Interactive Variance Attention based Online Spoiler Detection for Time-Sync Comments

Wenmian Yang
Shanghai Jiao Tong University
State Key Lab of IoT for Smart City,
CIS, University of Macau
sdq11111@sjtu.edu.cn

Weijia Jia*
State Key Lab of IoT for Smart City,
CIS, University of Macau
Shanghai Jiao Tong University
jiawj@um.edu.mo

Wenyuan Gao
Shanghai Jiao Tong University
gaowenyuan1028@sjtu.edu.cn

Xiaojie Zhou
Shanghai Jiao Tong University
szxjzhou@sjtu.edu.cn

Yutao Luo
Shanghai Jiao Tong University
luoyt1996@sjtu.edu.cn

ABSTRACT

Nowadays, time-sync comment (TSC), a new form of interactive comments, has become increasingly popular on Chinese video websites. By posting TSCs, people can easily express their feelings and exchange their opinions with others when watching online videos. However, some spoilers appear among the TSCs. These spoilers reveal crucial plots in videos that ruin people's surprise when they first watch the video. In this paper, we proposed a novel Similarity-Based Network with Interactive Variance Attention (SBN-IVA) to classify comments as spoilers or not. In this framework, we firstly extract textual features of TSCs through the word-level attentive encoder. We design Similarity-Based Network (SBN) to acquire neighbor and keyframe similarity according to semantic similarity and timestamps of TSCs. Then, we implement Interactive Variance Attention (IVA) to eliminate the impact of noise comments. Finally, we obtain the likelihood of spoiler based on the difference between the neighbor and keyframe similarity. Experiments show SBN-IVA is on average 11.2% higher than the state-of-the-art method on F1-score in baselines.

CCS CONCEPTS

• **Computing methodologies** → *Lexical semantics*.

KEYWORDS

Opinion Mining, Time-Sync Comments, Attention Mechanism, Spoiler Detection

*Corresponding author

1 INTRODUCTION

Recently, people enjoy watching videos online in their leisure time. During viewing time, many people are willing to share their feelings and exchange ideas with others. Time-sync comment (TSC) is such a new form of interactive comment designed to meet this demand. People can post TSCs at any time during their viewing, and these comments posted will be seen by other viewers synchronously on the video screen. However, many TSCs contain the spoiled content that reveal crucial plots in videos such as the content of movie ending, the murderer identity of criminal investigation TV-series, or final scores of sports games which can ruin the surprise of those who have not watched this video. To avoid being spoiled, lots of people have no choice but to hide all the TSCs regardless of the TSCs are spoilers or not. Some people even stay away from certain online video websites with TSCs to minimize the chances of being spoiled, which makes the users lose the opportunity to discuss with others. To ensure that users can communicate through TSCs and avoid being spoiled by TSCs, it is critical to detect spoilers from TSCs.

Most of the existing spoiler detection methods are based on key-matching and traditional machine learning methods. For instance, Nakamura *et al.* [22] and Golbeck *et al.* [8] filter out spoilers according to predefined keywords. Guo *et al.* [9] use LDA-based machine learning method to rank the most likely spoilers. Chang *et al.* [3] propose a deep neural spoiler detection model using a genre-aware attention mechanism. However, these methods are based on common reviews or articles on social media. They are not well designed for TSC data since they do not consider the unique properties of TSCs. Distinguished from common video comments, TSCs have short-text, interactive, real-time and high-noise properties.

The most prominent property of TSCs is that they are **short texts** with lots of network slang. TSCs are usually not complete sentences, and generally, have no more than 10 words [29]. The words in TSCs do not contribute equally to determining the spoilers. An example of criminal investigation TV-series is shown in Fig. 1. The words "murderer", "delivery guy" and "Captain Guan" are plot-related words or important character names, which have a higher possibility to trigger spoiled contents. Therefore, we need to pay different attention to words with different importance in TSCs.

arXiv:1908.03451v2 [cs.IR] 21 Aug 2019

Online Time-sync Videos

关队出场了！！！　　这不是关队，是俺弟弟
凶手是送外卖的！凶手是送外卖的！！　头皮发麻
关队威武霸气　　2333333

Time-sync Comments　　| video's playback time

A	Captain Guan appeared !!!	17:11
B	The murderer is a delivery guy	17:14
C	Captain Guan is domineering	17:16
D	Blood freezes	17:20
E	This is Captain Guan's brother	17:21
F	2333333	17:25

......

