StreamQuilt: A Timeline-Aware Integration of Heterogeneous Web Streams

Riho Nakano
Graduate School of Media and Governance
Keio University
Fujisawa, Kanagawa, Japan
e-mail: nrihos@sfc.keio.ac.jp

Shuichi Kurabayashi
Faculty of Environment and Information Studies
Keio University
Fujisawa, Kanagawa, Japan
e-mail: kurabaya@sfc.keio.ac.jp

***Abstract*— This paper proposes a timeline-aware integration system for web streams, such as micro-blogs and real-time information. This system, called StreamQuilt, analyzes implicit and temporal context-dependent relationship among heterogeneous media streams on the web. Our concept is to capture features from a media stream according to its temporal status and relationships to other streams. This system assigns different features to the stream at the different time. This system provides a synchronized streams association mechanism generate a new stream by evaluating the implicit relevance between heterogeneous streams along a timeline. This mechanism utilizes a sequence of data filters to be applied to different filters to streams. Each filter removes contextual ambiguity and various noises from media streams. This approach is advantageous in providing new information by detecting implicit relationships, such as cause-effect relationship and provider-consumer relationship, among web streams. Our system is applicable to mobile advertisement, participatory entertainment systems, and sentiment analysis of social networking services.**

Keywords-Cross-media infrastructure; heterogeneous stream association; implicit relationship analysis.

I. INTRODUCTION

Streams, such as television streams and radio streams, are among the most important media types in our daily lives. Although the number of streams in era of television and radio was hundreds at most, the popularization of broadband networks and high-performance devices has provided us with opportunities to access millions of streams nowadays via the Internet. Such rapid popularization of streams has generated a serious demand for extracting useful information and inter-stream relationships by conducting an integrative analysis of streams.

However, there are no services that integrate heterogeneous streaming media, such as television, radio, live Internet video, and live messages on social network services (SNSs). In order to compute the implicit relationships among those streaming media, it is necessary to deal with temporal changes in the contexts of the streams. To develop a novel stream management system that integrates heterogeneous streams and analyzes the integrated streams to extract useful information, two factors should be considered. The first factor is the heterogeneity in data structures of the streams. Because each stream has its own story and organization of data, it is difficult to make direct comparisons. The second factor is temporal-context dependency in relationships between streams. The meaning and position of each element are difficult to identify uniquely, because the meanings of and inter-relationships between elements in a stream should be interpreted according to its own temporal context. It is essentially a novel stream management method to integrate everyday streaming media data.

Toward the above objectives, we propose in this paper a media stream management system for both Internet media and conventional broadcast media such as television and radio. This system, called "*StreamQuilt*", provides a dynamic integration method for heterogeneous media streams. The key technology of this system is a data analysis functions for detecting implicit and temporal-context-dependent relationships between streams by removing contextual ambiguity and various noises from media streams. This method invokes a data analysis filter associated with a specific time window in order to extract a temporal-context dependent metadata from a media stream. The system then generates comparable feature sets from diverse types of data in order to find the highly inter-related fragments in different streams by evaluating correlations between metadata generated from those heterogeneous streams.

The advantage of the system is a new data provision for detecting implicit relationships, such as cause-effect relationships and provider-consumer relationships, between heterogeneous streams. Our system configures a metric space by removing irrelevant features to calculate relevance score of streams at a specific time. Each metric space consists of features related to the specific time content. This mechanism is effective for calculating the context-dependent relevance score of elements from streams. According to this context-dependent relevance score, our system integrates heterogeneous streams to create a new data stream consisting of contextually related information. This system is applicable to mobile advertisements, participatory entertainment systems, sentiment analysis on SNSs.

As an example of such implicit relationships between streams, inductive relationships are found between a television stream and an SNS stream. In this case, content broadcast on television causes various reactions in SNS communications. The system extracts such indirect effects of one stream on the other stream by triggering filters for the SNS according to information in the television program.

