TSCREC: Time-sync Comment Recommendation in Danmu-Enabled Videos

Jiayi Chen*†,Wen Wu‡✉, Wenxin Hu‡,Liang He*†✉
*Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, China
†School of Computer Science and Technology, East China Normal University, Shanghai, China
‡School of Data Science and Engineering, East China Normal University, Shanghai, China
Email: jychen@ica.stc.sh.cn, {wwu, wxhu}@cc.ecnu.edu.cn, lhe@cs.ecnu.edu.cn

***Abstract*—In recent years, Time-sync Comment (TSC), as known as "Danmaku" or "Danmu", has been increasingly recognized as a valuable representation of video being incorporated into the process of generating video or highlight recommendation. However, little work has studied how to recommend proper TSC when users are watching the video. Providing some suggestions for users when they want to post TSCs can not only enhance the real-time interactions among users within videos, but also motivate users to post more TSCs which could be useful for video or highlight recommendation in return. In order to accomplish the TSC recommendation task, we extract candidates from existing comments sent by other users. Specifically, we propose a model, namely "TSCREC", which uses a bidirectional Gated Recurrent Unit to capture the semantic meaning of existing comments and assigns scores for comments by a Multi-Layer Perceptron. Furthermore, considering the importance of correlations among comments, we also take similarities among comments as features. In addition, we design a set of novel evaluation metrics which combines n-grams overlapping and ranking order to measure the quality of the recommendation list. We conduct experiments on real-world datasets, and the results show that our TSCREC model outperforms baseline approaches.**

1. Introduction

The past decade has witnessed the booming increase of online videos, such as online tv series, live streaming and user-generated short videos. Recently, an emerging type of interaction, Time-sync Comment (TSC), becomes popular on online video websites in China and Japan. Different from the traditional online video websites where users can only leave their comments in a separate zone, TSCs sent by users are flying over the screen when the video is playing. This kind of interaction can not only allow users to leave real-time comments related to the corresponding video content, but also bring users the feeling of companionship [1].

Therefore, researches regarding TSCs have attracted more and more attention in recent years. Existing studies mainly focus on treating TSCs as the content representation of the video and further incorporating TSCs into the process of generating video recommendations [2, 3] or highlight recommendations [4, 5]. However, how to recommend proper TSCs to help users select for posting during video has been rarely studied. Actually, providing users with some alternative TSCs enables them to express their real-time thoughts more conveniently and enhance the user-user interaction largely. In addition, it may facilitate the generation of TSCs data which would be useful for video or highlight recommendation in return.

For accomplishing the task, in this paper we select candidates from existing TSCs that other users have sent before rather than generate new comments by textual, visual or audio information, even if text generation such as document summarization has drawn great attention [6–8]. An important reason is that the quality and fluency of machine-generated comments cannot be guaranteed. As proven in [9], the quality of generated comments is lower than that of human. Though generated comments might be useful to enrich the content, recommending such comments for users to post may have a negative impact on user experience.

Concretely, we design a deep learning-based model which incorporates a widely used Recurrent Neural Network to learn the semantic meaning of comments and a Multi-layer Perceptron (MLP) to assign scores for each comment. More specifically, in order to model the dependency of surrounding comments, we directly compute the similarities among these comments and treat them as part of input features, because TSCs do not contain a sequential order that traditional sentences have in documents. In addition, compared with the conventional user feedbacks that contain implicit feedback like clicks and explicit feedback like ratings, TSCs cannot be directly transformed to numerical signals, which leads to the difficulty in generating training samples and evaluation metrics. To solve the issues, we propose a set of novel evaluation metrics (Max_{ROUGE} and $NDCG_{ROUGE}$) to measure the quality of recommendation list by combining both NDCG (Normalized Discounted Cumulative Gain) [10] and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [11], which focus on both relevance and ranking order. To generate training samples, we follow [6] to optimize scores of candidates according to ROUGE directly.

