Predicting the Lifespan and Retweet Times of Tweets Based on Multiple Feature Analysis

Yongjin Bae, Pum-Mo Ryu, and Hyunki Kim

In social network services, such as Facebook, Google+, Twitter, and certain postings attract more people than others. In this paper, we propose a novel method for predicting the lifespan and retweet times of tweets, the latter being a proxy for measuring the popularity of a tweet. We extract information from retweet graphs, such as posting times; and social, local, and content features, so as to construct prediction knowledge bases. Tweets with a similar topic, retweet pattern, and properties are sequentially extracted from the knowledge base and then used to make a prediction. To evaluate the performance of our model, we collected tweets on Twitter from June 2012 to October 2012. We compared our model with conventional models according to the prediction goal. For the lifespan prediction of a tweet, our model can reduce the time tolerance of a tweet lifespan by about four hours, compared with conventional models. In terms of prediction of the retweet times, our model achieved a significantly outstanding precision of about 50%, which is much higher than two of the conventional models showing a precision of around 30% and 20%, respectively.

Keywords: Tweet, lifespan, retweet, popularity, prediction, social network.

Manuscript received July 10, 2013; revised Nov. 8, 2013; accepted Dec. 30, 2013.

This work was supported by the IT R&D program of MSIP/KEIT (10044577, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services).

Yongjin Bae (phone: +82 10 2399 1036, yongjin@etri.re.kr) is with the SW·Content Research Laboratory, ETRI, Daejeon and also is a Master's student in the Department of Computer Software and Engineering, University of Science and Technology, Daejeon, Rep. of Korea.

Pum-Mo Ryu (pmryu@etri.re.kr) and Hyunki Kim (hkkim@etri.re.kr) are with the SW·Content Research Laboratory, ETRI, Daejeon, Rep. of Korea.

I. Introduction

Social network services (SNSs), such as Twitter, Facebook, and Google+, are a relatively new phenomenon in the Web 2.0 of user-generated contents. One of the most popular SNSs is Twitter. When a user publishes a tweet, which should be no longer than 140 characters, followers can subscribe to the user's postings in Twitter. Twitter's popularity and pervasiveness of information seem to have been further increased thanks to the proliferation of mobile devices.

Predicting the popularity of a tweet has emerged as a major concern in a variety of fields. Based on this information, companies want to promote products through online advertisements [1]. Public organizations can also take advantage of the tweets predicted to be popular to confront future social phenomena. To meet the social needs of users we should predict the popularity of tweets. The popularity of a tweet can be defined as follows [2]:

(a) How many times will it be retweeted?

(b) How long will it remain popular?

Case (a) is based on the number of retweets from other users, and (b) takes in to consideration the lifespan of a tweet. Figure 1 shows the different retweet patterns in Twitter used in the experiment. Tweets 1, 2, and 3 are about a disaster, politics, and daily life, respectively. We can see the properties of the tweets when comparing the three graph patterns. Tweets 1 and 2 were retweeted about 300 times. However, in terms of lifespan, they developed differently over time. Tweet 1 was retweeted 300 times within a very short time period. However, it did not receive any more attention after one hour.

On the contrary, we observed tweet 2 steadily propagate after 24 hours. Based on the comparison of tweets 1 and 2, we assume that the lifespan of a tweet is an important factor of its

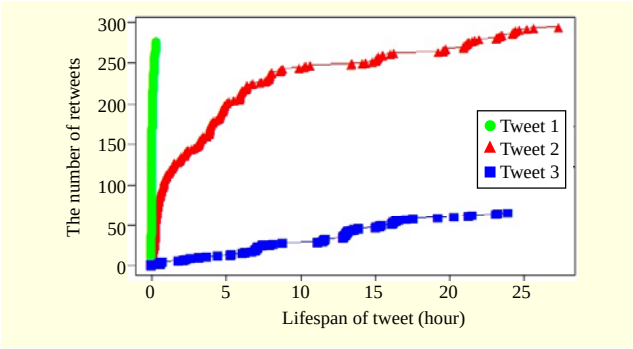

Fig. 1. Difference in retweet patterns.

popularity. In addition, through a comparative analysis of tweets 2 and 3, we found that the number of retweets is a factor for predicting popularity. Tweets 2 and 3 were consistently retweeted over a 25 hour period. However, Tweet 3 was not shared by a large number of users.

To tackle the above problems, we propose an algorithm for predicting the lifespan of a tweet and the number of times it is retweeted, which is a proxy for measuring its popularity. The rest of this paper is organized as follows. Section II introduces Twitter. Section III investigates related work, and section IV describes the features for prediction knowledge bases. Section V explains the algorithm. Section VI describes a performance evaluation of the proposed model as compared with conventional models. Finally, the conclusion is presented in section VII.

II. Understanding Twitter

We describe the Twitter system and the terms used. A simple social graph of Twitter is shown in Fig. 2. Twitter allows user C to follow user A if so interested. After following user A, user C will receive the tweets written by A. Based on this relation, we can say that user C is user A's follower, and user A is user C's followee. When two users are following each other, they are considered friends, as in users A and B in Fig. 2.

To represent the propagation of information, we create a retweet graph that is a type of directed graph. A retweet graph is created for each seed tweet (there are three types of tweet: normal tweet, retweet, and reply; with a seed tweet being a normal tweet). Retweet graph G = (V, E) is made up of set V consisting of retweeters and the author of the tweet and set E consisting of edges connecting the author and the retweeters.