 ISBN: 978-1-61208-280-6

Figure 1. Screenshot of Community Generation System

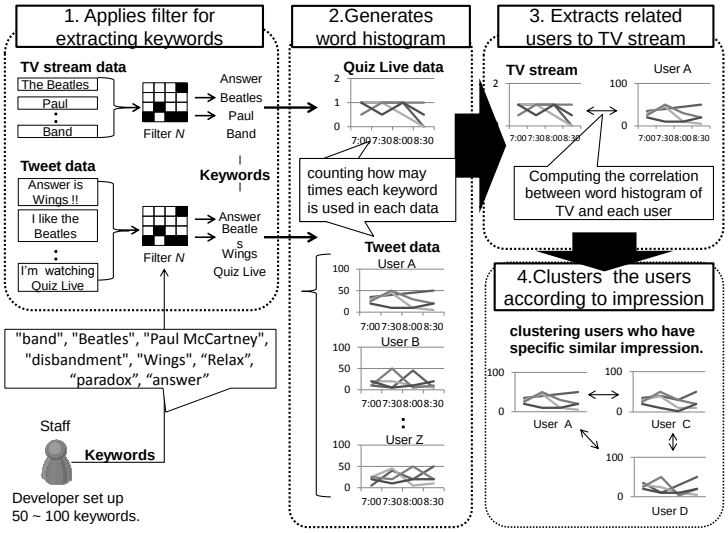


Figure 2. Concept of Community Generation System based StreamQuilt system

The information obtained about implicit relationships is very useful to plan a new television show and tie-in events on SNSs.

The remainder of this paper is structured as follows. Section II describes a temporal-context analysis of the relationship between a television stream and an SNS stream as a motivating example. Section III discusses several related studies. Section IV shows an architectural overview of our system. Section V shows the prototype of our system. Section VI evaluates the effectiveness of our system. Finally, Section VII concludes this paper.

II. MOTIVATING EXAMPLE

In this section, we present a motivating example of our StreamQuilt system. For example, there are staffs of TV program "Quiz LIVE", who want to edit program according to reactions of TV viewers during broadcasting. To get live feedback from viewers, the staffs tried to use a community generation system [1] for their Quiz LIVE. This system automatically groups viewers who have similar impression about "Quiz LIVE" as community, by analyzing SNS user messages. To realize the community generation system, the staffs can use our StreamQuilt. In this case, our StreamQuilt system extracts similarities between SNS users according to topic of TV program. This system is effective for two official positions in television station follow as.

1) Staffs who produce TV program in studio: The staffs can know name or number or member of community, through inputting Quiz LIVE data. From the information, they can judge how viewers feel, and how many viewers do so. Here, before the staffs broadcast the program, they need to set up 50~100 keyword sets to the system in each topic of Quiz LIVE. The keywords are noun or adjective for impressions. For example, there is a question "What is name of the band group that Paul McCartney formed after the Beatles disbanded?" in the program. Also, the question has three options as the answer: A)"Wings", B)"Relax", C)"Paradox". For this topic, they input "band", "Beatles", "Paul McCartney", "disbandment", "Wings", "Relax", "paradox", "answer". Figure 1 illustrates a screenshot of the community generation system. When the system starts with a broadcasting of Quiz LIVE, it visualizes community information on a view space according to the set topic in the program. Also, the system dynamically displays the program rating on twitter with using a graph. In addition, the staffs can see a transition of the information from the past to the present by dragging slider (Figure 1). While the staffs are broadcasting the program, they try to exchange the next on-air question according to interests or thoughts of viewers by checking community information.

2) Staffs who develop community generation system: When the staffs implement the community generation system, they need to suppose text data of two types as stream data: A) Quiz LIVE data is text as an annotation of the program (eg., talent, title, staff, topic of television contents). B) tweet data consists of its message and its time information and user id. The staffs design the system as follows (Figure 2). First, the system continuously reads Quiz LIVE data and tweet data. Secondly, from these data, the system extracts keywords "band", "Beatles", "Paul McCartney", "disbandment", "Wings" etc. And then, the system generates a word histogram of Quiz LIVE and each user by checking how frequently each keyword is used in Quiz LIVE data and their tweet data. Next, the system extracts users who tweeted about Quiz LIVE by computing the relevance of the word histogram of Quiz LIVE data and their tweet data. In the same way, the system is able to extract user who thinks that Paul McCartney formed "Wings" by comparing the word histogram of "Paul McCartney", "Wings" and users. In addition, the system is able to group users into several communities automatically

 ISBN: 978-1-61208-280-6

who love Beatles by computing the similarities of their word histogram.

