Seeker: Topic-Aware Viewing Pattern Prediction in Crowdsourced Interactive Live Streaming

Cong Zhang*, Jiangchuan Liu*, Ming Ma**, Lifeng Sun**, Bo Li†
*School of Computing Science, Simon Fraser University, BC, Canada
**Department of Computer Science and Technology, Tsinghua University, China
†Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, China
{congz, jcliu}@cs.sfu.ca, {mm13@mails., sunlf@}tsinghua.edu.cn, bli@cse.ust.hk

ABSTRACT

Recently, Crowdsourced Interactive Live Streaming (CILS), such as *Twitch.tv* and *Periscope*, has emerged as one of the most popular streaming applications over the Internet. In such applications, a large number of geo-distributed users publish live sources to broadcast their game sessions, personal activities, and other events, while fellow viewers not only watch these live streams, but also contribute interactive messages to influence streaming content. Such explosively increasing popularity has posed significant challenges to predict viewing patterns using traditional time-series approaches, which lack the start/end knowledge of live streams and cannot capture the viewing burst very well.

In this paper, we closely examine the characteristics of interactive messages in the real-world datasets, we find that the strong topic relevances exist in the viewers' discussions. Motivated by this observation, we design a crowdsourced framework `Seeker` to overcome aforementioned challenges. It explores the correlation between three viewing patterns (i.e., start/burst/end of live streams) and viewers' interactive messages (even before a live broadcast) through capturing the key topics. Our trace-driven evaluation and case study show the effectiveness of our solution, which can predict aforementioned patterns in advance and achieve much higher performance than the time-series approaches.

CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Computing methodologies →** *Topic modeling*;

KEYWORDS

Crowdsourced interactive live streaming, viewing patterns, interactive messages, Twitch.tv

ACM Reference format:
Cong Zhang, Jiangchuan Liu, Ming Ma, Lifeng Sun, Bo Li. 2017. Seeker: Topic-Aware Viewing Pattern Prediction in Crowdsourced Interactive Live Streaming. In *Proceedings of NOSSDAV'17, Taipei, Taiwan, June 20-23, 2017,* 6 pages. http://dx.doi.org/10.1145/3083165.3083179

NOSSDAV'17, June 20-23, 2017, Taipei, Taiwan
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5003-7/17/06…$15.00
https://doi.org/http://dx.doi.org/10.1145/3083165.3083179

1 INTRODUCTION

Nowadays, Crowdsourced Interactive Live Streaming (CILS) platforms, such as Twitch.tv[1] (or Twitch for short) and Periscope[2] have shifted the role of Internet user from a traditional passive audience to a content broadcaster (i.e., crowdsourcer). These broadcasting amateurs, unlike the professional content providers (e.g., American Broadcasting Company), launch their personal channels to perform daily shows or interactive game sessions conveniently. Taking Twitch as an example, one earlier report published by Wall Street Journal showed that, as of February 2014, Twitch accounted for 1.8 percent of peak Internet traffic in the US, driving the fourth most traffic, and more than 10 thousand broadcasters perform game sessions and attract 0.5 million concurrent viewers at the peak time; in August 2015, this number has increased to 25 thousand broadcasters and 2.1 million audiences. Another red-hot live-streaming application Periscope also exhibits such similar trends [6]. Besides the growing popularity, our previous measurement shows that the crowdsourced instinct and interactive behavior lead to more dynamic workloads both in the streaming ingest and content distribution in this crowdsourced multimedia paradigm [15].

There have been pioneering works on jointly modeling and predicting the characteristics of viewers in the Video-on-Demand services [1, 12] (e.g., YouTube) and Peer-to-Peer streams [9, 13] (e.g., PPTV). These studies design the centralized control components to collect the information of streams, including duration, bitrate, and historical viewing patterns, before scrutinizing how to predict the popular streams and allocate resources efficiently. CILS platforms, however, show several distinct features: (1) the duration is unknown due to the broadcasters' unpredictable activities; (2) the live streams usually generate the frequent flash-crowd (i.e., live burst); (3) the limited historical information cannot be used to predict the details of viewing pattern in real-time. All of these make the viewing pattern prediction challenging.