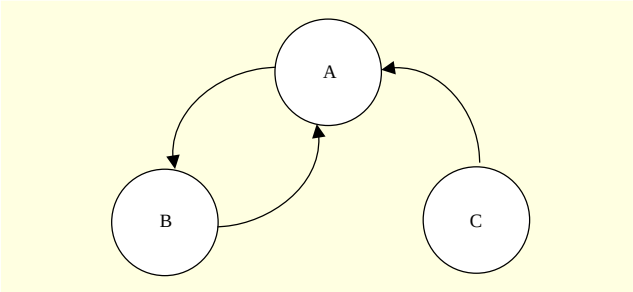

Fig. 2. Example of a social graph in Twitter.

III. Related Works

1. Predicting the Number of Retweets

Hong and others [3] classified the features of a tweet into four distinct sets and generated a training model using a logistic regression algorithm. They generated two prediction models: one is to classify whether the target tweet is retweeted or not, and the other is to predict the number of retweets. Unankard and others [4] proposed various approaches such as a *prior* model, which has been proven superior to the normal case [5], [6]. A classification approach based on user preference showed outstanding results. They assumed that people's interests in types of tweets differ, just as people's interests differ. They called this phenomenon "user preference." Zhang and others [7] categorized the tweets based on the events and predicted the number of retweets by simulating a retweet curve. A retweet curve represents the number of retweets in a successive time unit.

2. Predicting the Lifespan of a Tweet

Kong and others [2] defined the lifespan of a tweet as the period between when a tweet is created and when its last retweet is made. In their study, after analyzing the retweet patterns of an author's target tweet—for a period of one hour from the time it was first posted—they compared the retweet patterns of the same author's tweets written prior to the target tweet and extracted the top k tweets that showed similar patterns to the target tweet. Finally, they inferred the target tweet's lifespan by averaging the lifespan of the top k tweets.

3. Predicting the Possibility of a Response

The possibility of a response to a tweet is defined as the likelihood that either a retweet or reply will occur. Artzi and others [8] classified the features: historical, social, aggregate lexical, local contents, posting, and sentiment into four distinct sets and created a training model using the multiple additive regression trees (MART) algorithm. As a result, they found that the social features (follower or followee) have a significant impact upon the possibility of a retweet. In another related work, Zaman and others [9] confined the response to a retweet and employed the social and tweet features in their prediction model. They assumed that the possibility of a retweet will be

different depending on the time of posting and created an hourly training model using the passive-aggressive algorithm. Consequently, they achieved a more precise prediction capability than a human user. The results of the experiment implied that social features are the most effective. The authors in [10] also confined the response to a retweet and used item and user features to build a prediction model. To predict the likelihood of a retweet, they built a prediction model called "Match Box" and found that the author of a tweet and its retweeters have a significant effect on the capability of predicting retweet times.

IV. Features for Prediction Knowledge Bases

We divide the features into four distinct sets: social features, content features, posting time features, and local features. These features are listed in Table 1.

1. Social Features

A. Followers

According to [11], the number of retweets of a user's tweet is proportional to the number of their followers. A tweet written by a user with many followers has a high possibility of being retweeted. We assume given any two users having a similar number of followers that the number of retweets for each user will also be similar.

B. User Reliability

Despite having many followers, a user cannot be regarded as a social influencer. According to Twitter's policy [12], a user who has more followers than followees is considered a spammer. The number of followers can be represented by $d_O(v_i)$ of user v_i and the number of followees by $d_I(v_i)$. We can measure the reliability of a twitter account—namely, $R(v_i)$—based on the proportion of followers and followees by

$$R(v_i) = \frac{d_I(v_i)}{d_O(v_i) + d_I(v_i)}. \quad (1)$$

Table 1. Features of the prediction knowledge base.

Features	Components
Social	Number of followers, user reliability, user activity
Content	Tweet text, tweet informativeness
Posting time	Post time of tweet
Local	Duration of the retweet interval, retweet time of the time interval

C. User Activity

Through Twitter, users post information in real-time. An active user, who posts a lot, has a high possibility of being retweeted, and we can call such a user an influencer [13]. We therefore look to measure a user's activity, which can then be used as the standard for predicting the possibility of a retweet. The activity of user v_i (that is, $A(v_i)$) results from the number of tweets per day and is given by

$$A(v_i) = \frac{status\ (v_i)}{account\ creation\ date\ (v_i)}, \quad (2)$$

where *status* (v_i) is the number of tweets currently generated and *account creation date* (v_i) is the number of days from the account creation to the present date.

2. Content Features

A. Tweet Text

Tweets written by various users fill the Twitter stream. In [14], the most popular topics in Twitter were determined to be world issues and travel information. Second, users are interested in technology, sports, and so on. In [15], the authors verified that propagation of tweets differ depending on the topic. Therefore, we should predict the popularity of a tweet differently depending on its topic. For a given target, we measured the similarity between target and prediction knowledge base using the Jaccard similarity coefficient.

B. Tweet Informativeness

Twitter allows users to post short messages of 140 characters or less. Tweets exceeding 140 characters insert a URL to add further information. Because of this limitation, the length of a tweet and the inclusion of a URL form the basic criteria when judging informativeness. The informativeness of tweet t_i (that is, $I(t_i)$) is calculated based on the proportion of its length to the maximum possible tweet length

$$I(t_i) = \begin{cases} 1 & \text{if URL} \in t_i; \\ \dfrac{length\ of\ used\ characters\ (t_i)}{maximum\ length\ of\ a\ tweet} & \text{else.} \end{cases} \quad (3)$$

3. Posting Time of a Tweet

The posting time of a tweet is the time given at its creation. There are different retweet patterns depending on whether it is created during the day or during the night. According to [16], a tweet has a high possibility of propagating if created between 8 a.m. and 11 p.m.—when most users are active. Tweets created at other times, however, have a low possibility of propagating. We assume that the propagation patterns of tweets

ETRI Journal, Volume 36, Number 3, June 2014
http://dx.doi.org/10.4218/etrij.14.0113.0657

will be similar if they are written at a similar time.

