Twitter Search Interface for Looking Back at TV Dramas

Taketoshi Ushiama
Faculty of Design
Kyushu University
Fukuoka, Japan
email: ushiama@design.kyushu-u.ac.jp

Haruka Nagai
School of Design
Kyushu University
Fukuoka, Japan
email: 1ds16408n@s.kyushu-u.ac.jp

***Abstract*—In recent years, while TV dramas are being broadcast, many comments and discussions about the dramas are posted on Twitter. These tweets are called "live tweets," and after watching a drama, users can search for live tweets about scenes of interest to them, enjoy the impressions of other viewers, and deepen their thinking from a different perspective. However, in the current Twitter search function, even if the user searches for a keyword of the target scene, the tweets including the keyword are only presented in sequential order of posting. It takes time for users to find the live tweets of the scene they are interested in. This paper proposes an interface that can efficiently look back at dramas by visualizing the similarity distribution of specific keywords by time for live tweets posted during the drama. In this paper, we propose two Word2Vec-based methods and one TF-IDF-based method to calculate the similarity between keywords and live tweets posted during segments of the drama for visualization. From the results of the evaluation experiments, we found that TF-IDF-based method is the most suitable method for calculating the similarity between keywords and situation segments for visualization. In addition, the results of a usability survey of subjects using the prototype system showed that the proposed interface was able to capture the characteristics of TV drama scenes and was an effective way to look back at TV dramas.**

Index Terms*—Twitter; social viewing; live-tweeting; TV drama; looking back.

I. Introduction

In recent years, social networking services (SNSs) have become widespread worldwide. In particular, Twitter is considered to be one of the most popular SNSs and is used on a daily basis for a variety of purposes, including the dissemination of opinions and communication.

In this context, social viewing, where people post live tweets while watching a TV program, is becoming increasingly popular. Live tweets are tweets posted while the poster is watching a TV program and include real-time reactions to the program, such as comments and opinions. By posting live tweets, SNS users can discuss the same programs with other users via Twitter, just as they normally do with their family and friends while watching TV programs [1]–[4].

Social viewing is not only fun for users who post live tweets but also for the users who only view the tweets rather than posting them. This paper focuses on live tweet searching after watching TV dramas, where viewers may want to know what others thought about a scene that left a strong impression on them or a scene that they have questions about. In such cases, they can look at the live tweets of other viewers of the scene and relate with the viewers that have similar opinions or gain new knowledge by seeing tweets with a different perspective.

Viewing live tweets can allow viewers to review the content of the drama and enjoy their reactions to the program more deeply. However, many live tweets can be posted about TV programs, and it is necessary to search through them to find the live tweets for the desired scene. This paper proposes a interface for finding the live tweets of TV dramas [1]. The term "TV drama review search" refers to the search for actual tweets for a specific scene in order to look back on the content of a drama after the initial viewing.

In the conventional Twitter search function, live tweets can be retrieved using hashtags. Hashtags are tags that begin with a "#" and classify posts by a specific topic. Many live tweets are tagged with the title of the program or its abbreviation, and hence people can search by hashtag to see live tweets posted by other people. However, whereas this search function is ideal for viewing real-time tweets about a scene being broadcast, it poses some problems when viewing past tweets, such as when the user wants to view tweets about an earlier scene after watching a TV program or when the user wants to record a TV program after it has aired. There are three problems users encounter when browsing past live tweets.

1) The number of live tweets of TV programs is huge, and it takes a lot of effort to check each result obtained by the tweet search function and to go back to the tweets of the scene that the user is interested in.
2) The contents of live tweets are often very brief. It can be difficult to tell from the tweet alone which scene the comment is about.
3) Users can also narrow down the tweets by searching for keywords that are characteristic of the target scene along with the title of the program or abbreviated hashtag, but only the tweets that match these keywords will be displayed, and hence if the keywords are ambiguous, users will not be able to obtain the tweets they want.

In this paper, we propose a tweet search interface that enables the efficient review of TV dramas to overcome these problems. This interface helps users efficiently discover live tweets of interest. In this system, the user inputs a tweet of interest, and the number of live tweets related to that keyword in the drama are visualized as a graph. Using this graph, the user can efficiently discover the time interval related to the interest and easily access the tweets of the scene the user is interested in.