The main contributions of our work are as follows:

- We propose a new recommendation task: Time-sync Comment recommendation for providing candidate TSCs when users want to leave real-time comments

2375-0197/20/$31.00 ©2020 IEEE
DOI 10.1109/ICTAI50040.2020.00021

during videos, which can enhance the user-user interaction.

- We use a deep learning-based model to extract candidates from existing surrounding comments. The model takes both semantic representations and inter-comment correlations as features to assign scores for each comment.
- We design a set of new evaluation metrics based on widely used Natural Language Processing (NLP) metrics for Time-sync comment recommendation task. The experimental results on real-world datasets show that our model obtains a better performance relative to baselines for recommending TSCs.

In the following, we first introduce related work on TSCs in Section 2. We then describe our proposed TSCREC model in Section 3. We further present the details of our experiment design and analyze the results in Section 4. We finally conclude the paper and indicate some future directions in Section 5.

2. Related Work

In recent years, an increasing number of researches on TSCs have been proposed. Some real-world TSC datasets were created like [12–14] and further used in different applications such as motivation and behavior analysis [1, 15, 16], recommendation [2–5, 17] and sentiment analysis [13].

As for the studies that focus on using TSCs to analyze user experience and motivations, Chen et al. explored the gratification such as entertainment, feeling of being in company, sense of belonging and information seeking from TSCs [1]. Similarly, Ma et al. studied user motivations and behaviors by conducting a survey and analyzing the features of TSCs [16]. In addition, Chen et al. found that users' motivations and hindrance of watching videos with TSCs are different from each other [15].

Regarding the applications of TSCs on recommendation, previous work can be further refined according to the type of recommendation (i.e., video-level and frame-level recommendations). On one hand, video-level recommendation models used TSCs as the representation of video content and predicted videos the user may click in the following. Wu et al. learned representations of TSCs by their temporal positions in a video for obtaining the video content and user preference [2]. Yang et al. studied the herding effect based attention for achieving the personalized time-sync video recommendation [3]. On the other hand, frame-level recommendation used TSCs to provide highlights of videos for users. Chen et al. proposed a unified model which combined textual and visual information and learned user preference to recommend video frames [4]. Ping et al. recommended video frames by a clustering-based model according to user moods extracted from TSCs [5]. Lv et al. proposed a semantic embedding model and labeled the highlights of videos by TSCs [17].

However, to the best of our knowledge, there is little work has studied how to recommend proper TSCs for users when they are watching videos. It is a new recommendation task we would like to accomplish in this paper, which could be useful to enhance the user-user interaction and generate more TSCs.

3. Methodology

In this section, we introduce our proposed model, namely **TSCREC**, for Time-sync Comment recommendation task.

3.1. Solution Overview

Before introducing the model, we first give a solution overview of TSC recommendation task, as shown in Solution 1. TSC recommendation works when a user wants to leave a real-time comment in a video. We use v to represent the video and pl to denote the playtime of v that the user shows the intention of posting comments, such as click the input zone. The model first collects existing TSCs whose playtime is close to pl and regards them as **surrounding comments** $T = \{c_1, c_2, ...c_{|T|}\}$. Then the model calculates scores of all comments in T and ranks these comments according to scores in descending order. Finally, the top-K comments with the highest scores are recommended to the user. We denote the comment sent by user after watching the video slot as t (**target comment**).

Solution 1 Time-sync Comment Recommendation

```
Input: Video $v$, Current Playtime $pl$
Output: A Ranked List of Time-sync Comments
function TIME-SYNC RECOMMENDATION($v, pl$)
    Find existing time-sync comments of $v$ whose playtime is close to $pl$, and construct the surround comments $T = \{c_1, c_2, ...c_{|T|}\}$.
    Compute the score $p_j$ of comment $c_j \in T$.
    Rank all comments in $T$ according to scores
    return top-$K$ comments with highest scores
end function
Finally, the user leaves a comment denoted by $t$.
```

3.2. TSCREC Recommender

In this section, we introduce our TSCREC recommender model in detail. The architecture of TSCREC model is shown in Figure 1.