4. Local Features

The local features consist of two components: one is the duration of the retweet interval (TRI) and the other is the retweet times of the time interval (RTI). In the following description, we use these abbreviations for convenience.

A. TRI

When a tweet is retweeted n times, we equally divide n into k (retweet-time) units. TRI indicates the time from the posting time of the first tweet in unit k to the posting time of the last retweet in unit k. We define the posting time of tweet t_i as $Time(t_i)$, and tri_r can be computed using the following formula:

$$tri_r = Time\left(t_{(r+1)\left(\frac{n}{k}\right)}\right) - Time\left(t_{r\left(\frac{n}{k}\right)}\right) \quad . \quad (4)$$
$$(0 \le r \le k-1)$$

We convert $TRI(t_i)$ into a numeric vector as follows:

$$TRI(t_i) = < tri_0, tri_1, ..., tri_{k-2}, tri_{k-1} > .$$

B. Retweet Times of RTI

When the lifespan of a tweet is maintained for more than j minutes, we equally divide j into p time units. RTI indicates the number of retweets occurring in a given time unit p. We can convert $RTI(t_i)$ into a numeric vector as follows:

$$RTI(t_i) = < RC_1, RC_2, RC_3, ..., RC_n > .$$

V. Framework for Predicting Tweet Popularity

We propose a framework for predicting the popularity of a tweet. As shown in Fig. 3, the framework is partitioned into two phases. In the generation phase of the prediction knowledge bases, using the Hadoop cluster, we first remove the tweets posted by a spammer and make a retweet graph. We then extract the features that are used to make predictions from the retweet graph, and based on the tweet creation time we generate knowledge bases. In the prediction of the tweet popularity phase, a given target tweet is entered into the knowledge base. N tweets—each having a similar topic with the target tweet—are then extracted. From the knowledge base, only M tweets among N—that have an analogous retweet pattern—are passed on to the next processing stage. We extract T tweets with a similar property to that of the target tweet. Finally, we predict the popularity of tweets based on the knowledge extracted from T tweets.

1. Prediction Knowledge Base Algorithm

We extract the features described in section IV from previous popular tweets. Algorithm 1 shows the generative processes of the prediction knowledge bases.

Algorithm 1 Generating prediction knowledge bases

Input
S: a set of seed tweets
SL: spammer list
TWEET: tweet collection
Output
KB_i: knowledge bases
1: T = MapReduce(TWEET, SL)
2: **for** $seed_t$ in S
3: enQueue(tweet_id of $seed_t$)
4: **while** queue_size > 0 **do**
5: tweet_id = deQueue()
6: **for** rtg_i in T
7: **if** tweet_id = name of rtg_i **then**

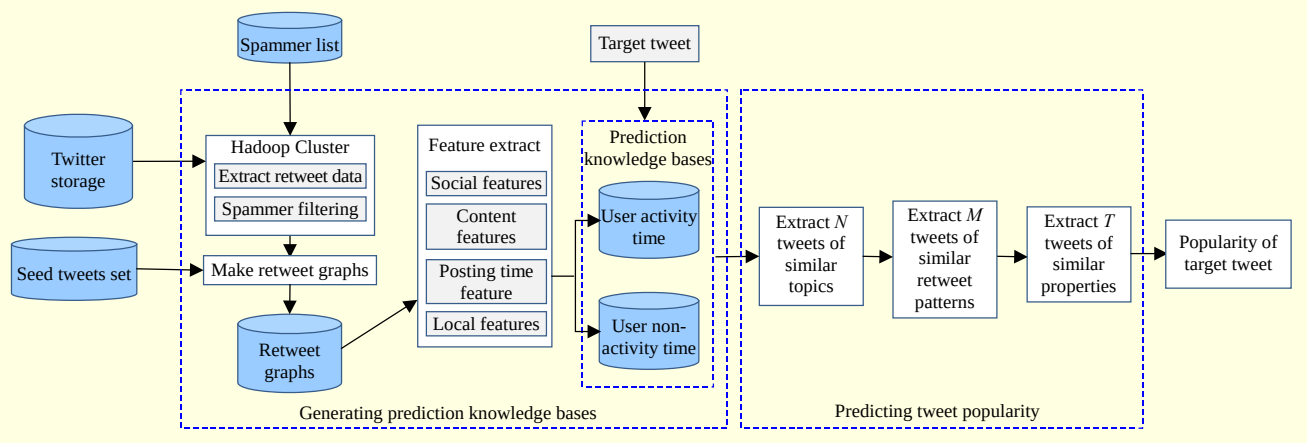

Fig. 3. Framework for predicting tweet popularity.

```
8:                addGraph(seed_i, rtg_i)
9:                enQueue(each tweet_id of rtg_i)
10:             end if
11:         end for
12:     end while
13:   feature_i = extractFeature(seed_i)
14:     store(feature_i, KB_i)
15: end for
```

A. Data collection

Only tweets written in Korean were included in this study, since the posting time changes depending on the time zone of users. We collected the tweets using a Twitter Stream API that allows the tweets to be searched based on keywords or strings.

B. Data Preprocessing

This step is the stage of preprocessing the collected tweets, which corresponds to line 1 in Algorithm 1. A tweet provides abundant metadata describing its own status. According to the type of tweet, examples of simple metadata are shown in Figs. 4 and 5. Figure 4 shows the status when a seed tweet is posted. In this example, the tweet has metadata that includes the identity (id) of both the tweet and the user.

Figure 5 shows the status when a follower retweets the seed tweet. In this example, the tweet includes not only its id and the id of the follower, but also the id of the seed tweet and id of the user who posted the seed tweet.

If several users retweet the same tweet, they share the same seed-tweet id. We therefore regarded tweets that share the same seed-tweet id as one group and assigned a name according to this id. In Algorithm 1, rtg_i belongs to one such group.