The contributions of this paper are as follows:

1) we propose a user interface suitable for viewing the opinions of TV dramas posted on Twitter, and
2) we demonstrated the effectiveness of the proposed interface through user experiments.

This paper is organized as follows. Section 2 positions this research with respect to related studies. Section 3 gives an overview of the system proposed in this paper, and Section 4 describes the details of the proposed method. Section 5 shows the results of the experiments, and Section 6 presents a summary and future work.

II. Related Work

There have been many studies about TV programs and live-tweeting on Twitter.

Nakazawa et al. [5] proposed a method for detecting important scenes from tweets related to TV programs, estimating the main characters and events in each scene, and assigning them with labels representing the scenes for the efficient viewing of recorded TV programs. Lanagan et al. [6] proposed a method for identifying events of interest within the video of live sports broadcasts. Ushijima et al. [7] focused on social viewing of TV dramas using Twitter and characterized TV dramas by "development pattern" by extracting the features of scenes in the drama's chronological order using live tweets posted during the drama broadcast. Vranić et al. [8] proposed a method for extracting drama patterns from viewer responses about TV dramas posted on social networking sites.

In these studies, the features of the scene and the sentiment of the tweets were extracted and visualized based on the live tweets. In this study, we further extract the engagement for keywords entered by the user and present them in chronological order.

Tsukuda et al. [9] proposed a method for estimating the scenes in which characters in a video attract the attention of viewers and estimating the degree of activity of each character in each scene using comments posted on Nico Nico Douga. In this method, the attention-grabbing scenes are estimated by focusing only on the characters. In contrast, in the method proposed in this paper, the attention-grabbing scenes are estimated not only using the names of characters but also using the keywords entered by users.

III. Proposed Method

The purpose of this study is to develop an interface that allows users to find live tweets related to the desired scene with simple operations in order to efficiently review TV drama programs.

A. System overview

Live tweets of TV drama programs represent the real-time responses of users who are watching the drama in question. Live tweets are considered to strongly reflect the content of the scene being broadcast at that time [7]. We assume that the scenes associated with the keywords specified by the user have many live tweets with high similarity to the keywords. The relevance of the keyword to the scene is then estimated using the content of the live tweets associated with the scene. Specifically, the timeline consisting of live tweets posted during the drama broadcast time is divided into segments, and the relevance between segments and keywords is determined based on the similarity between the tweets and keywords in each segment. Then, by visualizing the transition of the relevance, users can easily find the segment they are interested in. When a user specifies a segment of interest, the user can then access the tweets contained in the segment.

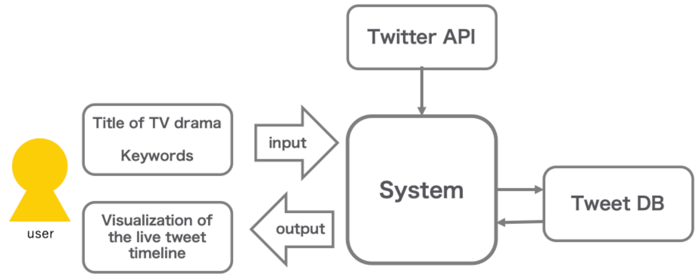

Figure 1. Overview of the proposed system.

Figure 1 shows an overview of the proposed system, and the procedure of the system is described as follows:

1) The system collects live tweets about TV drama programs using the Twitter application program interface (Twitter API). Specifically, tweets that include the title of the TV drama program hashtag posted during the broadcast time of the target TV drama program are collected and stored in the tweet database (tweet DB). Retweets and replies are excluded from the stored tweets.
2) The tweets of the TV drama program specified by the user are retrieved from the tweet DB, and the timeline of the collected tweets is divided into segments according to time in order to obtain the characteristics of the tweets over time.
3) Morphological analysis is performed on the tweets in the segment.
4) The tweets and keywords in the segment are vectorized.
5) The cosine similarities of the vectors are calculated. The similarity between each segment and the keyword is also calculated.
6) The similarity of each segment is visualized and presented to the user.