Keyframes of the Video

一眨眼菜刀换成枪了　　外卖小哥竟然是凶手
副朝刚次医泰明有点像　　美团来打一波
你是通缉犯，送外卖的知道你的身份　　萌面，送外卖的是和人合租的

Keyframes Comments

G The delivery guy turned out to be the murderer

H The plot is similar to the forensic Qin Ming

I The younger brother is a wanted criminal, and the delivery guy know his identity
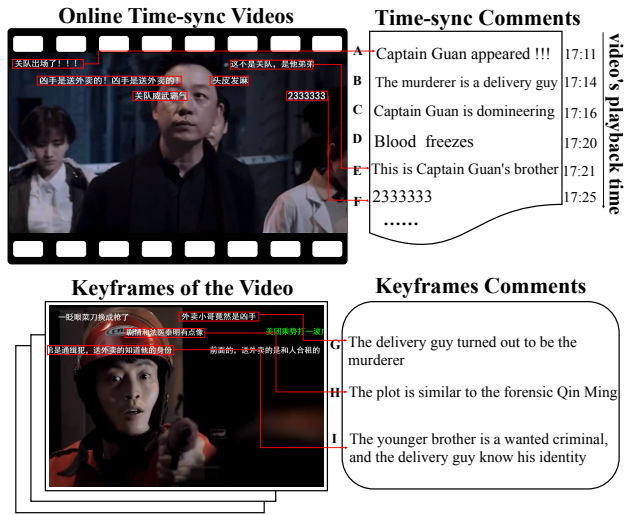
Figure 1: Example of the short-text, interactive, real-time and high-noise properties of TSCs.

Another important property of TSCs is **interactive**. The topic of the latter posted TSCs usually depends on the former, which is similar to the herding mentality effect [10, 32, 35] in social science. In Fig. 1, the TSC A, C, and E share the same topic of discussing Captain Guan when Captain Guan's face appears on the screen. TSC A first shows his interest in the appearance of Captain Guan. Then this topic arouses other viewers' interests, and other TSCs within the same topic (TSC C and E) appear due to the impact of TSC A. In the example, the generation of a TSC is not independent but influenced by the previous TSCs. This interactive property is helpful to distinguish spoiler TSCs: If the topic of a TSC is closed to its surrounding TSCs, it is less likely to be a spoiler because it describes the current content in the video.

The **real-time** property is also helpful to study the relationship between TSCs. As shown in Fig. 1, each TSC has a timestamp synchronous to the playback time of the video, and its content is often related to what is happening currently in the video. Therefore, a TSC is more likely to become a spoiler when it has a high semantic similarity to the TSCs of the video highlights (such as the end of the movie, the moment of revealing the murderer in the drama, and the final stage of the sports game). In this paper, we call those video highlights as "keyframes". In Fig. 1, the content of TSC B is different from its surrounding TSCs but very close to the keyframe TSC G, which means TSC B describes an important plot on purpose and has high possibility to be a spoiler. And actually, TSC B is a spoiler indicating the identity of the murderer.

In addition to the above three properties, TSCs are **high-noise**. Many TSCs unrelated to the video content appear due to the viewers' diverse topics and particular ideas. For example, TSC D and F in Fig. 1 express viewers' feelings irrelevant to the video content. They have weak semantic relevance with their neighboring TSCs, and they are not spoilers, either. The existence of noise causes great troubles to establish semantic relationships between TSCs. Therefore, how to make use of the interactive and real-time properties of

TSCs and eliminate the impact of noise is the central challenge to predict the spoilers accurately and effectively.

Based on the above motivation and challenges, we propose a novel Similarity-Base Network with Interactive Variance Attention (SBN-IVA) to classify TSCs as spoilers or not. Specifically, due to the short-text property, we first extract textual features of TSCs through the word-level attentive encoder, which assigns the weights based on the importance of words. Then, based on the interactive and real-time properties of TSCs, we design a Similarity-Base Network (SBN) to acquire the similarities between each TSC and its neighbors and keyframes respectively according to the semantic and timestamps. The higher the neighbor similarity and the lower the keyframe similarity, the less likely the TSC is a spoiler. Considering the high-noise property, we implement Interactive Variance Attention (IVA) mechanism in the framework to effectively reduce the impact of noise due to the low semantic similarities between the noise and its surrounding TSCs [34]. Finally, we obtain the likelihood of the spoilers based on the difference between the neighbor and keyframe similarity.