MapReduce algorithm is suitable for processing the retweet data, as shown in Fig. 5. Algorithm 2 is designed to filter spammers and collect retweet data. After the end of the map function, the reduce function in Algorithm 3 receives pairs of <key, List<value>> values. Key corresponds to an id of a user who posts the seed tweet, and List<value> corresponds to a group of users that share the id of the seed tweet. The reduce function is used to provide results in the preprocessing step. The output of the reduce function is a set of tweet-id groups that exclude tweets written by spammers.

Algorithm 2 Map function for preprocessing

```
1: global var spammer_list ← allocate()
2: map(LongWritable key,/*default */, Text tweet/*Collection of Tweets*/)
3:     if(spammer_list_contains(user id of tweet))
4:         continue
5:     if(is retweets (tweet))
6:         continue
7:     write(id of user who posts seed tweet, id of tweet)
8: spammer_list ← free()
```

```
{
  "id" : 386926275769544704
  "text" : "131004 Pusan International Film Festival"
  "user" : {
             "id" : 1574482472
           }
}
```

Fig. 4. Example metadata of tweet.

```
{
  "id" : 286943947328593920
  "text" : "RT @XXXXX 131004 Pusan International Film Festival"
  "user" : {
             "id" : 103268882
           }
  "retweeted_status" : {
                         "id" : 386926275769544704
                         "text" : "131004 Pusan International Film Festival"
                         "user" : {
                                    "id" : 1574482472
                                  }
                       }
}
```

Fig. 5. Example metadata of retweeted tweet.

Algorithm 3 Reduce function for preprocessing

```
1: reduce(id of seed tweed, List<id of tweet> list)  /*<key, List<value> >*/
2:     for id of tweet in list
3:         write(id of seed tweet, id of tweet)
4:     end for
```

C. Creating Retweet Graphs

Lines 4 through 12 in Algorithm 1 show the steps for creating retweet graphs. We create retweet graphs using seed tweets—classified as popular tweets—and a set of groups MapReduce. For the first step, we try to detect whether a set of groups include the id of the seed tweet. If the same id is found, we generate a retweet graph by adding a group to the seed tweet. In the second step, since a graph can have a sub-graph, we try to detect other groups by comparing each tweet id of the graph generated in the first step with the names of the groups in the set.

D. Extracting Features

We extract the social features, content features, posting-time features, and local features in line 13 of Algorithm 1. We extract the TRI and RTI from the retweet graphs and social features, content features, and posting-time features from the seed tweet—some of which are used for extracting other

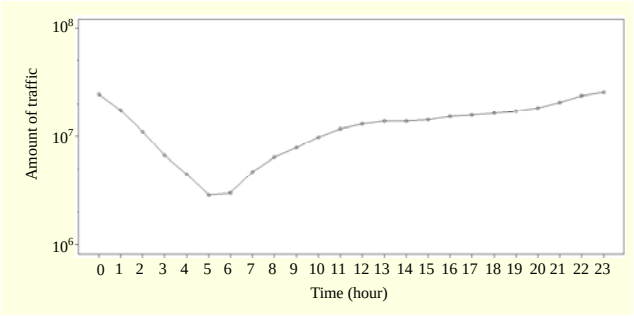

Fig. 6. Traffic distribution in Twitter.

features such as informativeness of a tweet and reliability or activity of authors.

E. Constructing Prediction Knowledge Bases

We store the extracted features in two knowledge bases, as considered in section IV. 3, which corresponds to line 14 in Algorithm 1. We analyzed the amount of traffic being driven by Twitter to divide our data into two parts: one is in user-active time and the other is in user-inactive time. Figure 6 shows that traffic in Twitter varies depending on the time.

As a result, we determined that user-active time is from 12 p.m. to 2 a.m. and user-inactive time is otherwise. In predicting the popularity of a tweet, we select the prediction knowledge base based on when the tweet was created.

2. Predicting the Popularity of a Tweet

When the target tweet is entered, we extract tweets with similar topics, retweet patterns, and properties sequentially from the prediction knowledge bases. Algorithm 4 shows this prediction process.

Algorithm 4 Predicting tweet popularity

Input

et_i: target tweet

KB_i: prediction knowledge bases

α, β, γ : the number of knowledge extracted at each step

Output

Popularity(lifespan, retweet times)

1: KB_i = selectKnowledgeBase(et_i)
2: TS_i = topicSimilarity(et_i, KB_i, α)
3: GS_i = RetweetPatternSimilarity(et_i, TS_i, β)
4: US_i = PropertiesSimilarity(et_i, GS_i, γ)
5: Let estimated popularity = 0
6: **while** $\gamma > 0$ **do**
7: estimated popularity+ = REALPOPULARITY($US_{i,\gamma}$)
8: $\gamma = \gamma - 1$
9: **end while**
10: Popularity = estimated popularity / γ

A. Extracting Tweets of Similar Topics

When the target tweet is given, we select the prediction knowledge base based on the posting time and extract the top α tweets that have a similar topic to the target tweet. This corresponds to lines 1 and 2 in Algorithm 4. We then measure the topic similarity, a function of the Jaccard similarity of a text bigram.

B. Extracting Tweets with Similar Retweet Patterns

This step corresponds to line 3 in Algorithm 4. We extract the top β tweets that have similar retweet patterns to the target tweet. The retweet patterns are the RTI and TRI. In this paper, we set parameters n = 100 and k = 5 in extracting $TRI(t_i)$ and parameters j = 60 and p = 6 in extracting $RTI(t_i)$. We measure the similarity between the target tweet et_i and the extracted top α tweets ht_i, using the Euclidian distance.