In this paper, we explore a novel design considering the crowdsourced features in CILS platforms. That is, the viewers' interactive messages provide active feedbacks and discussions, which in turn creates the unique opportunities for investigating the viewing patterns even before a live broadcast. Our measurements further show the strong semantic similarity between these messages. Motivated by this observation, we propose `Seeker`, a topic-aware viewing pattern prediction framework, to capture the dynamics of viewers in live streams, i.e., start/burst/end stages, through jointly monitoring

[1] www.twitch.tv, owned by Amazon.com in September, 2014.
[2] iOS/Android app, acquired by Twitter in January, 2015.

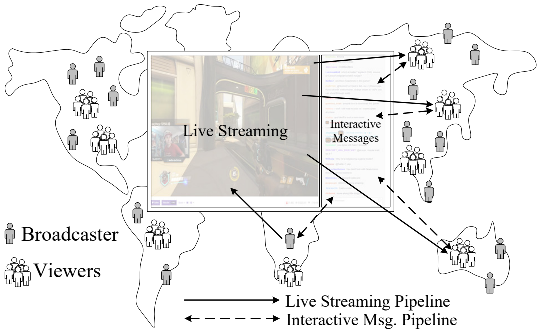


Figure 1: Illustration of CILS paradigm

and analyzing crowdsourced interactive messages. The trace-driven evaluation shows that Seeker can get the audience's patterns in advance and achieve a higher prediction performance than the time-series approaches. To the best of our knowledge, this is the first paper that closely examines the interactive messages in CILS platforms, and further studies the correlation between these messages and the viewing patterns.

2 BACKGROUND AND MOTIVATION

In this section, we provide an overview of CILS platforms and introduce the details of our datasets. Such commercial CILS systems as Twitch, Periscope, and YouTube Gaming, have attracted an increasing number of broadcasters and viewers. Figure 1 depicts the generic diagram of a CILS platform with the live streaming and interactive message pipelines that jointly serve the geo-distributed broadcasters and viewers. The live sources are managed by these broadcasters and driven by a large number of viewers in real-time using interactive messages (e.g., in TwitchPlaysPokemon[3]). Since the sources are controlled by the non-professional broadcasters, issues like streaming quality and smoothness are hardly controlled at a certain level. Moreover, given the diversity of the broadcasters' activities, the effective prediction of viewing patterns becomes more challenging. Therefore, we aim to explore a more effective method to solve these issues. More specifically, we exploit the semantic feature of crowdsourced interactive messages to build the relationship between the viewing patterns and interactive discussion in CILS platforms.

In our study, the stream data was crawled from Twitch every five minutes in five weeks from Jan. 25th to Feb. 28th, 2015. Through the Twitch APIs[4], our online crawler obtained information from each broadcaster. The message data used in this study was collected using a client-side Internet Relay Chat (IRC) crawler to connect Twitch channels and gather interactive messages. We retrieved the stream dataset and interactive message dataset through polishing aforementioned data. We now give a brief explanation:

- in Stream dataset: each trace records the number of viewers every five minutes and other properties including the start time, duration, broadcaster's channel ID, etc.
- in Interactive Message dataset: each trace collects the interactive messages of viewers during a live broadcast and includes the time, viewer's ID, broadcaster's channel ID, and message content.

To understand the viewers' characteristics, we closely examine the stream dataset. Figure 2 shows the viewing patterns in two representative broadcasters' channels. As shown in this figure, first, the live duration either is extremely long without any interruption (in A's streams) or shows irregular start time and end time (in B's streams); second, the number of viewers increases suddenly at the initial stage of live streams and disappeared when the streams are terminated (in B); third, the frequent flash crowds generate dynamic viewing patterns, which cannot be predicted timely (in A and B). Therefore, using historical information to predict the viewing patterns does not work in the current scenario. Figure 3 indicates the distribution of the duration in Stream dataset. This figure shows that the durations of 50% of live streams are more than 50 minutes, which is longer than that in the traditional streaming services (e.g., 700 seconds in YouTube [5]). Because the duration mainly depends on the broadcasters' activities and preferences, the real-time prediction of viewing patterns is challenging.