Sentence Encoder For each time-sync comment c_j from T, we first learn its semantic meaning. Specifically, we use a Bidirectional Gated Recurrent Unit (Bi-GRU) [18] as the sentence encoder, which takes the word sequence of c_j as the input and generates a latent embedding of the sentence. The GRU unit is defined as:

$$z_i = \sigma(W_z[x_i, h_{i-1}]) \tag{1}$$

$$r_i = \sigma(W_x[x_i, h_{i-1}]) \tag{2}$$

$$\hat{h_i} = tanh(W_h x_i, r_i \odot h_{i-1}) \tag{3}$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \hat{h_i} \tag{4}$$

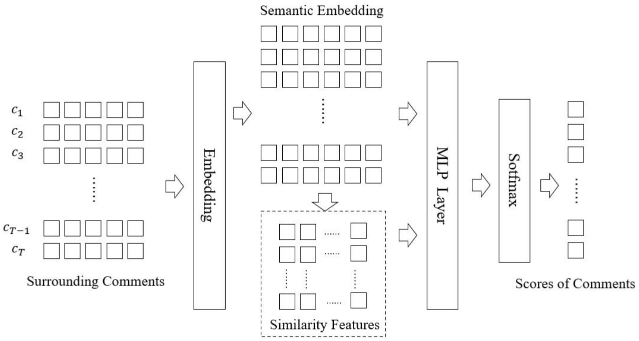


Figure 1. The Architecture of TSCREC Model

where x_i is the word embedding of the i-th word of c_j. The Bi-GRU contains a forward GRU network and a backward GRU network. The forward network takes the word sequence from left to right, while the backward network reads the input sentence reversely. We concatenate the last hidden states of Bi-GRU as the output of the comment:

$$hg_j = [\overleftarrow{h_{|c_j|}}, \overrightarrow{h_{|c_j|}}] \tag{5}$$

Score Function After obtaining the latent representations of each comment, we need to predict the score of these surrounding comments. Here we use a Multi-Layer Perceptron (MLP) to calculate the score of each comment:

$$\hat{p}_j = \sigma(MLP(hg_j)) \tag{6}$$

$$p_j = \frac{\exp\{\hat{p}_j\}}{\sum_{k \in T} \exp\{\hat{p}_k\}} \tag{7}$$

where p_j is the score of TSC c_j calculated by our model.

Inter-comment Correlations However, such a score function treats each sentence independently. Previous work has proven that modeling contextual information is useful for extracting important sentences [7]. Therefore, we take inter-comment correlations into consideration. Unlike sentences in a document which follow an order from beginning to end, TSCs do not own this attribute because they are crowd-sourced User Generated Content. Therefore, it is hard to model relationships among TSCs according to the traditional methods. Inspired by TextRank [19] which takes similarities between sentences to extract important sentences, we use similarity as an auxiliary feature of TSC. Cosine similarity is applied to compute similarities between two comments via the hidden states generated by Bi-GRU, and the auxiliary feature vector hs_j of comment c_j contains similarities with all surrounding comments:

$$sim(c_j, c_k) = \frac{hg_j \cdot hg_k}{|hg_j||hg_k|} \tag{8}$$

$$hs_j = [sim(c_1, c_j), ..., sim(c_{|T|}, c_j)] \tag{9}$$

Finally, we concatenate the hidden state hg_j and the auxiliary feature hs_j as the input feature of MLP, and calculate the score similarly.

$$p_j = softmax(\sigma(MLP([hg_j, hs_j]))) \tag{10}$$

3.3. Objective Function

Different from other recommendation or information retrieval tasks, there is no numerical feedback such as click data or ranking data in TSC recommendation scenario. Instead, the feedback of a user is what he actually sends at the time slot. Therefore, we need to design a mechanism to transform such a sentence-based feedback information to numerical information which can be used for model training. Inspired by [6], we generate feedback information by directly compute the ROUGE score between surrounding comments T and the target comment t, which is the base of our evaluation metric. Suppose $\hat{r}(c_i, t)$ denotes the ROUGE-1 score [11] (the definition of ROUGE criteria will be introduced in Section 4.2) where the target comment t performs as a reference, the normalized score is written as:

$$r(c_i, t) = \frac{\hat{r}(c_i, t) - min\{\hat{r}(c_j, t)|c_j \in T\}}{max\{\hat{r}(c_j, t)|c_j \in T\} - min\{\hat{r}(c_j, t)|c_j \in T\}} \tag{11}$$

