Trendy Episode Detection at a Very Short Time Granularity for Intelligent VOD Service: A Case Study of Live Baseball Game

Hogun Park, Sun-Bum Youn, Geun Young Lee, Heedong Ko
Imaging Media Research Center, Korea Institute of Science and Technology (KIST), Seoul, Korea
{hogun,dmonkey,gylee,ko}@imrc.kist.re.kr

ABSTRACT
For intelligent VOD service, social networking services (SNS) like Twitter provide great potential for extracting practical metadata of live TV contents. For example, when breaking news occurs, social media bring not only background but also the trend of public opinion almost in real-time. Many previous researches addressed to analyze social media to detect meaningful topics and trends, but they did not attempt to approach a *live event at a very short time granularity*. In contrast, we aim to identify a time constrained entity-relation graph, so-called *"episode."* The episode is to annotate a real-time event and to provide intelligent VOD service for future IPTV. In our experimental study, we evaluated 33 baseball games and also implemented a baseball watching system.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information Filtering; *J.4* [**Computer Applications**]: Social and Behavioral Science

General Terms
Design, Algorithms

Keywords
Social Media, Twitter, Bursty Feature Detection

1. INTRODUCTION
With the increasing popularity of Twitter, numerous spatio-temporal data are being created in real-time. Its real-time nature benefits us to understand a current event at the very moment. For example, as soon as Mr. Choo in Cleveland hits homerun, the news propagates immediately over Twitter, so that we can notice it at the very moment it occurs. This characteristic has great advantage to provide immediate and intelligent support for viewers who are watching TV. The support includes extracting a rich episode description with relevant videos and helping viewers to join a new social community. To support them intelligently, it is necessary to provide a framework to detect an emerging episode and associate it with relevant videos and people. In this paper, we propose a framework to discover trendy episodes of a live event and implemented a prototype of live baseball watching system. For evaluation, we attempted to discover noticeable trendy episodes of 33 baseball games and compared it with official baseball records. The live baseball watching system detects a trendy moment during a live event and visualizes its trendy episodes with NL (Natural Language)-based description and relevant videos.

EuroITV2011, June 29th-July 1st 2011, Lisbon, PORTUGAL.

2. RELATED WORK
In contrast to many attempts to extract hot topics in news and blogs domains, the small number of works has targeted real-time social media. Most of hot topic detection in social media stream has been attempted in finding a global topical trend and detecting a single instance of seminal events and disasters. *Tweettronics*[1] provides a summary using topical trends and sentiment tendencies with respect to company brands or products. Web2express Digest[2] makes use of Twitter to find real-time topical trends. It monitors Twitter's real-time stream and helps people to discover any topic of interest to follow. A recent study [5] tried to address tweets tagged with a semantic web conference. They observed that people utilized Twitter to identify what happened during a conference, and they were willing to spread interesting news. However, they did not approach a *live event at a very short time granularity*.

For intelligent VOD Service of TV contents, there were several attempts for extracting practical metadata from social media. [7] suggested a concept that chat data of viewers can be extracted to provide a summary of TV contents. In addition, [2][8] implemented a synchronized group watching interface and indexing VOD contents using their chat data. However, only keyword-level indexing was accomplished and did not detect a topic at a level of *episode*.

3. DEFINITION OF EPISODE
DEFINITION **[Episode]** Given a *live event*, *L*, an *episode, EP* is composed of dynamic relationship graphs, $R=(E, E, W, T, S)$ *where* E represents an entity, T is an appearance time, and S stands for corresponding social network information of opinion leaders and their messages.

An episode is intuitively a set of dynamic relationship graphs. The graph is composed of relations among two entities with a time stamp and also includes social network information of

[1] http://www.tweettronics.com/

[2] http://help.web2express.org/about-digest

corresponding messages of opinion leaders for later interaction. Entity is represented by a noun which indicates people or activities. Therefore, in order to detect an episode from social media, we have four sub-problems to solve: *(1) entity detection (2) relationship graph construction (3) appearance time estimation (4) social network information*. In this paper, we focused on describing (1),(2)&(3) and evaluating (1)& (3) using official baseball records. (4) is also an important challenging problem and have been many research efforts. However, for concentrating our core framework, we simply recommended users who mentioned the entity frequently.