B. Modeling situations of TV dramas

The aim of the proposed method is to estimate and visualize the excitement related to keywords for each unit of time according to the progress of the TV drama program. We divide the timeline of collected live tweets into segments of a certain time interval. The set of tweets in the segmented time interval is called a situation segment, and each situation segment is considered to strongly reflect the characteristics of the scene broadcasted at that time.

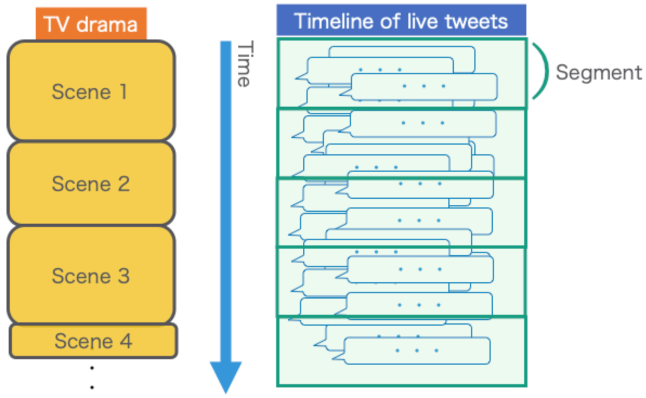

Figure 2. Relationship between scenes of a TV drama and situation segments in a timeline.

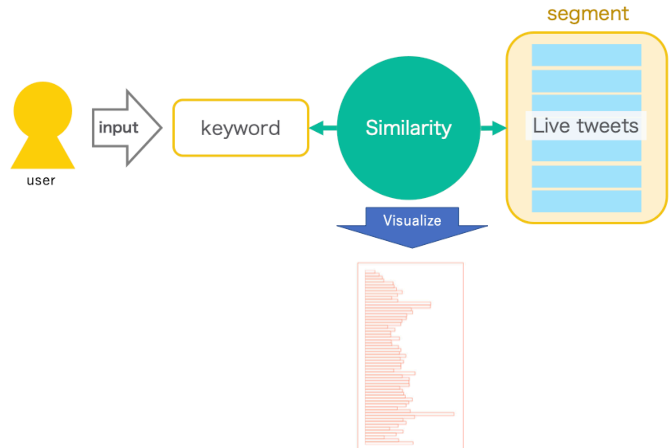

Figure 3. Visualization approach.

Figure 2 illustrates the relationship between the scenes of a TV drama and situation segments in a timeline.

We represent the timeline tl of live tweets as a series $tl = (tw_1, tw_2, \cdots, tw_n)$ using tweets tw_i. By denoting the time of posting a tweet tw as $\mathrm{time}(tw)$, any two tweets tw_i, tw_j in the timeline will satisfy $\mathrm{time}(tw_i) < \mathrm{time}(tw_j)$ if $i < j$.

This study introduces the concept of situation segmentation to describe the real-time content targeted by live tweets. A situation segment is a time interval in the targeted real-time content, which is defined as $\mathrm{s}(tl, st, et)$. Here, tl represents the target timeline, st represents the start time of the segment, and et represents the end time.

In this study, we divide the targeted real-time content into situation segments of equal length (unit situation segments) using a time window and model the features as a unit. To generate a unit situation segment, we apply a time window of length m to the real-time content, move it by $m/2$ width, and allow the windows to overlap halfway so that we can also properly model the boundaries of the segment. When a unit situation segment is defined for the target real-time content, the state of the real-time content can be represented as a series of unit situation segments. Hereafter, unless otherwise specified, the term "situation segment" refers to a unit situation segment.

For each situation segment s, we consider the corresponding live tweet series $\mathrm{TW}(s)$, which represents a subseries of the timeline targeted by the situation segment s.

C. Visualization

In this method, we provide a user interface that visualizes and displays the obtained similarity of each segment as a graph. The visualization approach is illustrated in Figure 3. The user first enters a keyword of interest q into the system. The system then calculates the similarity $\mathrm{sim}(q, s)$ of the entered query keyword q and the situation segment s in the target timeline. A single situation segment is represented in a bar graph with one horizontal bar, where the length of the bar represents the similarity. By looking at the graph, the user can determine the time the scene related to the keyword was broadcasted, and by moving the mouse over the graph, the user can view the live tweets posted at that time. The right half of Figure 4 shows an example of timeline visualization.