In this paper, we highlight the following three types of viewing patterns: (1) the start of live streams, referred to as Live Start or S1 for convenience, which indicates the initial resource occupation; (2) the burst of viewers' requests, referred to as Live Burst or S2, which shows the huge rise of the number of viewers and will trigger the increasing requirements in computing and bandwidth resources for the service providers; (3) the end of live streams, referred to as Live End or S3, which means that the ingesting, transcoding and distribution resources will be released suddenly in a short time. Figure 4 illustrates our basic concept, that is, we can predict the viewing patterns through exploring the semantic features of their interactive messages. For example, several viewers first enter into live channel based on the broadcaster's announcement in online social networking services, e.g., Twitter, and wait for the start of this live stream. Then, some of them may say "Hey" or other greeting words. The black circles and the corresponding messages on the timeline in Figure 4 also indicate that we could "capture" the various viewing patterns through exploiting viewers' discussions.

To provide insights into viewers' interactive messages, we also analyze the interactive message dataset. Figure 5a shows that the number of terms[5] in 90% of messages is less than 10. More specifically, 39% of messages only have one term, which indicates that we have to integrate the interactive messages in a time-slot into one document instead of exploring topics for every message [11]. Another interesting finding is that the numerous viewers in Twitch prefer to use emoticons[6]. We further find that the mixture of emoticons and words accounts for a huge amount of messages. The total percentage of mixture messages is about 29% in the whole dataset. These mixture messages are contributed by 0.73 million viewers, that is, more than a third of viewers utilize emoticons to interact with others during live streams. To closely understand the utilization of emoticons, we plot the distribution of emoticons in mixture

[3] http://www.twitch.tv/twitchplayspokemon
[4] http://dev.twitch.tv/
[5] In this paper, we split every message into terms by the "space" character.
[6] The emoticon in Twitch is the pictorial representation of viewer's emotion, the full list can be accessed from https://twitchemotes.com/

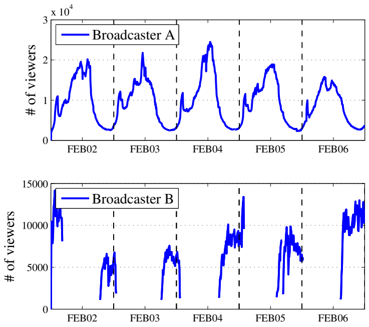


Figure 2: Sample viewing pattern.

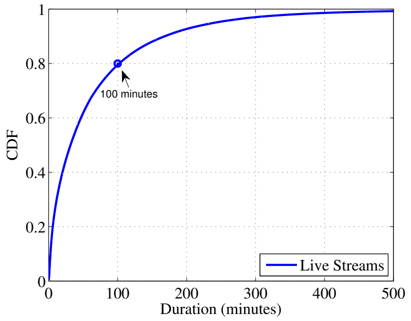


Figure 3: The distribution of duration.

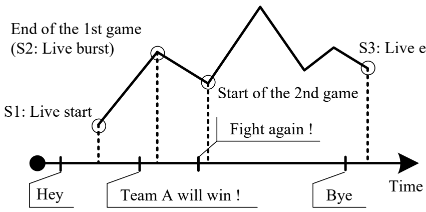
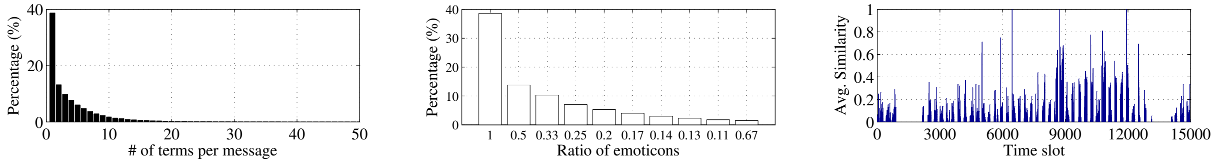


Figure 4: An example of motivation.



(a) Distribution of terms

(b) Percentage of emoticons

(c) Avg. similarity per time-slot

Figure 5: Characteristics of words/emoticons in interactive messages.

messages in Figure 5b. We observe that 38.5% of mixture messages only have emoticons without any words, then the 13.8% (*resp.* 10.3%) of mixture messages include 50% (*resp.* 33%) of emoticons. These findings suggest that the use of emoticons is helpful for exploring the topics. Moreover, we examine the average Jaccard Similarity[7] of messages in neighboring time-slots in Figure 5c. This figure demonstrates that the diverse average similarities along with the time-slot (one minute) and reveals that the messages in one time-slot show the strong topic preference.