The labels generated above are normalized by softmax function:

$$q_{c_i,t} = \frac{\exp\{r(c_i, t)\}}{\sum_{k \in T} \exp\{r(c_k, t)\}} \tag{12}$$

And we apply KL-Divergence as the objective function:

$$D_{KL}(P||Q) = \sum_{c_j \in T} p_j \log \frac{p_j}{q_{c_j,t}} \tag{13}$$

4. Experiment

4.1. Dataset and Experiment Setup

Item	Movie	Sport
number of videos	20	50
number of total comments	143,593	166,084
number of target comments	15,943	18,430
number of training samples	12,743	14,727
number of validation samples	1,600	1,844
number of testing samples	1,600	1,859

TABLE 1. THE STATISTICS OF THE DATASET

Dataset We use a real-world dataset provided by [14] whose TSC data is downloaded from bilibili, a popular Danmu-enabled video website in China. We first select 20 videos with most TSCs from category "Movie", and 50 videos from category "Sport", to investigate whether our model can perform well on different types of videos. For each video, we further split its TSCs into two parts: the surrounding part and the target part, according to the timestamp. The most recent sent TSCs construct the set of target comments, and will be used to train the model and evaluate the performance. The surrounding part contains the remaining comments that will be recommended for users. The ratio of target comments is set to 10%. Among target TSCs, we split these comments into training, validation and testing set, according to timestamps. The percentage of validation and testing set over the target comments are also

10%, and the remaining comments construct the training set. The details of the dataset are shown in Table 1.

Experiment Setup We first remove all punctuation in the dataset, and split words by jieba[1] which is a public NLP tool for Chinese. We train the word embeddings by the Skip-Gram algorithm [20], with the dimension of 300 and the window size of 3. We select 50 comments for each target comment as the surrounding comments which are most close to the playback time of the target comment as surrounding comments.

To optimize the model, we use Adam [21] as the optimization algorithm. For the hyperparemeter α, we set it to 0.0005 which is tuned on the validation set, and two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively. The size of each batch is 64. The size of word embedding and hidden state in Bi-GRU is set to 300.

4.2. Evaluation Metrics

In this section, we introduce our proposed evaluation metrics, MAX_{ROUGE} and $NDCG_{ROUGE}$. The basis of these two metrics is **ROUGE** [11], which is used for machine translation or abstractive summarization.

$\boldsymbol{Max_{ROUGE}@K}$ The quality of the list C depends on the relevant degree between the most relevant comment to the target comment t. Formally, in our case, $MAX_{ROUGE}@K$ is defined as the maximum ROUGE score between each comment c_j in C and target comment t:

$$MAX_{ROUGE}@K = max\{ROUGE(c_j, t)|c_j \in C\} \tag{14}$$

where $ROUGE(c_j, t)$ is the ROUGE score whose reference is the target comment t. The specific evaluation metrics $MAX_{ROUGE-1}@K$ and $MAX_{ROUGE-2}@K$ are abbreviated as $MR-1@K$ and $MR-2@K$ respectively.

$\boldsymbol{NDCG_{ROUGE}@K}$ The $MAX_{ROUGE}@K$ metric does not consider the ranking order of recommended comments. Under this circumstance, we follow the widely used ranking based metric, NDCG [10], to assess the ranking performance of recommendation models. We use ROUGE score as the relevance, and the Discounted Cumulative Gain $DCG_{ROUGE}@K$ is defined as:

$$DCG_{ROUGE}@K = \sum_{i=1}^{K} \frac{2^{ROUGE(c_i,t)} - 1}{\log_2(i+1)} \tag{15}$$