The main contributions of this paper are as follows:

1. We propose a novel Similarity-Base Network with Interactive Variance Attention (SBN-IVA) to detect spoilers from TSCs.
2. We combine attention mechanism both in word-level and sentence invariance level in our model, which takes the short-text, interactive and real-time properties of TSCs into account and effectively reduces the impact of noise.
3. We evaluate our proposed model with real-world datasets in the categories of movies, TV-series, and sports on mainstream video websites. Experiments show our model outperforms the baselines in Precision, Recall, and F1-Score.

2 RELATED WORK

In this section, we introduce the related work in three aspects.

Time-sync comment (TSC) provides a new data source for text mining regarding the online videos and attracts research on it increasingly. Wu *et al.* [29] first formally introduce the TSC, and propose a preliminary method to extract video tags. Yang *et al.* [34] systematically collate the features of TSCs and introduces a graph-based algorithm to eliminate the impact of noise prominently. Then further usages of TSCs data are proposed like extracting highlight shots [24, 30], labeling important segments [19], detecting events [16], generating temporal descriptions of videos [31], and video recommendation [5, 23, 32]. Besides, some other researchers turn to build and improve TSC dataset. Liao *et al.* [17] present a rich self-labeled TSC dataset with four-level structures for user experience improvement. Chen *et al.* [4] and Chen *et al.* [6] also do similar evaluating work on real-world multimedia dataset. The above methods show the promising potential of TSC data and greatly inspire our work.

Spoiler detection (or plot detection) has achieved great attention on studies for review documents or articles in social media. The recent studies of spoiler detection mainly focus on two domains, keyword matching, and machine learning methods. Keyword matching methods filter out spoilers based on predefined keywords, such as actor and character names [8], the name of the sports team or sporting event[22], or words in the latter half of the document [20].

However, the keyword matching methods require human-fixed input and are not widely used in various application scenarios. Besides, keyword matching methods usually have high recall and low precision performance since they treat many positive comments as spoilers. The other domain of research is machine learning methods. Guo *et al.* [9] use bag-of-words representation and LDA-based model to rank spoilers. They calculate the spoiler probability scores by the similarity between comments and item descriptions from IMDB. Iwai *et al.* [13] evaluate five conventional machine-learning methods and improve the method through generalizing keywords. Jeon *et al.* [14] detect spoilers by SVM classification according to four features: "named entity", "frequently used verb", "objectivity + URL", and "tense". Hijikata *et al.* [11] introduce location information and neighborhood plot probability to help identify spoilers with contextual information by SVM. Moreover, Chang *et al.* [3] propose a deep neural spoiler detection model using a genre-aware attention mechanism. However, the above methods are not well-designed for detecting spoilers in TSC data, as they ignore the interactive, real-time, and high-noise properties of TSC data.

Attention Mechanism is firstly proposed and used in machine translation [1, 2] in natural language processing. They integrate attention mechanism with an encoder-decoder framework to align the phrase in source language with the target language before translation. Vinyals *et al.* [28] introduce a novel agnostic attention-enhanced seq-to-seq model to solve syntactic constituency parsing problem. Sukhbaatar *et al.* [26] and Kumar *et al.*[15] use the attention mechanism to improve traditional methods in question answering problem. Recently, Vaswani *et al.* propose a new kind of seq-to-seq network architecture, Transformer [7, 25, 27, 37], which is more suitable to solve the long-range dependencies problems. However, since the TSCs are usually short-text, the Transformer is not suitable for TSCs. Our work adopts attention mechanisms both in word and sentence level. The word-level attentive encoder assigns the high weight to informative words as Sukhbaatar *et al.* [26]. The sentence-level Interactive Variance Attention (IVA) refers to soft attention [18, 33, 36], where higher attentive weights indicate that the corresponding features are more informative for the end task.

3 PRELIMINARIES AND PROBLEM DEFINITION

In this section, we first provide the problem definition in Section 3.1. And then, the problem formulation is introduced in Section 3.2.

3.1 Problem Definition

In this paper, we aim at detecting and labeling spoilers, i.e., TSCs containing certain plots in videos. We call the TSC we are studying as the target TSC. The key idea of our framework is to compare the neighbor similarity and the keyframe similarity of the target TSC. Intuitively, we build a similarity-based deep framework to solve this task in a supervised way. Specifically, we first represent the TSCs as semantic vectors of through word-level attentive encoder. Then, we compute the neighbor similarity and the keyframe similarity of the target TSC though SBN with sentence-level IVA. The definitions of

Table 1: Notations and descriptions.