3 TOPIC-AWARE VIEWING PATTERN PREDICTION FRAMEWORK

In this section, we present the design of Seeker based on the topic model [3]. We first illustrate the basic components of Seeker, then propose the EMoticon-aware Topic Model (EMTM).

3.1 Workflow

The framework Seeker has four components: Data Parsing, Model Training, Topic Extraction, and Pattern Prediction. Its workflow is shown in Figure 6.

Data Parsing: The interactive messages in one minute are considered as one document. To fit the topic model, we introduce the bag-of-words model[8] to transform these messages based on the word/emoticon mapping and frequency. For the historical interactive data, we also use this method to process them as the training data in our following topic model.

Model Training: We use the historical interactive messages to train our EMTM model before the topic extraction. This process is independent for each broadcaster. To capture the viewing features, we train the model using the historical traces. In this stage, we not only train the model parameters, but also explore the topic features of specific viewing patterns.

Topic Extraction: We propose the EMTM model to extract the topics of messages per time-slot. The EMTM has two generative processes for words and emoticons, respectively. We will provide the details of the EMTM model in the next subsection. In this paper, we only discuss the offline training and extraction, which can be extended to online version easily.

Pattern Prediction: After extracting the corresponding topics, this module gives the pattern prediction result according to the topic features in the model training process. We illustrate the details in Section 4.

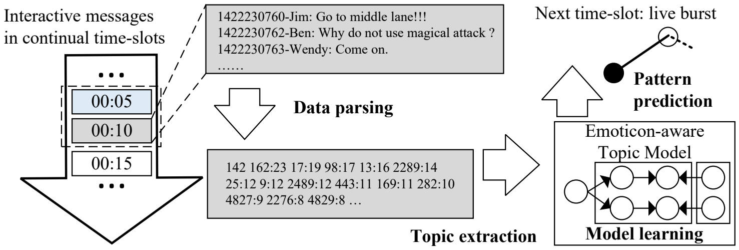
3.2 Emoticon-aware Topic Model

Topic model is a suite of statistical models that analyze the words of the original documents to extract the topics without any prior annotations or labeling of these documents. In this paper, the EMTM model is based on the Sparse Topical Coding (STC [16]), which matches the short mixture of texts/emoticons and the sparse feature in word space. Given the importance of emoticons, we extend the original STC model to the emoticon-aware topic model through appending the emoticon generative process, which introduces an additional emoticon space and increases the sparsity. In this subsection, we mainly illustrate its generative process.

In the EMTM model, we use $\mathcal{V}$ and $\mathcal{E}$ to denote the sets of text words and emoticons respectively and assume that the a document d contains the interactive messages in one minute, $d \in D$. Each time-slot t (five minutes in this paper) contains $|D|$ documents, $t \in [1, T]$. Each document $d = \{\boldsymbol{w}_d, \boldsymbol{c}_d\}$ consists of two parts: $\boldsymbol{w}_d$ is the word occurrences vector for each word in d, $\boldsymbol{w}_d = [w_{d1}, w_{d2}, \ldots, w_{d|M|}]^T$, where M is the index set of words in d;

[7] Jaccard Similarity [8] is a statistic used for comparing the similarity of sample sets. $Jaccard(A, B) = \frac{|A \bigcap B|}{|A \bigcup B|}$

[8] https://en.wikipedia.org/wiki/Bag-of-words_model



1422230760-Jim: Go to middle lane!!!
1422230762-Ben: Why do not use magical attack ?
1422230763-Wendy: Come on.
……

142 162:23 17:19 98:17 13:16 2289:14
25:12 9:12 2489:12 443:11 169:11 282:10
4827:9 2276:8 4829:8 …

Figure 6: Workflow of the framework Seeker.

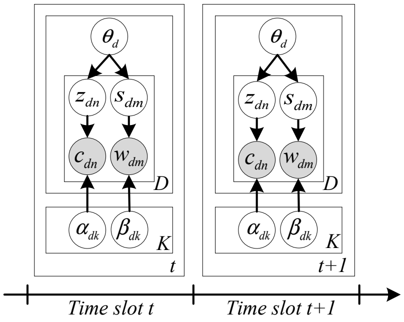


Figure 7: A graphical illustration for the EMTM model.

$\boldsymbol{c}_d = [c_{d1}, c_{d2}, \ldots, c_{d|N|}]^T$ representing the emoticons occurrences vector for each emoticon in d where N is the index set of emoticons.