To calculate the similarity $\mathrm{sim}(q, s)$ between a keyword q and a situation segment s, several methods can be considered. In this paper, in Section IV, we propose three methods for computing the similarity and evaluate their performance in an evaluation experiment.

D. User interface

The proposed system provides an interface that enables users to view many tweets about scenes of interest using a visualization based on the similarity between keywords and situation segments. Figure 4 shows an example of the interface provided by the proposed system. A generated bar graph is shown on the left side of the interface. Users can click on any part of the graph, and tweets posted at the time represented by that location are displayed on the right side. The color of the background of each tweet indicates how well it matches the user's query. The closer the background is to red, the more similar the tweet is to the user's query.

IV. Computational Methods for Queries and Situation Segments

In the proposed system, the similarity between the user's query and the situation segment is calculated and used for visualization. There are several possible methods to calculate this similarity. In this section, we propose three similarity calculation methods. The performance of each method is evaluated based on the experimental results presented in Section V.

A. W2VE method

We propose the W2VE method as the first similarity calculation method. This method is based on the Word2Vec [10], which is a word vectorization method that uses a neural network consisting of two layers for text processing. By learning the weights of the neural network using a corpus,

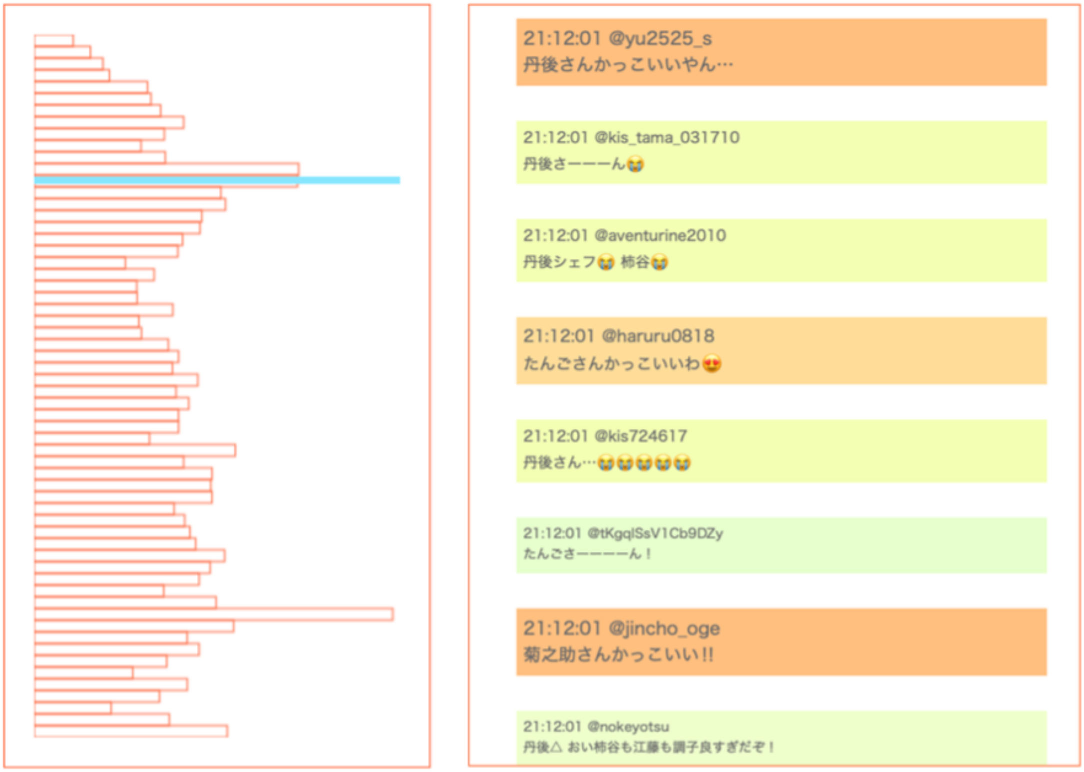

Figure 4. Screenshot of the user interface of the proposed system.

a vector representation of words can be obtained. To calculate the similarity between a situation segment and a keyword, Word2Vec is used to calculate the similarity between the query and each keyword in the segment.