4. SYSTEM OVERVIEW

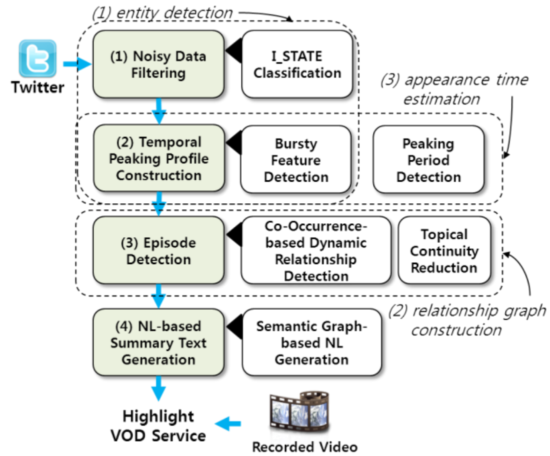

Figure 1. System Architecture

The architecture of our proposed approach is illustrated in Figure 1. First, noisy Twitter data were detected by our semantic classifier. We created a SVMs classifier (*Section 3.1*) to find noisy tweets. It was trained given a set of positive and negative training samples to filter out I_STATE [6] expressions (e.g. hope/expectation) and other types of noisy messages. Each training set is composed of keyword features like "seems like" or "I believe that" and statistical features. After the classification, a temporal peaking profile construction module firstly finds boundaries of individual peaking periods utilizing their gradients and averages on its frequency curve (*Section 4.2.1*.) Later on, it identifies bursty features of the period by utilizing our parameter-free bursty feature extraction algorithm (*Section 4.2.2*.) In order to find bursty features, it computes a probability that each feature in a period is likely to make a burst. Subsequently, the extracted bursty features are used for episode detection (*Section 4.3*) as a form of dynamic relationship graphs. Detected episode is summarized with NL-based description (*Section 4.4*.) Figure 2 shows a sample result of our system, and it detects two trendy episodes with rich contextual information like "*who*," "*what*," and "*opinion leader*."

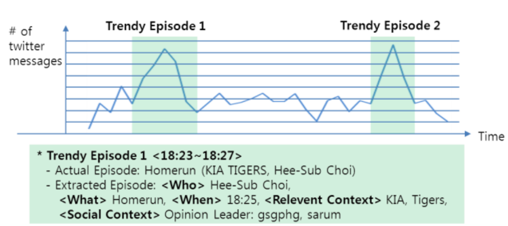

Figure 2. Examples of System Output

4.1 Noisy Social Media Filtering

Table 1. Types of Noisy Social Media

Noisy Types	Description
I_State [6] (e.g. believe, intend, or want)	Presenting their expectation or expressing their hope
Others	Miscellaneous things (meaningless mentions, excessive repeats of words, or messages which is too short to understand)

For detecting the above types of noise, we use of two groups of features for each tweet as follows:

(1) Keyword features: terms in a tweet.

(2) Statistical features: the number of words appearing at a message and the maximum term frequency at a message.

Given the features, we created a SVMs (Support Vector Machines) classifier to find noisy Twitter data. We trained the classifier using a collection of positive and negative training sets from Twitter, and only positive Twitter data were used for later temporal peaking profile construction.

4.2 Temporal Peaking Profile Construction

4.2.1 Peaking Period Detection

Algorithm : Peaking Period Detection

Let C be a collection of tweets
Let $\mu \leftarrow \mu(C)$ // *the average of C*
Let $\sigma \leftarrow \sigma(C)$ // *the standard deviation of C*
$k \leftarrow 0$
repeat
  $k \leftarrow k + 1$
  $E_k \leftarrow \emptyset$
  find the highest count c_k in a given C
  s_l = left end of the spike having a peak c_k
  s_r = right end of the spike having a peak c_k
  for each $s_l < c_j < s_r$ **do**
    $C \leftarrow C - \{ c_j \}$
    $E_k \leftarrow E_k + c_j$
  end for
until $\forall c \in C > \mu + 2\sigma$
return $\{ E_1, \ldots, E_k \}$

A peaking period is a unit of trendy time periods of an event. Detecting a peaking period can help to understand the most meaningful moment from a point of social impact. In this section, we will describe how to detect a peaking period of a baseball game using temporal patterns of social media. As in the below graph of Figure 3, sudden spikes caused by emerging number of tweets could be constantly found. To detect a spike, our algorithm checks out only if it appears significantly bigger than the average and finds spikes that increase exponentially. To specific, it firstly looks for global maximums from the set *C*. Then, utilizing the timestamp of each maximum value, it investigates left and right-ends of the corresponding spike. The above algorithm describes details.