Figure 7 illustrates the graphical model of EMTM, where $\boldsymbol{\theta}_d$ is the document code (i.e., coefficient vector) of document d, $\boldsymbol{s}_{dm}$ is the word code of word m, and $\boldsymbol{z}_{dn}$ is the emoticon code of emoticon n. We use $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ to denote the matrices of K topic bases for each text words and emoticons respectively. To reflect the generative differences between words and emoticons in documents, as shown in Figure 7, we first sample topic distributions $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ from uniform distribution on $\mathcal{P}_{|\mathcal{V}|-1}$ and $\mathcal{P}_{|\mathcal{E}|-1}$, respectively. We then propose the following generative process[9] for each document d in time-slot t:

1. sample the document code $\boldsymbol{\theta}_d$ from a prior $p(\boldsymbol{\theta})$.
2. for each observed word $m \in \mathcal{M}$
   (a) sample the word code $\boldsymbol{s}_{dm}$ from a conditional distribution $p(\boldsymbol{s}|\boldsymbol{\theta}_d)$
   (b) sample the observed word count w_{dm} from a distribution $p(w_{dm}|\boldsymbol{s}, \boldsymbol{\beta}_k)$.
3. for each observed emoticon $n \in \mathcal{N}$
   (a) sample the emoticon code $\boldsymbol{z}_{dn}$ from a conditional distribution $p(\boldsymbol{z}|\boldsymbol{\theta}_d)$
   (b) sample the observed emoticon count c_{dn} from a distribution $p(c_{dn}|\boldsymbol{z}, \boldsymbol{\alpha}_k)$.

This generating process defines the following joint distribution for each document in a time-slot t.

$$\begin{aligned} &p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \\ &p(\boldsymbol{\theta}) \prod p(\boldsymbol{s}_m|\boldsymbol{\theta}) p(w_m|\boldsymbol{s}_m, \boldsymbol{\beta}) \prod p(\boldsymbol{z}_n|\boldsymbol{\theta}) p(c_n|\boldsymbol{z}_n, \boldsymbol{\alpha}) \end{aligned} \tag{1}$$

To fulfill the sparsity of model, we keep the distribution settings about document code and word code in original STC and extend the settings to emoticon code. Moreover, the distributions in this model belong to the experiential family, which benefits to the following maximum a posteriori (MAP) estimation and optimization.

In step 1, we define the prior distribution of document code $\boldsymbol{\theta}$ as Laplace prior distribution.

$$p(\boldsymbol{\theta}) \propto exp(-\lambda \parallel \boldsymbol{\theta} \parallel_1)$$

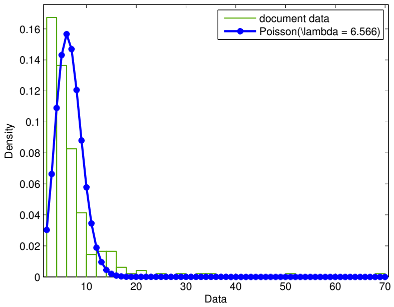


Figure 8: Distribution of word count in one time-slot.

In step 2, we define the conditional distribution of word code $\boldsymbol{s}$ and emoticon code $\boldsymbol{z}$ as the following composite distribution.

$$p(\boldsymbol{s}_m|\boldsymbol{\theta}) \propto exp(-\gamma_1 \parallel \boldsymbol{s}_m - \boldsymbol{\theta} \parallel_2^2 - \rho_1 \parallel \boldsymbol{s}_m \parallel_1)$$

$$p(\boldsymbol{z}_n|\boldsymbol{\theta}) \propto exp(-\gamma_2 \parallel \boldsymbol{z}_n - \boldsymbol{\theta} \parallel_2^2 - \rho_2 \parallel \boldsymbol{z}_n \parallel_1)$$

In step 3, based on the observation of word count in one time-slot, as shown in Figure 8, we define the conditional distributions of word count $\boldsymbol{w}$ and emoticon count $\boldsymbol{c}$ as Poisson distribution with the mean parameters $\boldsymbol{s}_m^T \boldsymbol{\beta}_{\cdot m}$ and $\boldsymbol{z}_n^T \boldsymbol{\alpha}_{\cdot n}$, respectively.