where $ROUGE(c_i, t)$ denotes the ROUGE score of candidate comment c_i at the position of i in the ranked recommendation list. Then the $NDCG_{ROUGE}@K$ is:

$$NDCG_{ROUGE}@K = \frac{DCG_{ROUGE}@K}{IDCG_{ROUGE}@K} \tag{16}$$

where the $IDCG_{ROUGE}@K$ is the normalization factor which is calculated by DCG of the surrounding comments T with an ideal order. Similar to $MR-1@K$ and $MR-2@K$, we abbreviate the $NDCG_{ROUGE-1}@K$ and $NDCG_{ROUGE-2}@K$ as $NR-1@K$ and $NR-2@K$.

4.3. Baselines

In this paper, we compare our proposed TSCREC model with the following methods:

- **Random** selects candidates from surrounding comments randomly.
- **Hot Words** tries to recommend comments that contain words frequently occurred in surrounding comments. The score of each candidate comment is computed by the sum of frequency of all words it contains, where the frequency is calculated over the surrounding comments. The main idea is inspired by TF-IDF [22].
- **TextRank** [19] is an unsupervised document summarization model which uses sentence similarity and PageRank algorithm to extract important sentences from documents. In this paper, we use the implementation textrank4zh[2] to deal with Chinese comments.
- **TSCREC w/o sim**, the variation of our TSCREC model which removes the similarity among comments and considers semantic features alone.

4.4. Results and Analysis

The comparison results are shown in Tables 2-3 (Movie dataset) and Tables 4-5 (Sport dataset).

In general, our TSCREC model achieves a better performance on both two datasets in terms of our proposed metrics, followed by TSCREC w/o sim, TextRank, Hot Words and Random. Compared to the TSCREC w/o sim model, TSCREC takes similarities among comments as features, which learns contextual information and better represents the importance of the comment under its context. Our TSCREC model also performs better than TextRank on the whole, though the performance of two models is close on Sport dataset in terms of ROUGE-2 when K=1 and K=5. This is because our model additionally considers the sequential information of a sentence by the GRU-based sentence encoder. For Random and Hot Words, they perform relatively worse because they do not consider either semantic meaning or contextual information.

However, some interesting observations are found in the results. To be specific, in some cases, the method Random achieves the best performance in terms of the evaluation metrics MR-1 and MR-2 when K=10. The reason is that these metrics mainly compute the maximum ROUGE score among candidates. When the size of candidates K is set to 10, which is relatively close to the size of surrounding comments ($|T| = 50$), the probability of recommending proper TSCs is increased by Random method. On the contrary, when K is set to a smaller value, the performance of Random w.r.t. MR-1 and MR-2 become poorer. Meanwhile, when using another evaluation metric $NDCG_{ROUGE}$, Random fails to achieve good performance (e.g., the score of Random w.r.t. NR-1@1 on Movie dataset is 0.150, which is

[1] https://github.com/fxsjy/jieba

[2] https://github.com/letiantian/TextRank4ZH

Methods	MR-1@1	MR-2@1	MR-1@5	MR-2@5	MR-1@10	MR-2@10
Random	0.058	0.020	0.175	0.068	**0.265**	**0.126**
Hot Words	0.080	0.016	0.157	0.043	0.206	0.068
TextRank	0.091	0.032	0.192	0.079	0.247	0.109
TSCREC w/o sim	0.105	0.032	0.204	0.074	0.259	0.112
TSCREC	**0.127**	**0.046**	**0.215**	**0.09**	0.262	0.118

TABLE 2. The performance of each model in terms of $Max_{ROUGE}@K$ on the dataset Movie (the best performances are in bold)

Methods	NR-1@1	NR-2@1	NR-1@5	NR-2@5	NR-1@10	NR-2@10
Random	0.150	0.036	0.178	0.040	0.213	0.049
Hot Words	0.224	0.033	0.272	0.043	0.312	0.051
TextRank	0.204	0.051	0.247	0.058	0.286	0.064
TSCREC w/o sim	0.244	0.048	0.286	0.056	0.324	0.064
TSCREC	**0.280**	**0.066**	**0.313**	**0.076**	**0.350**	**0.084**