Notations	Descriptions
TSC	TSC sequence
TSC_i	i-th TSC in TSC
L	Number of TSCs.
T	Timestamps sequence of TSCs
t_i	Timestamp of TSC_i
WL_i	Word sequence of TSC_i
K	Number of words
$wl_{i,j}$	j-th word of TSC_i
R	Number of former TSCs
F	Sequence of former neighbors
F_i	i-th former neighbor
P	Number of keyframes
KEY	Sequence of keyframes
KEY_i	i-th keyframe
$x_{i,j}$	Embedding vector of $w_{i,j}$
$LSTM$	Long Short Term Memory Network
$h_{i,j}$	LSTM hidden state of $x_{i,j}$
$\alpha_{i,k}$	Word-level attention score of $h_{i,j}$
W_s	Weight matrix
u_s	Context vector
$Tseq_i$	Target sentence vector of TSC_i
$NSEQ$	Sentence vector sequence of former neighbors
$Nseq_i$	Sentence vector of former neighbor F_i
$KSEQ$	Sentence vector sequence of keyframes
$Kseq_i$	Sentence vector of keyframe KEY_i
$Nsim_r$	Semantic similarity between $Tseq_i$ and $Nseq_r$
G_{Nsim_i}	Weighted average similarity of TSC_i's neighbors
β	Hyper-parameter of decay function
$Ksim_p$	Semantic similarity between $Tseq_i$ and Keq_q
G_{Ksim_i}	Maximum keyframe similarity of TSC_i
$\hat{y}_i$	Prediction result of TSC_i
y_i	Ground truth of TSC_i
S_r	Semantic similarity of F_r to other neighbors
$S_{i,j}$	Semantic similarity between $Nseq_i$ and $Nseq_j$
$\bar{S}_r$	Normalized similarity of F_r to other neighbors
$\bar{S}_{i,j}$	Normalized similarity between $Nseq_i$ and $Nseq_j$
D_r	Similarity variance of $\bar{S}_r$
$\bar{D}_r$	Sentence-level attention score of F_r

the neighbor similarity and the keyframe similarity in our two-level framework are shown as follows:

Neighbor similarity: Former neighbors are several consecutive TSCs in front of the target TSC. We define the semantic similarity between target TSC and each of its former neighbors as neighbor similarity. We employ the weighted average of the neighbor similarities as overall neighbor similarity for each target TSC. The high overall neighbor similarity means that the target TSC describes current video content, and thus it is less likely to be a spoiler.

Keyframe similarity: Keyframes are the highlight clips of the videos, such as the end of the movie, the moment of revealing

the murderer in the drama, and the final stage of the sports game. The TSCs in the keyframes contain a large number of keywords that are related to the videos' important plots. If these keywords appear in the TSCs before these keyframes, these TSCs are spoilers. For each video, we define the similarity between target TSC and each keyframe as keyframe similarity. If the content of the target comment is similar to any of the keyframe, it is very likely to be a spoiler. Therefore, we take the maximum keyframe similarity as the overall keyframe similarity. We perform a qualitative analysis of the cumulative occurrence proportion of key-TSCs (TSCs with keywords) with the video's playback time ratio. As shown in Fig. 2, the key-TSCs usually post in the last quarter of the video. Therefore, the keyframes generally appear in the last quarter of the video. Moreover, the keyframes are usually the most intensive time periods of TSCs [30]. As a result, we cut the last quarter of the video into frames, and the duration of each frame is 10 seconds. We count the number of TSCs contained in each frame and select P frames with the most TSCs as keyframes.
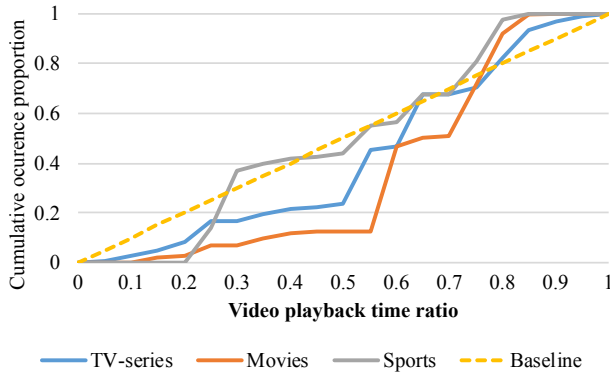

Figure 2: Cumulative occurrence proportion of key-TSCs with the video's playback time ratio.