$$p(w_m|\boldsymbol{s}_m, \boldsymbol{\beta}_k) \propto Poiss(\boldsymbol{w}_m; \boldsymbol{s}_m^T \boldsymbol{\beta}_{\cdot m})$$

$$p(c_n|\boldsymbol{z}_n, \boldsymbol{\alpha}_k) \propto Poiss(\boldsymbol{c}_n; \boldsymbol{z}_n^T \boldsymbol{\alpha}_{\cdot n})$$

where

$$Poiss(\boldsymbol{w}_m; \boldsymbol{s}_m^T \boldsymbol{\beta}_{\cdot m}) = \frac{(\boldsymbol{s}_m^T \boldsymbol{\beta}_k)^{\boldsymbol{w}_m}}{\boldsymbol{w}_m!} exp(-\boldsymbol{s}_m^T \boldsymbol{\beta}_{\cdot m})$$

$$Poiss(\boldsymbol{c}_n; \boldsymbol{z}_n^T \boldsymbol{\alpha}_{\cdot n}) = \frac{(\boldsymbol{z}_n^T \boldsymbol{\alpha}_k)^{\boldsymbol{c}_n}}{\boldsymbol{c}_n!} exp(-\boldsymbol{z}_n^T \boldsymbol{\alpha}_{\cdot n})$$

We use the standard MAP estimation to determine the parameters in joint distribution (1) through finding the most probable values of w and c given the training set $\{\boldsymbol{c}_d, \boldsymbol{w}_d\}$. The method of MAP estimation can be used to obtain a point estimation of the posterior based on observed data. Non-negative hyper-parameters $(\lambda, \gamma_1, \gamma_2, \rho_1, \rho_2)$ will be selected using the cross-validation approach. We omit more details here due to the space limitation, which can be found in [14].

[9] To simplify the notation, we omit the time index t.

4 EXPERIMENTS

In this section, we first depict the experimental settings to test and validate our proposed model with two state-of-the-art topic models, i.e., Latent Dirichlet Allocation (LDA) [3] and Correlated Topic Model (CTM) [2]. We also propose topic correlation (TC) as the metric to evaluate the prediction performance for the characteristics of viewing pattern. In addition, we conduct a case study to demonstrate the effectiveness of the framework Seeker in CILS scenario.

4.1 Experimental Setup

The interactive messages are combined into different documents by their *timestamp*. Each time-slot t contains $|D|$ documents (i.e., $0 \leq |D| \leq 5$). The set of words $\mathcal{V}$ is changed in different stream, while the set of emoticons $\mathcal{E}$ is fixed based on the unofficial collection[10], $|\mathcal{E}| = 39582$. The number of topics K will be examined from 2 to 20.

We predict three viewing patters: S_1: live start; S_2: live burst; S_3: live end. As such, the documents in all time-slots are classified into these three categories. We implement the EMTM model based on the original STC[11]. For our test system, we use a desktop PC with an Intel i7 3770 3.4 GHz quad core CPU, 16GB 1600MHz DDR3 ram, a 256GB solid-state disk and a 2TB 7200 RPM hard drive. We compare the performances of state-of-the-art topic models (LDA/CTM) with our EMTM model. LDA model assumes that the terms of each document arise from a mixture of topics, each of which is a distribution over the vocabulary, while CTM explicitly models the correlation between the latent topics in the collection[12].

4.2 Performance Evaluation

We first use held-out log-likelihood [2] to evaluate three methods. The better model will assign a higher probability to the held-out data. Each model is fitted with various numbers of topics based on a smaller collection of message data. This collection includes three types of documents in different live stages: S_1, S_2, and S_3. Every type of document contains viewers' interactive messages in forty minutes and is divided into two twenty-minute collections (training data and held-out data) for training model and computing held-out log-likelihood. Figure 9 illustrates the performance of the EMTM model, which has a higher held-out log-likelihood in all three stages, especially in the stage S3. The EMTM model shows the better performance for the short messages with words and emoticons. The difference of held-out log-likelihood among S_1, S_2, and S_3 in each method depends on the number of words/emoticons in each document. For example, stage S_1 has fewer messages as compared to S_2 and S_3.