TABLE 3. The performance of each model in terms of $NDCG_{ROUGE}@K$ on the dataset Movie (the best performances are in bold)

Methods	MR-1@1	MR-2@1	MR-1@5	MR-2@5	MR-1@10	MR-2@10
Random	0.078	0.026	0.218	0.092	0.283	**0.135**
Hot Words	0.105	0.029	0.210	0.078	0.263	0.114
TextRank	0.119	**0.047**	0.216	**0.094**	0.268	0.126
TSCREC w/o sim	0.117	0.033	0.225	0.087	0.281	0.126
TSCREC	**0.132**	0.042	**0.228**	0.093	**0.284**	0.132

TABLE 4. The performance of each model in terms of $Max_{ROUGE}@K$ on the dataset Sport (the best performances are in bold)

Methods	NR-1@1	NR-2@1	NR-1@5	NR-2@5	NR-1@10	NR-2@10
Random	0.171	0.069	0.209	0.087	0.245	0.117
Hot Words	0.250	0.083	0.296	0.113	0.339	0.147
TextRank	0.252	**0.119**	0.284	0.138	0.321	0.168
TSCREC w/o sim	0.278	0.097	0.322	0.129	0.366	0.167
TSCREC	**0.297**	0.116	**0.337**	**0.146**	**0.379**	**0.186**

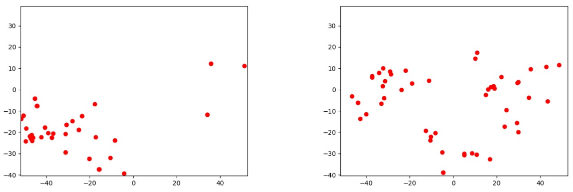
TABLE 5. The performance of each model in terms of $NDCG_{ROUGE}@K$ on the dataset Sport (the best performances are in bold)

approximately half of our TSCREC model), because NDCG measures the ranking quality.

In addition, we also analyze the distribution of surrounding comments (see Figure 2), because our TSCREC model considers the inter-comment correlations, which may influence the final performance. To visualize the distribution, we first train sentence embeddings by PV-DM model [23], where the dimension of sentence embedding is set to 300. We then apply the t-SNE algorithm [24] to transform latent embeddings to 2-dimensional space and obtain the distribution finally.

As illustrated in Figure 2, there are two typical types of surrounding TSCs. For the first type (Figure 2(a)), most comments follow a template and become extremely similar, and then there is a center in these comments. For example, it has been proven that TSCs contain large quantities of meaningless terms like "2333" and "hhhh" (onomatopoeia of laughing) [25]. For this type of surrounding comments, our TSCREC model achieves better performance than the overall performance (i.e., the MR-1@10 and MR-2@10 on test samples which contain "2333" are 0.434 and 0.395 respectively in Movie dataset)

For the second type of surrounding comments (Figure 2(b)), sentences do not follow a shared template. Instead, they reflect several different topics, and minor sentence centers are included rather than a major term such as "2333". Under this circumstance, our model has some ability to predict the comments accurately in this case, although the performance is a little weaker than the first type. It is reasonable because the diverse surrounding comments brings more difficulty in prediction.



(a) Surrounding Comments of a Single Center (b) Surrounding Comments of Multiple Centers

Figure 2. Two Typical Types of Surrouding Comments

5. Conclusion and Future Work

In this paper, we propose a new task to recommend Time-sync Comments for users to select when watching the video. To accomplish the task, we extract candidates from existing surrounding comments. We concretely propose a deep learning-based approach which takes both sentence semantic representation and inter-comment correlations into consideration. Meanwhile, we put up with new evaluation metrics which are based on widely used NLP metrics to assess the performance of TSC recommendation. Experiments on real-world datasets show that our model can provide better recommendations compared to baseline models. In the future, it would be interesting to study how to accomplish the TSC recommendation task better. In addition, considering user's previous behaviors to improve the quality of recommended TSC list would be another future direction. Novel evaluation metrics could be designed as well.