III. RELATED WORK

In this section, we summarize the related work of our approach. Our approach focuses on the integration of heterogeneous media streams by considering their temporal-context-dependent relevance. This approach is related to the traditional data mining methods such as [2][3][4]. Those data mining methods focus on finding valuable knowledge from the large-scale homogeneous data such as relational tables. There are several studies [3][4] that apply such data mining concepts and techniques into social network data. For example, the system in [3] recommends users to each other by clustering users into several groups. The system presented in [4] finds semantic similarities between users by comparing SNS user profiles. Those approaches are effective to integrate large-scale profiles and SNS activities. However, they are not suitable to integrate heterogeneous media streams derived from various network resources such as TV, SNS, and other streaming services.

As methods for extending the conventional data mining technologies to support heterogeneous social network analysis, a behavior targeting (BT) advertisement [5] is a widely adopted approach to recommend user an appropriate advertisement by analyzing the stored user behavior data. The data analysis methods proposed in [6][7] extract a user preference by applying machine learning method and data mining methods to a user search history and/or access log to web site. And then, they compute the relevance of online-advertisement and the user preference. BT is effective to find the relevant item data by computing the relation of stored stream data and the item data. Those approaches merely use timeline of data streams because those approach focuses on finding a valuable patterns or knowledge commonly appearing on data streams.

These are several methods to find a change of relations or patterns for differing from the conventional data mining methods to find them at all time. For example, these system[8][9][10] find most frequently used words and topics and their volume of sentiments such as "positive" or "negative", from tweet on twitter according to time. These approaches are effective for detecting trend of social activity in a specific SNS by classifying similar data in chronological log of posts. These approaches are not suitable to compute the similarities among stream data in heterogeneous media.

Our StreamQuilt system differs from these existing data mining methods in the following two aspects. First, our system indirectly computes the relevance of heterogeneous stream for various type data. Here, "indirectly" means that the system compares feature values extracted from media data and doesn't compare the media data by itself in order to compute the relevance. Second, the system converts streams into feature values in same temporal range because the system analyzes multi streams which share same timeline. Concretely, the system sets particular feature sets to filter according to time window. The system extracts feature value from different streams by using same feature sets of the filter.

As an approach similar to ours, there are studies [11] [12] for clustering TV viewers on SNS or for extracting short messages related to TV from SNS. For example, Doughty et al [11] analyze audience networks by detecting a connection between users through communication about a specific TV program. A filtering method proposed in [12] computes similarities between feature keywords of Twitter message and particular program in order to collect relevant tweet. These approaches are effective for comparing constant features of TV program and tweets without timeline, by extracting features in particular time or one TV program. They are not suitable to find similarities between program and tweets in variation of feature. Our approach extracts long-term changes of features values from each stream by reducing values of irrelevant feature along with timeline. The long-term changes of features value indicate how one stream affects other stream not only immediately but also previously or subsequently. Our system is able to find stream data, which has the changes of feature values similar to other stream with them.

IV. SYSTEM ARCHITECURE

Figure 3 illustrates the architectural overview of the proposed system. StreamQuilt system consists of the three main components as follows: 1) an event-driven engine, 2) a filter module and 3) a relevance computation module. The event-driven engine invokes the filter module when the module recognizes matching of the current time and property of filters. And then, the module sends the matched filter into the filter module. The filter module generates metadata of streams by using the filter. Concretely, this module extracts specific data from streams as metadata according to feature set of the filter. In addition, this filter module sends the metadata to the relevance computation module. The relevance computation module calculates the correlation among the metadata sent by the filter module.

The most important mechanism of our system is to switch filter according to time. Our system has diverse filters corresponding to a specific time window because the system needs to extract changing different feature of stream according to time. The filters specify feature keyword as a criterion of similarity among streams. To realize this mechanism, the system provides the event-driven engine. This event-driven engine enables the filter module to extract different metadata from stream data at each time window. By using the proper metadata, the relevance computation module is able to output changing relevance of streams with time.

4.1 Data Structure

The data structure in this system consists of four data which are now explained in detail as follows.