In this method, the tweets and keywords in the segment are vectorized using the Word2Vec model learned by the above method, the cosine similarity with respect to the keywords is calculated for each tweet, and the average is used as the final similarity for the segment. The similarity of the W2VE method is defined as follows:

$$\mathrm{W2VE}(q,s) = \frac{1}{|\mathrm{TW}(s)|}\sum_{i\in \mathrm{TW}(s)} \mathrm{csim}\left(\mathbf{w2v}(q), \mathbf{w2v}(i)\right) \tag{1}$$

where q is the query keyword, s is the situation segment, $\mathbf{w2v}(q)$ is a function that vectorizes the query keyword q based on the Word2Vec method, and $\mathrm{csim}(\mathbf{a}, \mathbf{b})$ represents the cosine similarity between vectors $\mathbf{a}$ and $\mathbf{b}$.

B. W2VS method

We propose the W2VS method as the second similarity calculation method. The W2VS method is a calculation method that also uses the Word2Vec method. In the first method, the average of the cosine similarities of vectorized queries and tweets is obtained by Word2Vec. In contrast, in this method, the vector of the situation segment is obtained by vectorizing all the tweets in the target situation segment using Word2Vec and calculating their average. Then, the cosine similarity between the query vector and the vector of the situation segments is calculated. The W2VS method is formally defined as follows:

$$\mathrm{W2VS}(q,s) = \mathrm{csim}\left(\mathbf{w2v}(q), \mathbf{avg}(s)\right) \tag{2}$$

$$\mathbf{avg}(s) = \frac{1}{|\mathrm{TW}(s)|}\sum_{i\in \mathrm{TW}(s)} \mathbf{w2v}(i) \tag{3}$$

C. TFIDF method

The third similarity calculation method proposed in this paper is the TFIDF method. The TF-IDF [11], [12] method calculates the importance of a word in a document based on the frequency of occurrence (TF) of the word in the target document and the inverse document frequency (IDF) of the word. The TF-IDF method has been proposed in the field of information retrieval and is currently used for various purposes. In this paper, we propose a method that calculates the importance of a word in each situation segment using situation segments instead of documents in the general TF-IDF method.

The TF value of t in s is defined by the following equation, where $\mathrm{freq}(t, S)$ is the frequency of occurrence of a word t in the target situation segment s.

$$\mathrm{tf}(t,s) = \frac{\mathrm{freq}(t,s)}{\sum_i \mathrm{freq}(i,s)} \tag{4}$$

$$\mathrm{idf}(t) = \log\left(\frac{|S|}{1 + |\{s|\mathrm{freq}(t,s) \geq 1, s \in S\}|}\right) \quad (5)$$

By multiplying the TF and IDF values calculated above, the importance $\mathrm{weight}(t,s)$ of a word t in a situation segment s is defined by the following.

$$\mathrm{weight}(t,s) = \mathrm{tf}(t,s)\mathrm{idf}(t) \quad (6)$$

Using the above weights, we define the similarity $\mathrm{TFIDF}(q,s)$ between query keyword q and situation segment s as follows:

$$\mathrm{TFIDF}(q,s) = \mathrm{csim}(\mathbf{h}(q), \mathbf{w}(s)) \quad (7)$$

where $\mathbf{h}(q)$ represents the one-hot vector of the keyword q, and $\mathbf{w}(s)$ represents the feature vector of the situation segment s, which is constructed using the word weights $\mathrm{weight}(t,s)$.

V. Evaluation

This section presents the experiments we conducted to evaluate the effectiveness of the proposed method and its results. We evaluated our method with respect to the following two issues:

1) the performance of the three similarity calculation methods proposed in this paper, and
2) the usability of the proposed system.

