on the number of tweets in 168 hours because it is hard to extract all related tweets from all tweets. Therefore the recall in 168 hours is 100%. F-measure in 72 hours is the highest value in this experiment. An appropriate time restriction depends on an event. A time restriction should be long when an event is continual.

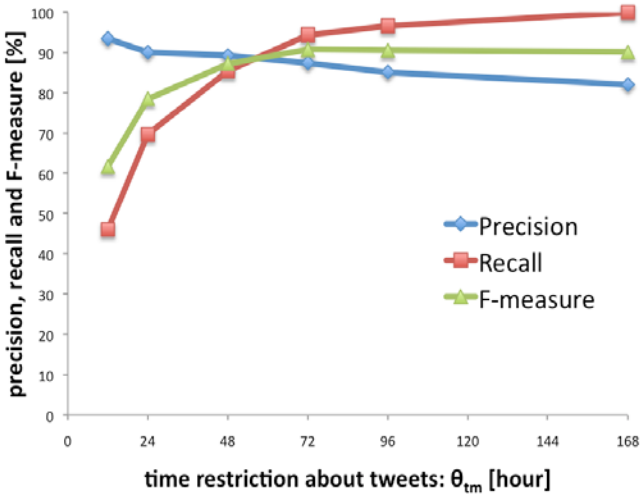

Fig. 8 Effects of the time restriction θ_{tm} for precisions, recalls and f-measures

VI. CONCLUSION

We proposed a system for collecting tweets related to a particular event using event-based structuring of Web contents. Our proposed system could collect more related tweets using news articles. The experimental results showed that the system could collect related tweets 300 times as many as tweets that have a news URL. Precisions were about 84% and 90% even when the system collected type D tweets. Related tweets can be retrieved using search queries based on our proposed similarity between an input news article and a tweet. It is important to collect tweets not to take times.

Our other works [18] have been tackled problems about disambiguation and contexts. Our proposed system will be able to provide users more priorities for collecting related tweets. The feature work is how to present related tweets to users.

ACKNOWLEDGMENT

This work was supported in part by SCOPE (Strategic Information and Communications R&D Promotion Programme) from Japan's Ministry of Internal Affairs and Communications.

REFERENCES

[1] N. Hirata, S. Shiramatsu, T. Ozono, and T. Shintani, “Generating an Event Arrangement for Understanding News Articles on the Web”, Lecture Notes in Computer Science, Part II, Vol. 6097, Springer, pp. 525-534, (IEA/AIE 2010), June 2010.

[2] Japanese Ministry of Economy, Trade and Industry, “Open Government Lab”, http://openlabs.go.jp/.

[3] S. Shiramatsu, R. M. E. Swezey, H. Sano, N. Hirata, T. Ozono, and T. Shintani, “Structuring Japanese Regional Information Gathered from the Web as Linked Open Data for Use in Concern Assessment”, The 4th International Conference on eParticipation (ePart 2012), 2012.

[4] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, “Topic Detection and Tracking Pilot Study Final Report”, The DARPA

DOI:10.5963/IJCSAI0302003

broadcast news transcription and understanding workshop}, pp. 194-218, 1998.

[5] D. Trieschnigg, and W. Kraaij "TNO Hierarchical topic detection report at TDT 2004", Topic Detection and Tracking 2004 Workshop, 2004.

[6] N. Hirata, Hiroyuki Sano, R. M. E. Swezey, S. Shiramatsu, T. Ozono, and T. Shintani, "A Web Agent Based on Exploratory Event Mining in Social Media", the third IIAI International Conference on e-Services and Knowledge Management (IIAI ESKM 2012), Sep. 2012.

[7] H. C. Lau,Y. R. Li, and D. Tjondronegoro, "Microblog Retrieval Using Topical Features and Query Expansion", The Twentieth Text REtrieval Conference, 2011.

[8] X. Chen, L. Li, G. Xu, Z. Yang, and M. Kitsuregawa, "Recommending Related Microblogs: A Comparison Between Topic and WordNet Based Approaches", The 26th AAAI Conference on Artificial Intelligence, pp. 2417-2418, 2012.

[9] Y. Sato, D. Yokomoto, K. Makita, T. Utsuro, and T. Fukuhara, "Collecting Blog Posts related to Topics in a News Article", DEIM Forum 2011 A6-3, 2011.

[10] B. Sharifi, M. A. Hutton, and J. Kalita, "Summarizing microblogs automatically", Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 685-688, 2010.

[11] H. Takamura, H. Yokono, and M. Okumura, "Summarizing a document stream", The 33rd European conference on Advances in information retrieval, pp.177-188, 2011.

[12] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://mecab.sourceforge.net.

[13] K. Lerman, and R. Ghosh, "Information Contagion, An Empirical Study of the Spread of News on Digg and Twitter Social Networks", The 4th International AAAI Conference on Weblogs and Social Media}, pp.90-97, 2010.

[14] A. Jackoway, H. Samet, and J. Sankaranarayanan, "Identification of live news events using Twitter", The 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks, pp.25-32, 2011.

[15] N. Naveed, T. Gottron, J. Kunegis, and C. A. Alhadi, "Bad News Travel Fast: A Content-based Analysis of Interestingness on Twitter", ACM WebSci'11, pp.1-7, 2011.

[16] C. Castillo, M. Mendoza, and B. Poblete "Information credibility on twitter", The 20th international conference on World wide web, pp. 675-684, 2011.

[17] Kyodo News, "Handbook for Editors & Writers 12th edition", 2010.

[18] R. M. E. Swezey, S. Shiramatsu, T. Ozono, and T. Shintani, "An Improvement for Naive Bayes Text Classification Applied to Online Imbalanced Crowdsourced Corpora", In Modern Advances in Intelligent Systems and Tools, Studies in Computational Intelligence, Vol. 431, pp. 147-152, Springer, 2012.

Norifumi Hirata is currently a PhD candidate at Nagoya Institute of Technology, Japan. He received his MS and BS degrees in computer science from Nagoya Institute of Technology of Nagoya City, Japan, in 2010 and 2008. He is currently (2013) a student at Nagoya Institute of Technology, Japan. His research interests include document classification, system-user interaction and text mining.

Shun Shiramatsu received his PhD degree in information science from Kyoto University, Japan, in 2009 and his MS degree in information science from Tokyo University of Science in 2003. He is currently (2013) an Assistant Professor of Computer Science. His research interests include discussion support and conversation modelling.

Tadachika Ozono received his MS and PhD degrees in computer science from Nagoya Institute of Technology of Nagoya City, Japan, in 1998 and 2000. He is currently an Associate Professor of Computer Science at Nagoya Institute of Technology of Nagoya City, Japan. Currently (2013) his main research interest is Web intelligence.

Toramatsu Shintani received his MS degree in industrial engineering and his PhD degree in computer science from Tokyo University of Science in 1982 and 1993, respectively. He was a research staff member at Fujitsu Limited from 1982 to 1993. He is currently (2013) a Professor of Computer Science at Nagoya Institute of Technology of Nagoya City, Japan. His current research interests include decision support systems and Web intelligence.

DOI:10.5963/IJCSAI0302003