 ISBN: 978-1-61208-280-6

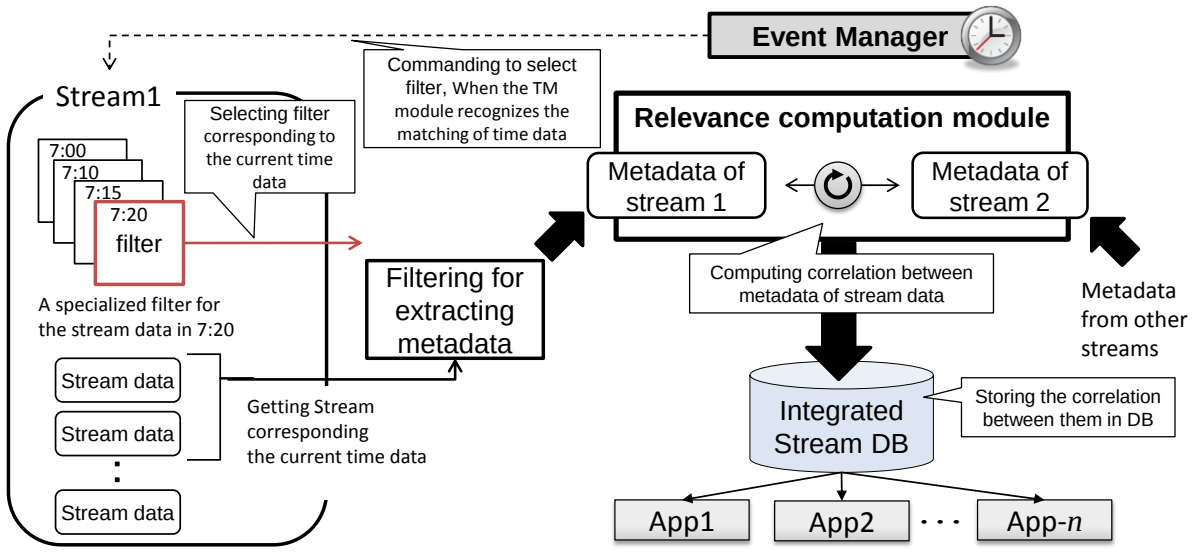


Figure 3. Architectural Overview of StreamQuilt Having an Event-driven Engine to Drive Heterogeneous Stream Integration

Snippet data s is a primary data structure in stream. The snippet data s is modeled as follows: $s := \{t, contents\}$, where s is a tuple consisting of timestamp information denoted by t and $contents$ (eg., text, image, moving image). The timestamp information t in snippet data s represents a time when the snippet data s flows into the stream.

Filter is a data for designating time window and feature set. This filter μ is modeled as follows: $\mu := \{t_s, t_e, f\}$, where t_s and t_e, which are start and end of time, are temporal range corresponding to the filter. f is feature set consisting of feature keyword and weight pairs. The weight is strength of the feature keyword in stream data. f is modeled as follows: $f := \{< k_1, w_1 >, < k_2, w_2 >, \dots, < k_n, w_n >\}$, where $k_{[1 \dots n]}$ is a feature keyword and corresponds $w_{[1 \dots n]}$.

Metadata is feature set consisting of feature keyword and value pairs. The feature keyword is the same as feature keyword of filter. A metadata a is modeled as follows: $a := \{< k_1, v_1 >, < k_2, v_2 >, \dots, < k_n, v_n >\}$, where $k_{[1 \dots n]}$ is a feature keyword and corresponds $v_{[1 \dots n]}$.

4.2 Functions

StreamQuilt system consists of two core function as follows; A) a metadata generation function and B) a relevance computation function.

Metadata generation function selects set of snippet data according to the current time in order to convert the set of snippet data into metadata. This function accepts the set of snippet data and filter and returns metadata. Metadata generation function is modeled as the equation (1).

$$f_{mgneration}(\langle s_1, s_2, \cdots s_i, s_{i+1} \rangle, \mu_j) \rightarrow a \quad (1)$$

where s_i denotes a i-th data and μ_j denotes a j-th filter. Here, The function accepted i+1 snippet data. This function computes a frequency or ratio of each feature keyword $k_{[1 \dots n]}$ in the set of snippet data. Concretely, this function counts how many times it gets m-th feature keyword from contents of set of snippet data. In addition, the function multiplies the count of m-th feature keyword and m-th weight $w_{[m]}$ in filter. This function sets the m-th product to m-th value of metadata.