A. Dataset, preprocessing, and prototype

The dataset used for the evaluation consists of live tweets about TV dramas collected using the Twitter API. We collected live tweets for 16 TV dramas (125 episodes) broadcasted on Japanese TV stations from July to September 2019, and a further 15 TV dramas (111 episodes) broadcasted from October to December 2019. The hashtags of the respective TV drama titles were used to collect the live tweets for the TV dramas during the broadcast times of the target dramas. Retweets and replies were excluded from these data. These tweets were written in Japanese.

Figure 5 shows an overview of the preprocessing required for this dataset. From the tweets included in the dataset, the hashtags and URLs of TV drama titles used in the collection were removed from the text because they could act as noise when obtaining the characteristics of the tweets. The other hashtags were not excluded because they can contain information such as the names of the actors in the current scene and thus become features of the scene.

All the tweets in the dataset were split into morphemes by MeCab [13], a major Japanese morphological analysis engine. For the MeCab dictionary, we used the mecab-ipadic-NEologd dictionary [14], which covers a wide range of Eigen expressions, collapsed notations commonly used on the web, and new words. Of the segmented morphemes, only nouns, verbs, adjectives, and adverbs were used, and for conjugated words, the original form of the word was used.

We implemented a prototype of the proposed system for the experiments. This system runs as a web application. PHP and JavaScript were used for its development, Apache was used as the webserver, and MySQL was used as the database management system. The Gensim library [15] was used to calculate Word2Vec, and the Twitter dataset described above was used as the corpus for training the Word2Vec model.

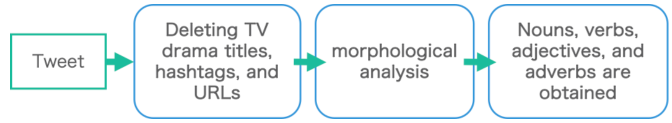

Figure 5. Overview of the preprocessing.

B. Performance comparison of the similarity calculation methods

1) Experimental method: In this paper, we proposed the W2VE, W2VS, and TFIDF methods to determine the similarity between the query keywords given by the user and the situation segments. We compared the performance of these three methods through experiments. For live tweets related to the target TV dramas, we determined the query keywords related to those TV dramas and calculated the similarity between each keyword and the situation segment. The number of target TV dramas was three. Ten query keywords were selected from each of the adjectives and nouns frequently found in the live tweets of each drama and used in the experiment. To create the ground-truth data, subjects were asked to read the tweets included in the target situation segment and give them a score from 0 to 10 on how similar their contents were to the keywords. The ground-truth data and the similarities derived by each method were normalized so that the maximum value was 1, and the error was calculated. The mean average error (MAE) was used as the measure of error.

2) Results: As an example, the results of the experiment in which the TFIDF method was used to calculate the similarity for a TV drama are shown in Figure 6. In this figure, the vertical axis represents the similarity and the horizontal axis represents the elapsed time after the start of the drama. The red line represents the calculated similarity, the green line represents the ground truth, and the blue dashed line represents the error.

The MAE values for each method are shown in Table I and the distribution of MAE for each keyword is shown in Figure 7. These results reveal that the TFIDF method yields the lowest MAE. We also analyzed whether there is a dominant difference in the MAE of each method using t-test. As a result, there was a significant difference between the W2VE and TFIDF results and between the W2VS and TFIDF results, whereas there was no significant difference between the W2VE and W2VS results. This indicates that TFIDF obtained the best performance.

C. Usability evaluation

1) Experimental method: To evaluate the effectiveness of the proposed method, we asked 20 male and 20 female users in their 20s to use the interface of the proposed method