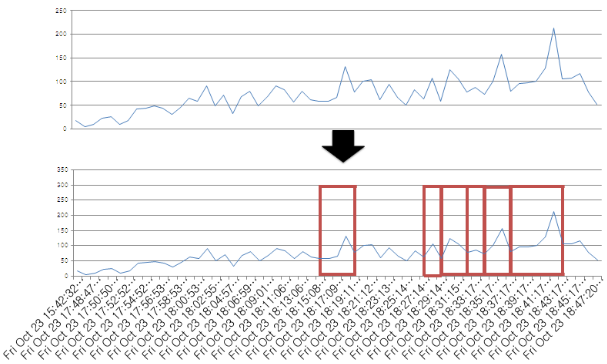

Figure 3. A Sample Result of Peaking Period Detection

4.2.2 Bursty Feature Extraction

Twitter data are normally much noisy and not formalized well, so it is hard to apply traditional TDT or keyword extraction approaches. Existing previous work of bursty feature extraction on Twitter is to exploit frequency-based term weights or utilize clustering-based term extraction [3]. However, they need many parameters or thresholds in deciding their *burstiness*, and the values would be much variable depending on the size and pattern of data. In contrast, our approach computes statistical probability that Twitter data steam is likely to contain a particular feature. Thus, we model the distribution of a feature in a peaking period by binomial distribution. Let N_w is the number of features in an peaking period, w, and n_{fre} is the frequency of f appearing in an period. Then, we compute the *burstiness* of each feature, f, $p_b(w,f;p_e)$ using the sum of probability distribution function, as in (1). It is the modification of [4] for bursty feature detection at each peaking period.

$$p_b(w,f;p_e) = \sum_{k=1}^{n_{fw}} p(k;N_w,p_e) \qquad (1)$$

$$p(k;N_w;p_e) = \binom{N_w}{k} p_e^k (1-p_e)^{N_w-k} \qquad (2)$$

$$p_e = \frac{1}{L}\sum\nolimits_{w \in W} \left(\frac{n_{fw}}{N_w}\right) \qquad (3)$$

$p(k;N_w;p_e)$ is a probability mass function, and its expected probability, p_e is calculated by (3). The expected probability denotes the average of term frequency over whole peaking periods. L means the number of peaking periods. To find out the probability that a feature is likely to be a bursty feature in a peaking period, if a function $p_b(w,f;p_e)$ is larger than 1, we can consider the feature distribution as an abnormal behavior. It means that the probability is apparently higher than the prior probability of the feature, and the feature can be selected as a bursty feature.

4.3 Episode Detection

4.3.1 Co-Occurrence- based Dynamic Relationship Detection

Relation detection is the task of identifying a relevant association between two features. Key challenges in detecting a relation include identifying the type of features and discovering temporally defined relationships at a given time period. In particular, from the dynamic nature of social media, it is hard to utilize any pre-existing schema or knowledge for understanding an episode. For example, Figure 4 describes sample relation graphs at two different temporal periods. A circle represents an individual activity, and a square means a person. They are associated with temporal episodes. Case 1 depicts that "*Park*" gets a "*strik-out*," but we cannot discover any relationship with "*homerun*" and "*Lee*." On the other hand, case 2 presents different relationships among three concepts. It is the case that "*Lee* just hit a *homerun* from *Park*." In this paper, a relation graph is constructed through co-occurrence–based dynamic relationship corroboration [1], and a type of entities to detect is restricted to baseball actions and player names. Entity detection is accomplished from matching bursty features with entries of official baseball league database.

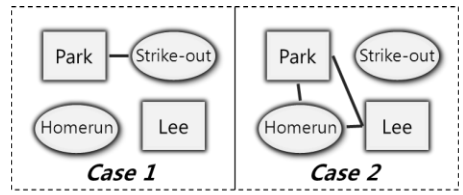

Figure 4. Sample Results of Dynamic Relation Detection

4.3.2 Topical Continuity Detection

This section describes topical continuity detection of extracted entities. When we detect an entity on social media, it is important to avoid a temporal bias for estimating correct appearance time of the entity. Thus, a topic about the entity is likely to remain over a social network even if it actually happened many hours ago. We define it as a *topical continuity* problem. For eliminating the topical continuity, our system computes cosine similarities among entities at near time periods. If the similarity is significantly large, it regards firstly appeared entity as true. The detail experiment results will be presented in our later publication.

4.4 Relation Graph-based NL(Natural Language) Text Generation

Table 2. A Sample Mapping Table for Generating a NL text

Relation	Natural Language Text to Generate
<Person>—*<Action>*	Are you curious of how *Person A* did *Action B* ?
<Person>—*<Person>*	Check out how *Person A and B* made an accident!

To provide VOD service, it is necessary to describe an episode with more expressive representations. In our proposed system, natural language texts can be generated from a mapping table, as in Table 2. For example, if there is a triple, *<Lee>* – *<Homerun>*, then a sentence, *"Are you curious of how Lee did Homerun?,"* will be generated. Therefore, each episode can have multiple summery sentences, and they are selectively utilized for VOD service.