3.2 Problem Formulation

For each video, we have the TSC sequence $TSC = \{TSC_1, TSC_2, ..., TSC_L\}$, where L is the number of TSCs in the video. We define $T = \{t_1, t_2, ..., t_L\}$ as the corresponding timestamps of TSCs, where $t_1 \leq t_2 \leq ... \leq t_L$. The word list of $TSC_i, i \in [1, L]$ is defined as $WL_i = \{wl_{i,1}, wl_{i,2}, ..., wl_{i,K}\}$, where K is the number of words in TSC_i.

For each target TSC_i, we define its R former TSCs as former neighbors $\{TSC_{i-R}, TSC_{i-R+1}, ..., TSC_{i-1}\}$. The target TSC_i and its R former neighbors are consecutive in the TSC sequence of each video, so their timestamps $t_{i-R} < t_{i-R+1} < ... < t_i$. To simplify the annotation, we use $F = \{F_1, F_2, ..., F_R\}$ to indicate the set of former neighbors, and $\{t_{F_1}, t_{F_2}, ..., t_{F_R}\}$ to indicate their corresponding timestamps, where $t_{F_1} \leq t_{F_2} \leq ... \leq t_{F_R}$.

We also give the definition of keyframes of each video as $KEY = \{KEY_1, KEY_2, ..., KEY_P\}$, where P is the number of keyframes in each video. As mentioned in Section 3.2, we treat the P most intensive periods of TSCs in the last quarter of each video as the keyframes.

Given all the $TSC = \{TSC_i | 1 \leq i \leq L\}$ and their corresponding timestamps $T = \{t_1, t_2, ..., t_L\}$, our work is to predict whether TSC_i is a spoiler or not. To make more clear presentation, we list the notations used throughout the paper in Table 1.

4 MODEL

In this section, we first use the word-level attentive encoder to extract textual features of TSCs in Section 4.1. Then, we propose a novel Similarity-Based Network to detect the spoilers in Section 4.2. To further make use of the interactive and real-time properties and eliminate the impact of noise among TSCs, we implement Sentence-Level Interactive Variance Attention(IVA) to improve the detection accuracy of SBN in Section 4.3.

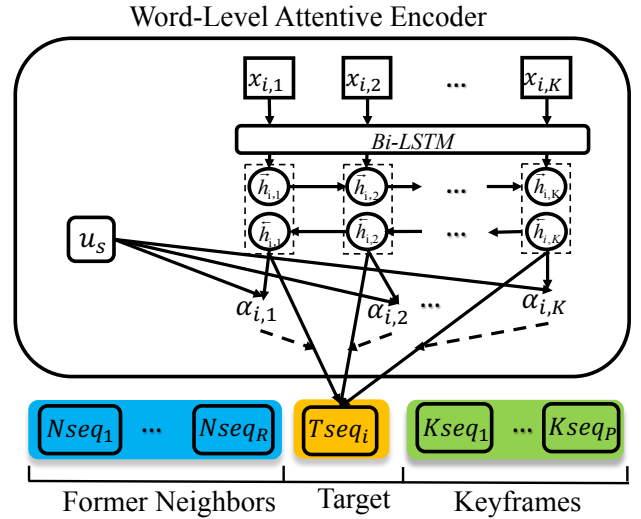## 4.1 Word-Level Attentive Encoder

Figure 3: Word-level attentive encoder of target TSC and its former neighbors.

In this section, we use the Bidirectional Long Short-Term Memory (Bi-LSTM) based word-level attentive encoder to extract textual features of each TSC. The architecture of the word-level attentive encoder is shown in Fig. 3.

Given the word sequence $WL_i = \{wl_{i,1}, wl_{i,2}, ..., wl_{i,K}\}$ of the TSC_i, we first embed the word $wl_{i,k}$ into fix-sized vector $x_{i,k}$. We pre-train the embedding matrix by Skip-Gram model [21] because it is better for infrequent words that often appear in TSCs. Afterwards, we use Bi-LSTM network [12] to extract textual features of each TSC. Specifically, for each $WL_i = \{wl_{i,1}, wl_{i,2}, ..., wl_{i,K}\}$, we have

$$\overrightarrow{h}_{i,1} = \overrightarrow{LSTM}(x_{i,1}) \tag{1}$$

$$\overleftarrow{h}_{i,K} = \overleftarrow{LSTM}(x_{i,K}) \tag{2}$$

$$\overrightarrow{h}_{i,k} = LSTM(x_{i,k}, \overrightarrow{h}_{i,k-1}), \overrightarrow{h}_{i,k-1}, k \in [2, K] \tag{3}$$