We further evaluate the Seeker with the data traces of top-10 broadcasters in the datasets. We first train the EMTM model using the messages from 2015-Jan-25 to 2015-Jan-31. The training traces are divided into three stages defined in previous sections. Because previous work in [4] presented held-out log-likelihood do not capture whether topics are coherent or not based on human experiments, we do not use this metric to predict the viewing patterns directly. On the other hand, we observe that the learned topics exhibit several unique features. In Stage S_1, most of the topics are about the waiting discussion; but several topics show that some viewers prefer to send related links (e.g., broadcaster's Facebook) in their interactive messages. While the meaning of topics in stages S_2 and S_3 cannot be understood reasonably because the interactive messages include lots of emoticons (e.g., kappa/nb3wc), match information(e.g., game player IDs and players' team name) and game slang (e.g., gg: good game), which is different from the topics of a article (e.g., human can easily understand the result of topic words in $\{apple, orange, banana\}$). To overcome this issue, we present a new metric, called Topic Correlation (TC), to evaluate the degree of correlated topics between two time-slots and predict the dynamic patterns. Let A_k^t denote the words set of topic i in time-slot t and B_k^n denote the terms set of topic j in stage S_n of model learning. We extend Jaccard Similarity to calculate TC^t in time-slot t through the following equation.

$$TC_n^t = \sum_{i=1}^{K} \sum_{j=1}^{K} \frac{|A_i^t \bigcap B_j^n|}{|A_i^t \bigcup B_j^n|}, \forall n \in 1, 2, 3$$

Therefore, we can use the EMTM model to discover the topics of time-slots and then calculate the topic correlations with training data in each stage. Based on the topic correlations, we acquire the prediction of viewing pattern for each time-slot.

Algorithm 1 Pattern Prediction

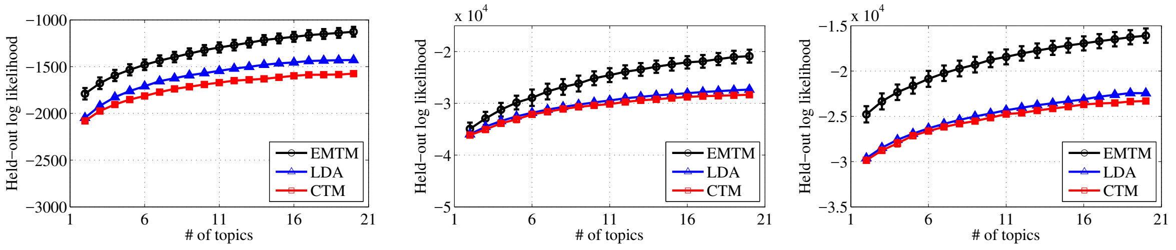
```
Input:
    Topic number $K$;
    Topic sets $B^n, \forall n \in 1, 2, 3$;
    Message traces $D_t$ in time slot $t$;
    Views number $N_v$;
    Prediction thresholds $H_n, \forall n \in 1, 2, 3$;
Output:
    Prediction Results ($Stage$, $Interval$)
 1: if $N_v == 0$ AND $|D_t| > 0$ then
 2:     if $TC_1^t > H_1$ then
 3:         return $S_1$
 4:     end if
 5: else if $N_v > 0$ AND $|D_t| > 0$ then
 6:     if $TC_2^t > H_2$ then
 7:         return $S_2$
 8:     else if $TC_3^t - TC_2^t > H_3$ then
 9:         return $S_3$
10:     end if
11: end if
```

Algorithm 1 illustrates the prediction process in one time-slot. We define several thresholds, $H_n, \forall n \in 1, 2, 3$, to determine which stage can be assigned to next time-slot. For the different broadcasters, we use the decision tree method [7] to adjust these thresholds based on the training results. In this algorithm, step 1-4 indicate that viewers are waiting for the start of live streams and returns a prediction result about live start, step 5-10 indicate that the broadcaster already starts live streaming and return a prediction result about live burst/end. If there is no any output, the algorithm will wait for a prediction interval and then start to predict the stage

[10] https://twitchemotes.com/apidocs

[11] We modify *medSTC* package from R-project (https://cran.r-project.org/web/packages/medSTC/index.html)

[12] We use the implementations of LDA/CTM in *topicmodels* package from R-project (https://cran.r-project.org/web/packages/topicmodels/index.html).



(a) S_1: Live start (b) S_2: Live burst (c) S_3: Live end

Figure 9: Held-out log-likelihood of three models in three stages.