Acknowledgments

We thank National Natural Science Foundation of China (under project No.61907016), Science and Technology Commission of Shanghai Municipality (under project No. 19511120200), and the Fundamental Research Funds for the Central Universities for sponsoring the research work.

References

[1] Y. Chen, Q. Gao, and P. P. Rau, "Understanding gratifications of watching danmaku videos - videos with overlaid comments," in *CCD 2015*, vol. 9180, 2015, pp. 153–163.

[2] Z. Wu, Y. Zhou, D. Wu, Y. Zhou, and J. Qin, "Crowdsourced time-sync video recommendation via semantic-aware neural collaborative filtering," in *ICWE 2019*, 2019, pp. 171–186.

[3] W. Yang, W. Gao, X. Zhou, W. Jia, S. Zhang, and Y. Luo, "Herding effect based attention for personalized time-sync video recommendation," in *ICME 2019*, 2019, pp. 454–459.

[4] X. Chen, Y. Zhang, Q. Ai, H. Xu, J. Yan, and Z. Qin, "Personalized key frame recommendation," in *SIGIR 2017*, 2017, pp. 315–324.

[5] Q. Ping, "Video recommendation using crowdsourced time-sync comments," in *RecSys 2018*, 2018, pp. 568–572.

[6] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *ACL 2018*, 2018, pp. 654–663.

[7] P. Ren, Z. Chen, Z. Ren, F. Wei, J. Ma, and M. de Rijke, "Leveraging contextual sentence relations for extractive summarization using a neural attention model," in *SIGIR 2017*, 2017, pp. 95–104.

[8] H. Xu, Y. Cao, R. Jia, Y. Liu, and J. Tan, "Sequence generative adversarial network for long text summarization," in *ICTAI 2018*, 2018, pp. 242–248.

[9] S. Ma, L. Cui, D. Dai, F. Wei, and X. Sun, "Livebot: Generating live video comments based on visual and textual contexts," in *AAAI 2019*, 2019, pp. 6810–6817.

[10] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[11] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *WAS 2004*, 2004, pp. 74–81.

[12] Z. Liao, Y. Xian, X. Yang, Q. Zhao, C. Zhang, and J. Li, "Tscset: A crowdsourced time-sync comment dataset for exploration of user experience improvement," in *IUI 2018*, 2018, pp. 641–652.

[13] Q. Bai, Q. V. Hu, L. Ge, and L. He, "Stories that big danmaku data can tell as a new media," *IEEE Access*, vol. 7, pp. 53 509–53 519, 2019.

[14] G. Lv, K. Zhang, L. Wu, E. Chen, T. Xu, Q. Liu, and W. He, "Understanding the users and videos by mining a novel danmu dataset," *IEEE Transactions on Big Data*, 2019.

[15] Y. Chen, Q. Gao, and P.-L. P. Rau, "Watching a movie alone yet together: Understanding reasons for watching danmaku videos," *International Journal of Human–Computer Interaction*, vol. 33, no. 9, pp. 731–743, 2017.

[16] X. Ma and N. Cao, "Video-based evanescent, anonymous, asynchronous social interaction: Motivation and adaption to medium," in *CSCW 2017*, 2017, pp. 770–782.

[17] G. Lv, X. Tong, E. Chen, Z. Yi, and Z. Yi, "Reading the videos: temporal labeling for crowdsourced time-sync videos based on semantic embedding," in *AAAI 2016*, 2016, pp. 3000–3006.

[18] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014*, 2014, pp. 1724–1734.

[19] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *EMNLP 2004*, 2004, pp. 404–411.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR 2013 Workshop Track*, 2013.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*, 2015.

[22] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[23] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML 2014*, 2014, pp. 1188–1196.

[24] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[25] Q. Wu, Y. Sang, S. Zhang, and Y. Huang, "Danmaku vs. forum comments: Understanding user participation and knowledge sharing in online videos," in *GROUP 2018*, 2018, pp. 209–218.