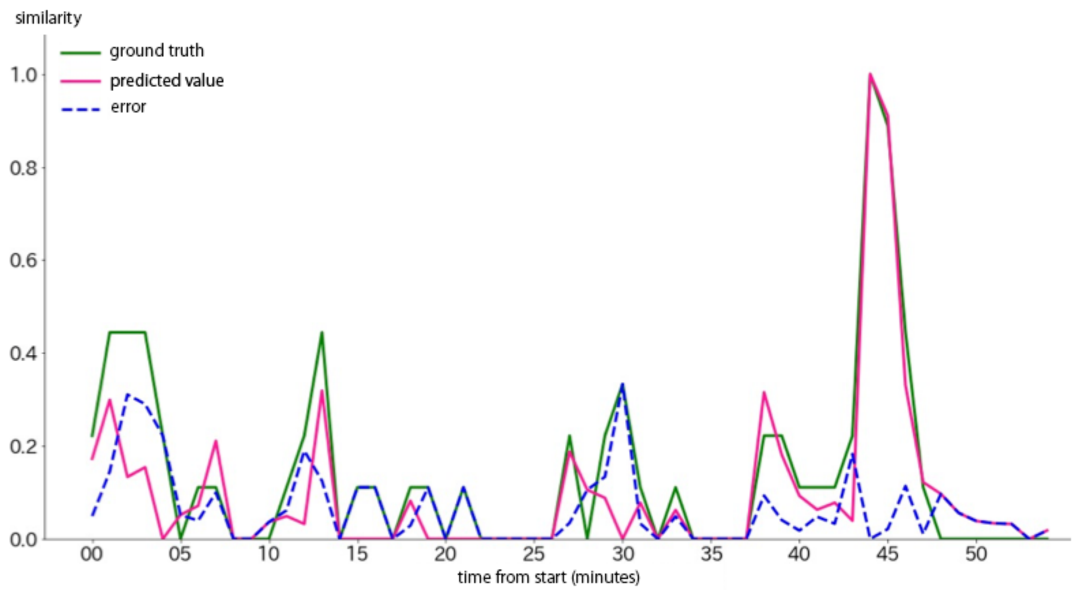

Figure 6. Example of timeline visualization.

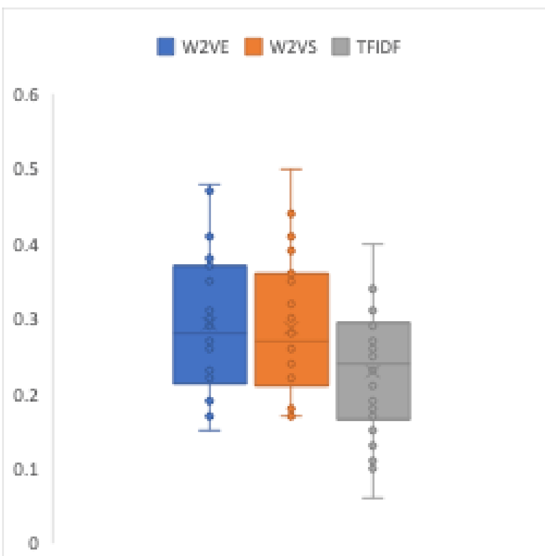

Figure 7. MAEs for the three methods.

TABLE I
AVERAGE OF MAE.

W2VE	W2VS	TFIDF
0.293	0.287	0.229

(developed using the TFIDF method) and to answer a questionnaire. The subjects were asked to enter a number of keywords for their favorite dramas, view live tweets, and answer the questionnaire. Each subject responded to each question on a five-point Likert scale from 1 to 5. 1 represents strong disagreement, and 5 represents strong agreement. The following are the questions in the questionnaire.

TABLE II
RESULTS OF THE USABILITY QUESTIONNAIRE FOR THE PROPOSED INTERFACE.

Question	Average Score
Q1	4.18
Q2	4.36
Q3	3.81
Q4	4.09
Q5	4.18
Q6	4.00

Q1: Were the graphs presented by the proposed interface able to represent the characteristics of the TV drama scenes?
Q2: Compared to browsing live tweets on a typical Twitter search interface, did you find it easier to find live tweets for scenes you were interested in using the proposed interface?
Q3: Was the proposed interface easy to use?
Q4: Was the visual appearance of the proposed interface good?
Q5: Is the proposed interface useful for looking back on TV dramas?
Q6: Would you like to use the proposed interface in the future?

2) Results: The results of the above questionnaire administered to the subjects are shown in Table II. This table shows the averages of the users' responses to each question.

For questions Q1, Q2, Q4, Q5, and Q6, the mean values were 4 or higher, which indicates that the proposed interface is an effective way to review TV dramas. The score for question Q3 is 3.81, which indicates that usability needs to be improved in the future.