Table 1: Comparison of three approaches

	Recall			Precision			F-score		
Stage	Seeker	ARIMA	MLR	Seeker	ARIMA	MLR	Seeker	ARIMA	MLR
S_1	0.88	X	X	0.78	X	X	0.83	X	X
S_2	0.76	0.32	0.60	0.81	0.71	0.79	0.78	0.45	0.69
S_3	0.56	X	X	0.48	X	X	0.52	X	X

of time-slot $t + 1$ based on the latest message collection. We use ARIMA [9] and MLR [10] as comparisons, these two methods only can be used to prediction S_2 based on the historical number of viewers. Table 1 exhibits the Recall, Precision, and F-score [7] in three stages[13]. Because we only consider TC_1 before the live stream, the F-score of S_1 is higher than others. The prediction of S_3 is disturbed by the topic correlation of S_2, therefore, the performance in this stage is the lowest.

5 CONCLUSION

In this paper, we consider the viewer’s interactive messages in the problem of predicting the viewing patterns in Crowdsourced Interactive Live Streaming (CILS) platforms. By exploring Twitch’s stream dataset and interactive message dataset, we have demonstrated that traditional methods entirely based on the historical information have several limitations. For example, they cannot predict the start of a live stream. To solve these challenges, we presented a topic-aware viewing pattern prediction framework Seeker, which investigates the relationship between the viewing patterns and interactive messages. We designed the emoticon-aware topic model to highlight the importance of emoticons in the interactive messages. we also proposed a new metric Topic Correlation (TC) to calculate the topic relationship between training data and test data. The results in case study show that Seeker not only can be used to predict the viewing start/end of live streams, but also achieves much higher performance than the time-series approaches in the prediction of live burst.

ACKNOWLEDGMENTS

This publication was made possible by NPRP grant #[8-519-1-108] from the Qatar National Research Fund (a member of Qatar Foundation), and by a Discovery Grant and a Steacie Memorial Fellowship from Natural Sciences and Engineering Research Council (NSERC) of Canada.

REFERENCES

[1] Ahmed Ali-Eldin, Maria Kihl, Johan Tordsson, and Erik Elmroth. Analysis and Characterization of a Video-on-demand Service Workload. In *ACM MMSys, 2015.*

[2] David M. Blei and John Lafferty. 2006. Correlated topic models. *Advances in neural information processing systems* 18 (2006), 147.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022.

[4] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., 288–296.

[5] Xianhui Che, Barry Ip, and Ling Lin. 2015. A Survey of Current YouTube Video Characteristics. *IEEE MultiMedia* 22, 2 (Apr 2015), 56–63.

[6] Stuart Dredge. 2015. Twitter’s Periscope video app has signed up 10M people in four months. http://www.theguardian.com/technology/2015/aug/13/twitter-periscope-video-app-10m-people/. (August 2015).

[7] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.

[8] Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

[9] Di Niu, Zimu Liu, Baochun Li, and Shuqiao Zhao. Demand forecast and performance prediction in peer-assisted on-demand streaming systems. In *IEEE INFOCOM, 2011.*

[10] Matthew Rowe. Forecasting Audience Increase on YouTube. In *Proc. of the International Workshop on User Profile Data on the Social Semantic Web, 2011.*

[11] Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. Exploiting Topic based Twitter Sentiment for Stock Prediction. In *ACL, 2013.*

[12] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How Useful Are Your Comments?: Analyzing and Predicting Youtube Comments and Comment Ratings. In *ACM WWW, 2010.*

[13] Alex Borges Vieira, Ana Paula Couto da Silva, Francisco Henrique, Glauber Goncalves, and Pedro de Carvalho Gomes. SopCast P2P Live Streaming: Live Session Traces and Analysis. In *ACM MMSys, 2013.*

[14] Cong Zhang and Jiangchuan Liu. 2016. Emoticon-aware Topic Model in Crowdsourced Interactive Live Streaming. *Simon Fraser University, Tech. Rep.* (2016). https://goo.gl/fVcxPu

[15] Cong Zhang, Jiangchuan Liu, and Haiyang Wang. Towards Hybrid Cloud-assisted Crowdsourced Live Streaming: Measurement and Analysis. In *ACM NOSSDAV, 2016.*

[16] Jun Zhu and Eric P. Xing. Sparse Topic Coding. In *UAI, 2011.*

[13] “X" means that the approach cannot predict the corresponding stage.