Relevance computation function calculates the inner product between metadata a and a'. The function is defined as the equation (2).

$$f_{revcomputation}(a, a') := \sum_{m=0}^{n} a_{[m]} \cdot a'_{[m]} \quad (2)$$

where the top- n is the number of feature keywords in a and a', and m is the m-th feature keyword.

V. IMPLEMENTATION

In this section, we implement a prototype of the StreamQuilt system. In this implementation, our system dynamically visualizes correlations between a specific tweet stream and video according to time to play video. Figure 4 shows the detailed architecture of our prototype system. On server side, the system processes from collecting tweet data to computing the relevance scores. On client side, the system visualizes the relevance score to end-user.

The server side module consists of the following five components: stream data collector, database management system, timeline management engine, feature conversion system, relevance computation. We describe these five components in detail. The stream data collector gathers tweet data at regular intervals by twitter API. Timeline management engine gets time data to play from video elements. When the engine recognizes matching of the time to play and time scope of video filters, it starts the feature

 ISBN: 978-1-61208-280-6

conversion system. The feature conversion system consists of the following sub-components; filter selector and feature extraction engine. The filter selector switches the filters according to time scope to apply of the filters. The system computes words frequency in tweets by the selected filter in order to extract feature metadata. Also, the system uses the filter as feature metadata of video. The relevance computation calculates the correlation score between feature metadata of tweets and video. The client side program dynamically re-visualizes the correlation with graph on web browser by utilizing HTML5 and jQuery API.

The most important feature of this implementation is compatibility between it and web technology for getting tweet data and for processing them. This is because we can get stream data through existing API. Many useful API are provided by SNS such as Twitter and Facebook. In addition, we need a technology which enables to process stream data effectively in real time. So, we develop the prototype system by Node.js which is the modern server side web technology with JavaScript. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient for data-intensive real-time applications that run across distributed devices. By utilizing Node.js, the system is able to sequentially process a large amount of requests from an engine that starts an analyzing process. It is effective for the system to permit other processing to continue, before inputting or outputting of data has finished by utilizing non-blocking I/O.

VI. EVALUATION

5.1 Outline of Experimental Studies

In this section, we evaluate the effectiveness of our system when applied to TV and Twitter. For this, we implement the Stream Quilt prototype system. This evaluation experiment investigates the timeline management to filter out noisy data and compute the relevance score in each time. The experiment evaluates the precision of relevance score. Here, the precision means the number of irrelevant tweet reduced by filter in each time window. We compare our relevance computation with time manager to conventional relevance computation without time manager. We have used filters for each relevance computation: A) feature sets cover any time window; and B) the feature sets are special for each time window and contain no irrelevant features. We show that to change the feature sets makes a significant contribution to the relevance computation for stream data.

For this experiment, we configured tweets and a television program as stream data. We chose a famous music program called “Kouhaku” which is broadcast once a year. This is because we can get more tweet data about Kouhaku than tweets about other television program. In Kouhaku, each artist appeared for about 5 min. So, we gathered 1) a random set of 500 tweets by searching for the hashtag “#Kouhaku” every 5 min, 2) one feature set for

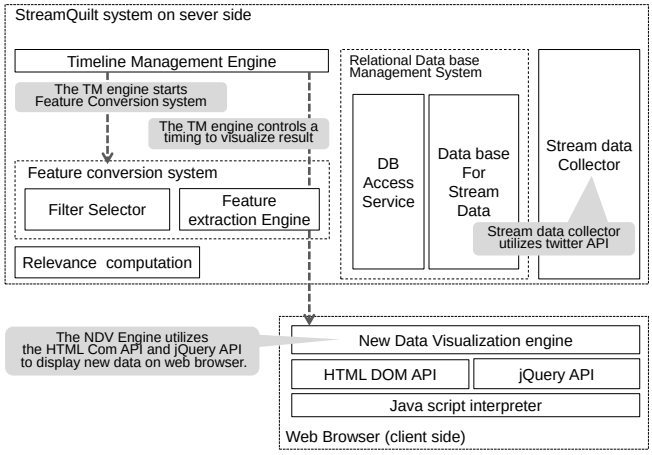


Figure 4. Prototype System Architecture of the StreamQuilt System Implemented Using Modern Technologies.