VI. Conclusion and Future Work

In this paper, we proposed an interface that allows users to efficiently view live tweets for the desired scene in order to review TV dramas. The interface divides the live tweets posted during the broadcast of a TV drama into situation segments by time interval and calculates the similarity between the tweets and keywords in each segment to visualize the changes in the excitement related to the keywords of the drama. In this paper, we proposed the W2VE, W2VS, and TFIDF methods to calculate the similarity between keywords and situation segments for visualization.

From the results of evaluation experiments, we found that TFIDF is the most suitable method for this task. In addition, the results of a usability survey conducted by subjects using the prototype system showed that the proposed interface was able to capture the characteristics of TV drama scenes and was an effective approach for looking back on TV dramas.

The following is a list of issues to be tackled in the future.

1) Sometimes, a time lag exists between when a user posts a tweet and when it appears on the timeline. It will be necessary to develop a function to compensate for the user's posting time.
2) Some live tweets may contain tweets that are not directly related to the TV drama scene; we need to develop a function to filter out tweets that are not related to the TV drama content.
3) The proposed interface may be applicable to domains other than TV drama reviews. We plan to extend the interface so that it can be applied to other purposes, such as viewing public opinion on the news.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 19H04219.

References

[1] T. Ushiama and H. Nagai, "Interface for looking back at tv dramas on twitter," in *WEB 2021, The Ninth International Conference on Building and Exploring Web Based Environments*. IARIA, 2021, pp. 1–2.

[2] *EuroITV '11: Proceedings of the 9th European Conference on Interactive TV and Video*. New York, NY, USA: Association for Computing Machinery, 2011.

[3] *EuroITV '12: Proceedings of the 10th European Conference on Interactive TV and Video*. New York, NY, USA: Association for Computing Machinery, 2012.

[4] M. Doughty, D. Rowland, and S. Lawson, "Who is on your sofa? tv audience communities and second screening social networks," ser. EuroITV '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 79–86. [Online]. Available: https://doi.org/10.1145/2325616.2325635

[5] M. Nakazawa, M. Erdmann, K. Hoashi, and C. Ono, "Social indexing of tv programs: Detection and labeling of significant tv scenes by twitter analysis," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, 2012, pp. 141–146.

[6] J. Lanagan and A. F. Smeaton, "Using twitter to detect and tag important events in sports media," in *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, L. A. Adamic, R. Baeza-Yates, and S. Counts, Eds. The AAAI Press, 2011. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2821

[7] D. Minami, M. Ushijima, and T. Ushiama, "How do viewers react to drama?: Extraction of scene features of dramas from live commentary tweets," in *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, IMCOM 2018, Langkawi, Malaysia, January 05-07, 2018*. ACM, 2018, pp. 87:1–87:4. [Online]. Available: https://doi.org/10.1145/3164541.3164616

[8] V. Vranić and A. Vranić, "Drama patterns: Extracting and reusing the essence of drama," in *Proceedings of the 24th European Conference on Pattern Languages of Programs*, ser. EuroPLop '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3361149.3361153

[9] K. Tsukuda, M. Hamasaki, and M. Goto, "Smartvideoranking: Video search by mining emotions from time-synchronized comments," in *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain*, C. Domeniconi, F. Gullo, F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu, Eds. IEEE Computer Society, 2016, pp. 960–969. [Online]. Available: https://doi.org/10.1109/ICDMW.2016.0140

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 3111–3119. [Online]. Available: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[11] K. Church and W. Gale, "Inverse document frequency (idf): A measure of deviations from poisson," in *Natural language processing using very large corpora*. Springer, 1999, pp. 283–295.

[12] T. Roelleke and J. Wang, "Tf-idf uncovered: A study of theories and probabilities," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 435–442. [Online]. Available: https://doi.org/10.1145/1390334.1390409

[13] "MeCab: Yet Another Part-of-Speech and Morphological Analyzer," https://taku910.github.io/mecab/, [accessed: 2021-12-09].

[14] "mecab-ipadic-NEologd : Neologism dictionary for MeCab," https://github.com/neologd/mecab-ipadic-neologd, [accessed: 2021-12-09].

[15] "Gensim: Topic modelling for humans," https://radimrehurek.com/gensim/, [accessed: 2021-12-09].