C. Extracting Tweets with Similar Properties

In this stage, we extract the top γ tweets with similar properties from the top β tweets, considering reliability, activity, and informativeness, in line 4 of Algorithm 4. We define $R(et_i)$, $A(et_i)$, and $I(et_i)$ as the reliability, activity, and informativeness of the target tweet, and $R(ht_i)$, $A(ht_i)$, and $I(ht_i)$ as the reliability, activity, and informativeness of the extracted top β tweets, respectively.

$$DIST(et_i, ht_i) = \sqrt{[(R(et_i)) - R(ht_i)]^2 + [(A(et_i)) - A(ht_i)]^2 + [(I(et_i)) - I(ht_i)]^2} \quad (5)$$

D. Predicting the Popularity of Tweets

This stage predicts the lifespan and retweet times of the target tweet, corresponding to lines 6 through 10 in Algorithm 4. We acquire a set of tweets $KB_i = \{ht_0, ht_1, \ldots, ht_\gamma\}$, which are the most similar with the target tweet in that they satisfy the topic, retweet pattern, and properties of the target tweet. Finally, we predict the lifespan and retweet times of the target tweet et_i by

$$Popularity(et_i) = (1/\gamma) * \sum_{i=1}^{\gamma} Popularity(ht_i). \quad (6)$$

VI. Experiment

1. Data Analysis

To evaluate our approach, we conducted data collection from

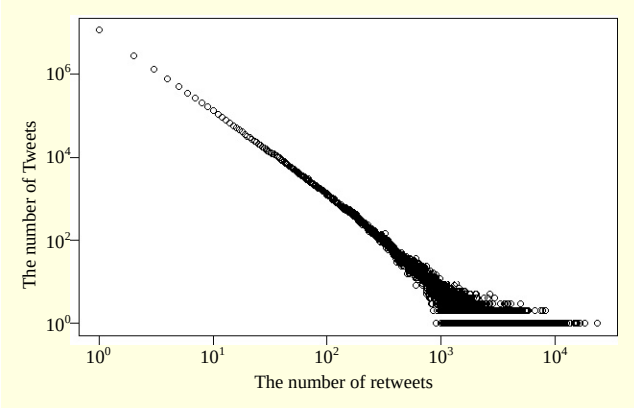

Fig. 7. Retweet distribution.

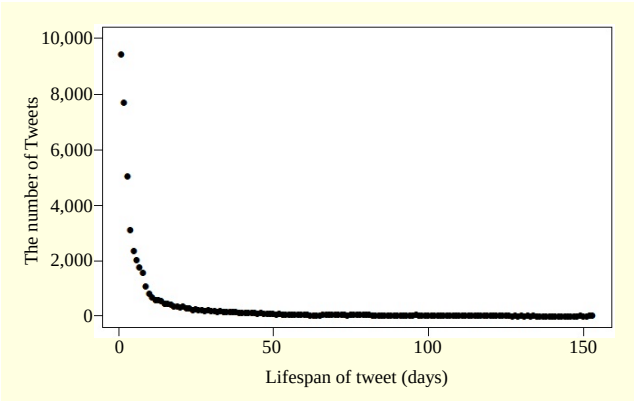

Fig. 8. Lifespan of a tweet distribution.

Twitter between June 2012 and October 2012. The dataset contains 473 million postings generated by about 3.7 million users. Figure 7 shows the distribution of retweets.

In Fig. 6, the ratio of tweets retweeted more than 100 times is very small, at about 1% (50,717 tweets), compared to the total number of tweets. In terms of the number of retweets, we assume tweets that are retweeted more than 100 times were popular tweets in the past. In addition, we analyze the lifespan of tweets that were retweeted more than 100 times, as shown in Fig. 8. Although the tweets show various lifespan distributions, the number of tweets after ten days is lower. Figure 8 shows that the number of tweets within a ten-day period accounts for 70% (34,862 tweets). We use only tweets that were created within the first ten days, since the number of tweets after this point is not enough to be used as a prediction knowledge base, and it can be assumed that they were retweeted accidentally. Throughout our analysis, we define popular tweets as those having been retweeted more than 100 times and having a lifespan within the ten-day period.

2. Experimental Data

We divided our five-month dataset into training data generated from June 2012 to September 2012 and the test data generated in October 2012.

Table 2. Experimental data for predicting lifespan.

Duration of a tweet's lifespan (hours)	Number of graphs used in prediction knowledge base	Number of graphs used in test data
0 – 24	8,697	2,394
24 – 48	5,638	1,783
48 – 72	3,071	1,352
72 – 96	2,189	779
96 – 120	1,709	572
120 – 144	1,470	455
144 – 168	1,314	384
168 – 192	1,049	330
192 – 216	741	204
216 – 240	580	151

Table 3. Experimental data for predicting retweet times.

Number of graphs used in prediction knowledge base	Number of graphs used in test data
26,458	8,404

Table 2 shows the dataset used for predicting the lifespan of a tweet. The authors in [2] suggested using a limited duration of 0 hours to 72 hours to evaluate the performance. However, while we used the range of prediction suggested by [2] we also extended it. Table 3 shows the dataset used in predicting the number of retweets.

3. Comparison of Prediction Models

We compare our model with other conventional models. There has been only one conventional research regarding the prediction of a tweet lifespan. Of the algorithms proposed in [2], ATR-KNN (K-Nearest Neighbor) outperformed other approaches. The ATR represents the same author, similar post time, and retweet patterns. However, it was impossible to predict the lifespan when there was no historical data at all or when less than five postings were written by the same author. We approached the following problems that may occur in the previous model [2]. For instance, when an author has posted only three tweets in the past, we extracted two similar tweets from other authors by simply considering the retweet patterns and posting time. Regarding the prediction of the number of retweets, we compared the classification based on user preference, which outperformed the various approaches proposed in [4]. The interestingness scores of all candidate

ETRI Journal, Volume 36, Number 3, June 2014
http://dx.doi.org/10.4218/etrij.14.0113.0657

users were trained by extracting the retweet information that represents the relation between an author who posts the seed tweet and the follower who posts it in a specific category. The interestingness score indicates how likely it is that a user will retweet the seed tweet in a specific category. The number of retweets can be calculated by adding candidate users that are over a certain preference threshold. However, because the training set did not contain all user preferences, it may be impossible to measure some user preferences in the test data. Therefore, we considered only candidate users whose preference information was included in the training data. In [7], tweets were classified into several categories depending on the event, and they selected the top N tweets from each category that were retweeted the most. They inferred the target tweet's number of retweets by measuring the curve similarity that relies on the ratio between the target tweet and the top N tweets in each time unit.