Method A, and 3) thirteen feature sets for Method B. These feature sets were formed in recognition of the basic structures of our filters. In all, we used fifteen data filters and 6500 tweets for 65 min. We employed words of nouns or adjective as feature sets for Kouhaku. We defined keywords and the values (weight) in various situations (TABLE I DEFINITION OF FEATURE SETSTABLE I). More specifically, the feature sets in method B consist mainly of three types as follow; a) keywords such as "Arashi" or "MC" appear through one hour. b) keywords such as "Opening" or "Desney" appear on only particular time. c) keywords such as "Top batter" or "character" appear according to keywords of type b, and they are related to keywords of type b. Values of these types change through one hour as in Figure 5.

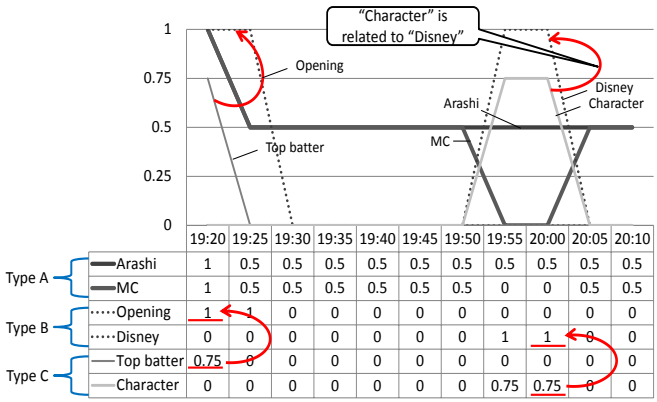
TABLE I DEFINITION OF FEATURE SETS

Value	Situation to use keyword	Example keywords
0.5	Any topic in “Kouhaku”	MC, team
0.75	Explanation of background for attention to any topic in “Kouhaku”	singer, top batter, character, Fukuoka (singer’s home)
1.0	A specific topic in “Kouhaku”	Opening, Disney

5.2 Experimental Result and analysis

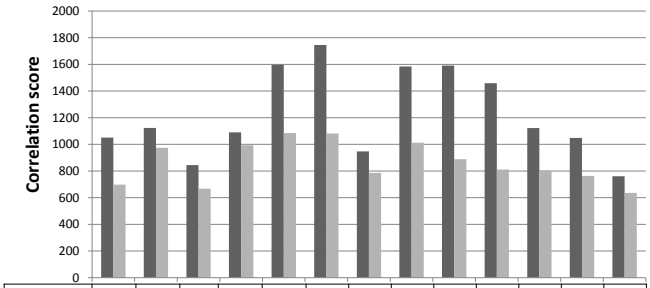
In this section, we evaluated the relevance computation in the previous subsection in order to clarify the effectiveness of our approach. Figure 6 shows the correlation between the program Kouhaku and the Twitter hashtag #Kouhaku for each approach. This result shows that score by Method B (using different feature sets at each time) is less than score by Method A (using the same feature sets at all time). It means that the relevance computation by Method B performs better than that by Method A. This is because Method B recognized more noisy tweets than Method A. Although the tweets obtained from the hashtag Kouhaku over the period are largely related to the program Kouhaku, several tweets in any specific topic are unrelated tweets. For

 ISBN: 978-1-61208-280-6



		19:20	19:25	19:30	19:35	19:40	19:45	19:50	19:55	20:00	20:05	20:10
Type A	Arashi	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	MC	1	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0.5	0.5
Type B	Opening	1	1	0	0	0	0	0	0	0	0	0
	Disney	0	0	0	0	0	0	0	1	1	0	0
Type C	Top batter	0.75	0	0	0	0	0	0	0	0	0	0
	Character	0	0	0	0	0	0	0	0.75	0.75	0	0

Figure 5. three types of feature sets in Method B



Time	7:20	7:25	7:30	7:35	7:40	7:45	7:50	7:55	8:00	8:05	8:10	8:15	8:20
Score By Method A	1051	1123	844	1090	1589	1745	947	1584	1591	1458	1122	1048	760
Score By Method B	698	974	667	993	1084	786	1012	889	889	822	804	763	636

Figure 6. Correlation between "Kouhaku" and the Twitter "hashtag Kouhaku"

example, many users tweeted about the artist "Nana Mizuki" even though that the artist did not appear in 7:25. Hence, the system needs to recognize tweets about Nana Mizuki as noisy data because those are unrelated to the topic in 7:25. This shows that our system effectively reduces noise for relevance computation by setting relevant feature sets to filter in specific time windows.