5. EXPERIMENT

5.1 Data Set and Evaluation Method

For the evaluation of proposed framework, we crawled data of 33 baseball games from Twitter. Twitter's messages referring to a baseball game could be found at tweets marked with corresponding hash tags such as *#gotigers* and *#doosanbears*. Test collection which we crawled using Twitter API[3] consists of 33 games (8 teams) and 1761.37 tweets per a day on average. In the

[3] Java library for the Twitter API, http://twitter4j.org/en/index.html

test set, each game had 9.52 peaking periods, and a peaking period included 2.48 bursty features on average. For evaluating our system, we measure the performance of entity extraction and appearance time estimation of episodes, respectively. In order to evaluate the performance of entity detection, we measured mean average precision (MAP) score by comparing extracted features with web casts of *Korea Baseball Organization*. The web casts describe the official baseball game records including player names and activities like a hit or a homerun. For the MAP score, entitys were tagged with true, only if web casts have corresponding players and activities within near 5 minutes time window. Appearnce time estimation of episode was also evaluated by comparing the time stamp of each episode's peaking period with actual appereance at web casts. For the time estimation, we only considered episodes including entities which tagged with true.

5.2 Results and Discussion

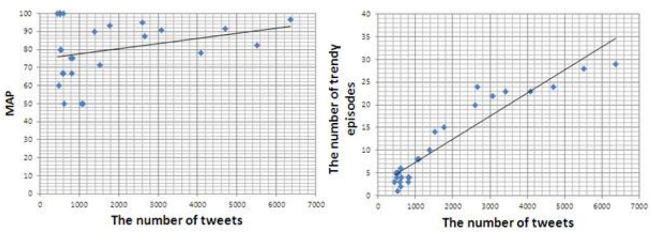

Figure 5. Influence of the Number of Tweets on Performance

Table 3. Performance of Entity Extraction and Appearance Estimation

MAP	Appearance Time Estimation Error (sec)
0.79	61.81

Table 3 suggests that our entity extraction and appearance time estimation are useful for understanding a real-time event. The MAP score of entity extraction was about 0.79, and appearance estimation error was about only 1 minute. Even though the remaining features are tagged with false, most features are still related to baseball topics such as cheer leaders and casters. In Figure 5, it is shown that how the number of tweets affects its performance. Each point represents the result of each game, and games which have lower than 400 tweets were ignored in these graphs. In the graph, the number of trendy episodes increased as more tweets are generated. At the same time, its MAP score of each game is also steadily increased after the number of tweets is more than about 1000. Thus, the descriptive power to a live event and precision are enhanced as people write more tweets.

6. PROTOTYPE OF LIVE BASEBALL WATCHING SYSTEM

Figure 6. Screen Dump of Prototype of a Live Baseball Watching System :Trendy Keywords and their Video Playback

As a prototype of intelligent VOD service on future IPTV, a live baseball watching system was implemented. It detects a trendy moment during a live event and visualizes its trendy episodes with description and relevant videos at that time, as in Figure 6. The system provides real-time trendy-keywords and highlights with relevant VODs.

7. CONCLUSION

In this paper, we propose a generic framework to discover trendy episodes for intelligent VOD service of future IPTV. This is also the first quantitative study on Twitter for *a live event* at *a very short granularity*. In the experiment, it showed much promising results and indicated that the real-time nature of micro-blog service is helpful to understand the event. In the future, our framework will be expanded into other live event domains and evaluated.

8. ACKNOWLEDGEMENT

This research is supported by Korea Institute of Science and Technology under "Development of Tangible Web Platform" project.

9. REFERENCES

[1] Sarma, A.D., Jain, A., Yu, C. 2011. Dynamic Relationship and Event Discovery, In Proc. of WSDM '11, 207-216.

[2] Shamma, D. A., Shaw, R., Shafton, P. L., and Liu, Y. 2007. Watch what I watch: using community activity to understand content. In Proc. of MIR '07, 275-284.

[3] Grinev, M., Grineva, M., Boldakov, A., Novak, L., Syssoev, A., and Lizorkin, D. 2009. Sifting micro-blogging stream for events of user interest. In Proc. of ACM SIGIR '09, 837.

[4] Fung, G.P.C., Yu, J.X., Yu, P.S., and Lu, H. Lu. 2005. Parameter free bursty events detection in text streams. In Proc. of VLDB '05, 181-192.

[5] Letierce, J., Passant, A., Breslin, J., Decker, S. 2010. Using Twitter During an Academic Conference: The #iswc2009 Use-Case, In Proc. of AAAI ICWSM '10, 279-282.

[6] Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., and Katz, G. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In Proc. of *IWCS-5*.

[7] Miyamori, H., Nakamura, S. and Tanaka, K. 2005. Generation of views of TV content using TV viewers' perspectives expressed in live chats on the web. In Proc. of ACM MULTIMEDIA '05, 853-861.

[8] Park, H., Youn, S.B., Hong, E., Lee, C., Kwon, Y.M., Ko, H., Park, M.W., Sohn, Y.T., and Kim, J.K. 2010. Sharing of baseball event through social media. In Proc. of MIR '10, 389-392.