4. Evaluation Metrics

In this research, we used different evaluation methods according to the prediction condition. In predicting the lifespan of the tweets, we use the root-mean-square error (RMSE) to obtain the time tolerance between the actual observed lifespan and the estimated lifespan. In (7), N represents the amount of test data, $Lifespan_r(t_i)$ represents the actual observed lifespan of tweet t_i, and $Lifespan_p(t_i)$ represents estimated lifespan. RMSE is calculated by

$$RMSE = \sqrt{1/N * \sum_{i=1}^{N} [Lifespan_r(t_i) - Lifespan_p(t_i)]^2}. \quad (7)$$

Instead of directly predicting the exact number of retweets, we evaluated the accuracy of the prediction. The prediction tolerance of tweet t_i, $PredictionError(t_i)$, is the ratio of the actual observed number of retweets, $RetweetTimes_r(t_i)$, to the estimated retweet times, $RetweetTimes_p(t_i)$, as shown in the following formula:

$$PredictionError(t_i) = |RetweetTimes_r(t_i) - RetweetTimes_p(t_i)| / RetweetTimes_r(t_i). \quad (8)$$

If the $PredictionError(t_i)$ is less than the error threshold, we can say that t_i is correctly predicted. We set up the various ranges of error threshold, ranging from 5% to 30%. In other words, the error threshold is the level of difficulty. The precision is the ratio of the number of tweets whose $PredictionError(t_i)$ is less than the error threshold to the total number of tweets, as shown in the following formula:

$$Precision = \frac{the\ number\ of\ tweets\ less\ than\ error\ threshold}{total\ number\ of\ tweets}. \quad (9)$$

5. Feature Analysis

We analyzed the features for measuring their usefulness. First, we evaluated the performance of each feature in the prediction tasks. We also calculated how combining features impact the performance. Figures 9 and 10 show the analysis results. Figures 9(a) and 10(a) are the results of predicting the popularity of a tweet using a single feature. Both results indicate that RTI and TRI are useful features related to retweet patterns. Informativeness of the tweet shows the lowest value in predicting the lifespan of a tweet. Considering the limited number of characters of a tweet, we found that it is difficult to predict the popularity of a tweet with only its informativeness. Figures 9(b) and 10(b) show the results of predicting the popularity of a tweet using a group of features. The group of features is woven from a similar disposition. The retweet patterns consist of RTI and TRI. The properties pattern consists of user reliability, user activity, and informativeness of a tweet. Both results indicate that the retweet patterns outperform the other group of features. In addition, we found that combining features shows a higher performance than using only single

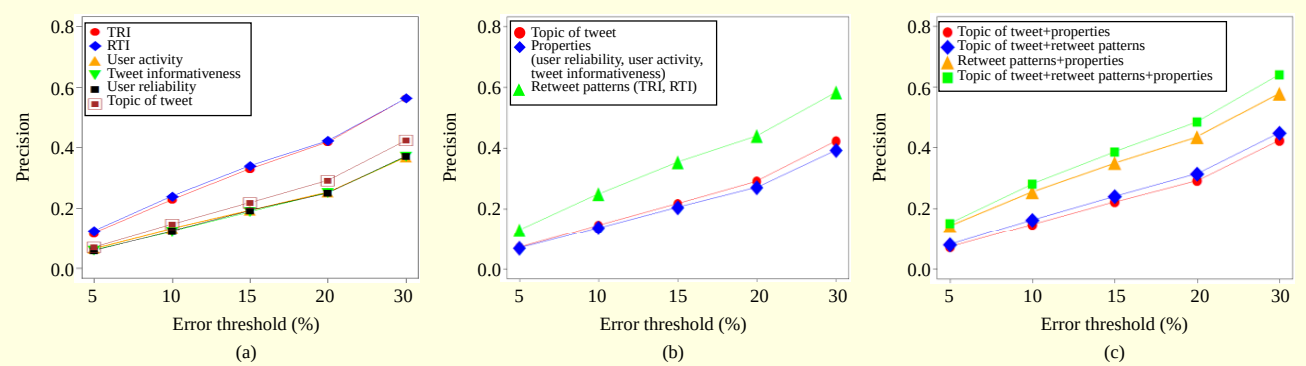

Fig. 9. Analyzing features for predicting lifespan of a tweet: (a) single feature, (b) group of features, and (c) combining groups of features.