VII. CONCLUSION

We have proposed a timeline-aware integration system for web streams, is called "StreamQuilt" that evaluates the implicit relationship among heterogeneous streams. This system extracts the indirect relevance between streams by calculating correlation among comparable metadata. Here, the metadata are extracted from various data type in streams. This system uses the relevance score for creating a new data stream consisting of the contextually relevant information. The unique feature of this system is a timeline-based data analysis mechanism that applies different filter in each time window. This mechanism generated a specific metric space by using feature sets related to the time window. This mechanism is able to reduce sequentially noisy stream data by computing the distance between stream data in the metric space generated.

ACKNOWLEDGMENTS

This research was supported by SCOPE: Strategic Information and Communications R&D Promotion Programme of Ministry of Internal Affairs and Communications, Japan: "Kansei Time-series Media Hub Mechanism for Impression Analysis/Visualization Delivery Intended for Video/Audio Media.

REFERENCES

[1] R. Nakano, and S. Kurabayashi, "A Stream-Oriented Community Generation for Integrating TV and Social Network Services", The Seventh International Conference on Internet and Web Applications and Services, May. 2012, pp.286-289.

[2] W. Wu, and L. Gruenwald, "Research Issues in Mining Multiple Data Streams" Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques – StreamKDD, 2010, pp. 56-60.

[3] A. Voulodimos, C. Patrikakis, P. Karamolegkos, A. Doulamis, and E. Sardis, "Employing clustering algorithms to create user groups for personalized context aware services provision" Proceedings of the 2011 ACM workshop on Social and behavioural networked media access - SBNMA '11, 2011, pp. 33-38.

[4] C. Akcora, B. Carminati, and E. Ferrari, "Network and profile based measures for user similarities on social networks" 2011 IEEE International Conference on Information Reuse & Integration, August. 2011, pp. 292-298.

[5] D, Anjali, G. Amar, and V. Samjeev, "Data Mining Techniques for Optimizing Inventories for Electronic Commerce" Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '00, 2000, pp. 480-486.

[6] G. Xiao, Z. Gong, and J. Guo, "Personalized Scheduling Search Advertisement by Mining the History Behaviours of Users" 2009 IEEE International Conference on e-Business Engineering, October. 2009, pp. 29-36.

[7] D. Hu, Q. Yang, and Y. Li, "An algorithm for analyzing personalized online commercial intention" Proceedings of the 2nd International Workshop on Data Mining and Audience Intelligence for Advertising - ADKDD '08, 2008, pp. 27-36.

[8] K. Vinh, S. Chaitanya, R. Rajiv, and R. Jay, "Towards Building Large-Scale Distributed Systems for Twitter Sentiment Analysis" Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC '12, March. 2012, pp. 459-464.

[9] H. Wang, D. Can, A. Kazemzadeh, B. François, and N. Shrikanth, "A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle" ACL '12 Proceedings of the ACL 2012 System Demonstrations, July. 2012, pp. 115-120.

[10] A. Marcus, M. Bernstein, O. Badar, D. Karger, S. Madden, and R. Miller, "TwitInfo: Aggregating and Visualizing Microblogsfor Event Exploration" Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11, May. 2011, pp. 227-236.

[11] M. Doughty, D. Rowland, and S. Lawson, "Who is on Your Sofa? TV Audience Communities and Second Screening Social Networks", Proceedings of the 10th European conference on Interactive tv and video - EuroiTV '12, 2012, pp. 79-86.

[12] D. Dan, J. Feng, and B. Davison, "Filtering microblogging messages for social tv", Proceedings of the 20th international conference companion on World wide web - WWW '11, 2011, pp. 197-200.

 ISBN: 978-1-61208-280-6