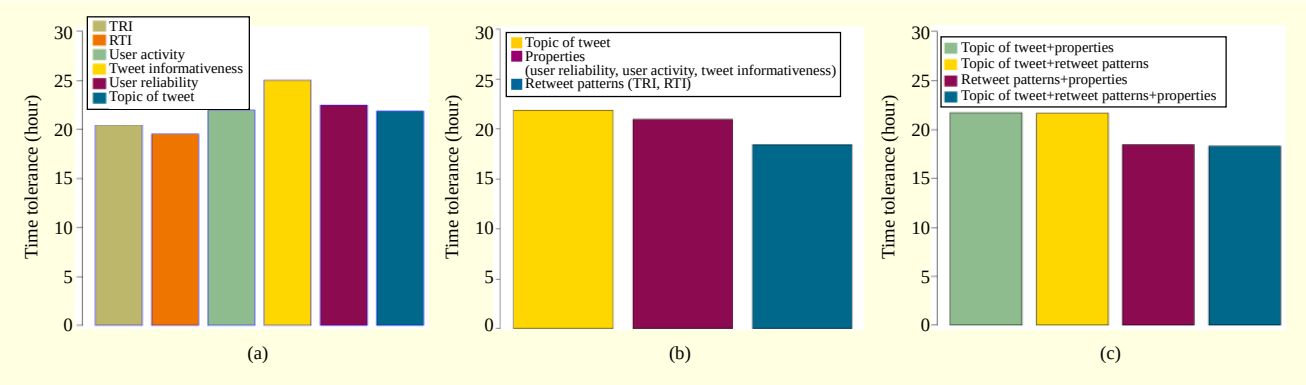

Fig. 10. Analyzing features for predicting the number of retweets: (a) single feature, (b) group of features, and (c) combining groups of features.

features. Based on the results of the feature analysis shown in Figs. 9(b) and 10(b), we combined each group to find the optimum combination of features. Figures 9(c) and 10(c) show the results of predicting the popularity of a tweet using groups of features. Both results show that the combination consisting of tweet topic, retweet patterns, and property patterns outperformed other combinations.

6. Experimental Results

We carried out the experiment based on various prediction ranges. Table 4 shows that about six hours of tolerance exists within a prediction range of 24 hours and that about 54 hours of tolerance exists within a prediction range of 240 hours. Because the previous model limits the prediction range to 72 hours, we evaluated the limited range to construct a similar experiment environment. In addition, we extended the prediction range to evaluate whether it works well under flexible conditions.

Table 5 shows the comparison of the two models within a period of 72 hours. The proposed model results in an outstanding accuracy compared with the previous model [2], having a tolerance of about 18 hours. In the previous model [2], the author's historic posting is the most significant feature for predicting the lifespan of a tweet. In other words, the tolerance increases when not enough historic postings are written by the same author. The performance results in extending the prediction range are shown in Fig. 11. The performance is similar to within three days. The immense prediction knowledge base of the first three days can be useful for the model in [2]. However, as the prediction range widens, the performance difference becomes increasingly larger.

The precision in predicting the number of retweets was evaluated according to the respective error threshold. Because using only 33 test data in the previous models [4] and [7] caused low reliability, we evaluated the test data shown in Table 3 to enhance the reliability. Figure 12 shows the results of the experiment. When we set the error threshold to about 20%, the proposed model achieved a significantly outstanding precision at about 0.5, in contrast to conventional models, which showed a precision of around 0.3 and 0.2, respectively. In [7], despite excluding 4,790 unpredictable datasets from all 8,944 test datasets, it shows the lowest performance among them. We concluded that the user-preference property is changeable as time passes and is not handled flexibly when new users are detected.

In [4], the model is similar to the proposed model, wherein it is based on similar historic tweets. However, it relies

Table 4. Time tolerance within prediction range.

Prediction range (hours)	Time tolerance (hours)
0 – 24	6.16
0 – 48	11.43
0 – 72	18.32
0 – 96	23.73
0 – 120	29.27
0 – 144	35.22
0 – 168	41.23
0 – 192	46.85
0 – 216	50.76
0 – 240	54.87

Table 5. Comparison of model performance within a range of 72 hours.

Algorithm	Time tolerance (hours)
ATR-KNN (Kong 12)	22.24
Proposed method	18.32

ETRI Journal, Volume 36, Number 3, June 2014
http://dx.doi.org/10.4218/etrij.14.0113.0657

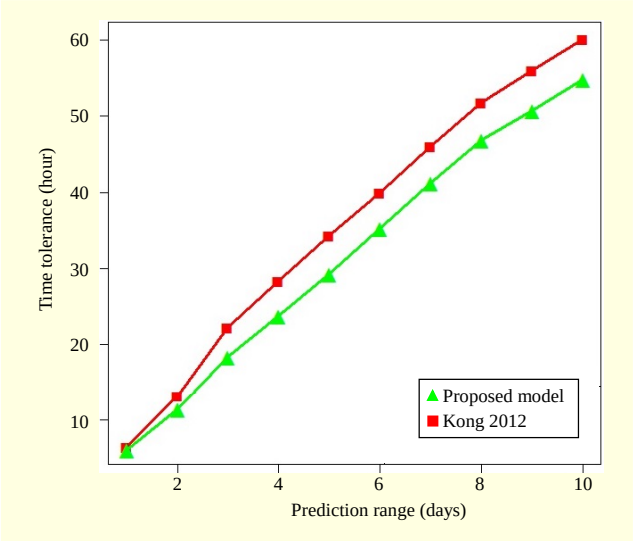

Fig. 11. Comparison of model performance according to time range.

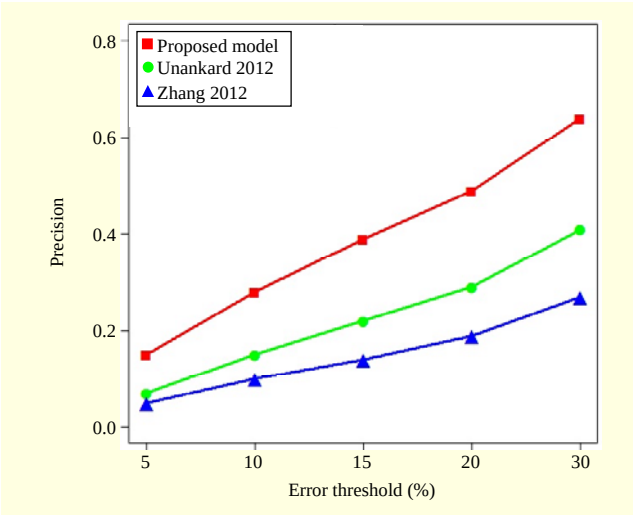

Fig. 12. Comparison of model performance according to error threshold.

significantly on the original number of retweets of the training dataset, and the results can therefore be variable.

VII. Conclusion

In this paper, we propose an algorithm to predict the lifespan of a tweet and the number of retweets, which are a proxy for measuring its popularity. To achieve this, we suggest a prediction framework of tweet popularity consisting of two phases: one is generating prediction knowledge bases, and the other is predicting the tweet's popularity. In the phase of generating prediction knowledge bases, we analyzed the features that affect a retweet to construct the prediction knowledge bases. In the phase of predicting the tweet's popularity, we extract historical tweets that have similar properties to those of the target tweet in a step-by-step manner.

As shown in the experimental results, our model can perform better than previous prediction models, for the following reasons. First, there are few constraints on the target for prediction. A previous model predicted the popularity of a tweet based on either user preferences or the historical tweets posted by the same author. As a constraint of the conventional model, the prediction is possible, if and only if, there is sufficient information; this can lead to difficulty in predicting the popularity. However, the proposed model does not have the above problems, because it is based on similarity with the target tweet. Second, our model has excellent scalability. In reality, the lifespan of a tweet is wide ranging; and to use conventional methods of prediction, historical tweets written by the author of the targeted tweet must consist of various distributions. In other words, sufficient historical data for each prediction range is required. However, our approach deals well with the above constraint as it considers collaborative features. In addition, this method has an advantage of extracting more similar historical tweets.

References

[1] H.J. Oh, C.K. Lee, and C.H. Lee, "Analysis of the Empirical Effects of Contextual Matching Advertising for Online News," *ETRI J.*, vol. 34, no. 2, Apr. 2012, pp. 292–295.

[2] S. Kong et al., "Predicting Lifespans of Popular Tweets in Microblog," *Int. ACM SIGIR*, Portland, Oregon, USA, Aug. 12–16, 2012, pp. 1129–1130.

[3] L. Hong et al., "Predicting Popular Messages in Twitter," *Int. Conf. WWW*, Hyderabad, India, Mar. 28 – Apr. 1, 2011, pp. 57–58.

[4] S. Unankard et al., "On the Prediction of Re-tweeting Activities in Social Networks - A Report on WISE 2012 Challenge," *WISE*, Paphos, Cyprus, vol. 7651, Nov. 28–30, 2012, pp. 744–754.

[5] C.K. Lee and M.G. Jang, "A Prior Model of Structural SVMs for Domain Adaptation," *ETRI J.*, vol. 33, no. 5, Oct. 2011, pp. 712–719.

[6] C.K. Lee and M.G. Jang, "A Modified Fixed-Threshold SMO for 1-Slack Structural SVMs," *ETRI J.*, vol. 32, no. 1, Feb. 2010, pp. 120–128.

[7] L. Zhang, Z. Zhang, and P. Jin, "Classification-Based Prediction on the Retweet Actions over Microblog Dataset," *WISE*, Paphos, Cyprus, Nov. 28–30, 2012, pp. 771–776.

[8] Y. Artzi, P. Pantel, and M. Gamon, "Predicting Responses to Microblog Posts," *NAACL HLT*, Montreal, Canada, June 3–8, 2012, pp. 602–606.

[9] S. Petrovic et al., "RT to Win! Predicting Message Propagation in Twitter," *ICWSM*, Barcelona, Catalonia, Spain, July 17–21, 2011, pp. 586–589.

[10] T.R. Zaman et al., "Predicting Information Spreading in Twitter," *Workshop CSSWC NIPS*, Whistler, Canada, Dec. 10, 2010.

[11] B. Suh et al., "Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network," *IEEE, Int. Conf. SocialCom*., Minneapolis, MN, USA, Aug. 20–22, 2010, pp. 177–184.

[12] Twitter Inc. *The Twitter Rules of Spam and Abuse*. Accessed Oct. 20, 2012. http://support.twitter.com/articles/18311-the-twitter-rules

[13] C. Castillo, M. Mendoza, and B. Poblete, "Information Credibility on Twitter," *Int. Conf. WWW*, Hyderabad, India, Mar. 28 – Apr. 1, 2011, pp. 675–684.

[14] W.X. Zhao et al., "Comparing Twitter and Traditional Media Using Topic Models," *ECIR*, Dublin, Ireland, Apr. 18–21, 2011, pp. 338–349.

[15] C. Wang and B.A. Huberman, "Long Trend Dynamics in Social Media," *EPJ Data Sci*., vol. 1, no. 1, May 2012.

[16] B. Krishnamurthy, P. Gill, and M. Arlitt, "A Few Chirps about Twitter," *WOSN*, Glasgow, Scotland, UK, Apr. 1–4, 2008, pp. 19–24.

Yongjin Bae received his BS degree in computer education from Mokwon University, Daejeon, Rep. of Korea, in 2012 and his MS degree in computer software and engineering from the University of Science and Technology, Daejeon, Rep. of Korea, in 2014. His main research interests are social big data analytics and text mining.

Pum-Mo Ryu received his BS degree in computer engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1995 and his MS degree in computer engineering from POSTECH, Pohang, Rep. of Korea, in 1997. He received his PhD degree in computer science from KAIST, Daejeon, Rep. of Korea, in 2009. Currently, he is a senior researcher in Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. His research interests include natural-language processing, text mining, knowledge engineering, and question answering.

Hyunki Kim received his BS and MS degrees in computer science from Chunbuk National University, Jeonju, Rep. of Korea, in 1994 and 1996, respectively. He received his PhD degree in computer science from the University of Florida, Gainesville, USA, in 2005. Currently, he is a principal researcher in Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. His research interests include natural-language processing, machine learning, question answering, and social big data analytics.