$$\overleftarrow{h}_{i,k} = LSTM(x_{i,k}, \overleftarrow{h}_{i,k+1}), \overleftarrow{h}_{i,k+1}, k \in [1, K-1] \tag{4}$$
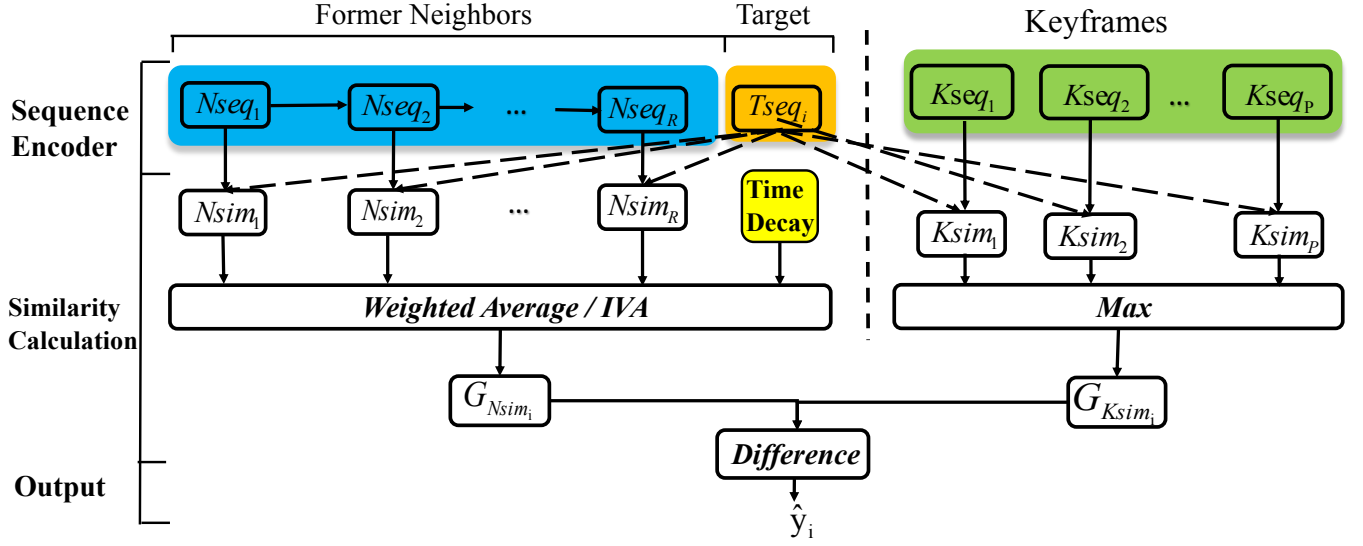
Figure 4: Similarity-Based Network.

and

$$h_{i,k} = (\overrightarrow{h}_{i,k} || \overleftarrow{h}_{i,k}) \tag{5}$$

where $||$ denotes vector concatenation.

As introduced in Section 1, each word in the word list $WL_i = \{wl_{i,1}, wl_{i,2}, ..., wl_{i,K}\}$ of the TSC_i does not contribute equally to determining whether TSC_i is a spoiler or not. For instance, words such character's name and verb related to survival or failure are highly likely to trigger spoilers, while common conjunctions, definite articles or emotional adjectives do not have a great influence on spoiler judgments. Therefore, we equip word-level encoder with attention mechanism to select words that can accurately represent the sentence meaning.

Given the Bi-LSTM state sequence $[h_{i,1}, h_{i,2}, ..., h_{i,K}]$, our attentive sequence encoder calculates a sequence of attention values $[\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,K}]$ by:

$$\alpha_{i,k} = \frac{exp(tanh(W_s \cdot h_{i,k})^T \cdot u_s)}{\sum_t exp(tanh(W_s \cdot h_{i,t})^T \cdot u_s)} \tag{6}$$

where W_s is a weight matrix, and u_s is the context vector to distinguish informative words from non-informative ones.

Finally, we obtain the target sentence vector $Tseq_i$ of TSC_i by:

$$Tseq_i = \sum_{k=1}^{K} \alpha_{i,k} \cdot h_{i,k} \tag{7}$$

According to the interactive property of TSCs, we consider the R former neighbors $F = \{F_1, F_2, ..., F_R\}$, and P keyframes $KEY = \{KEY_1, KEY_2, ..., KEY_P\}$ of of TSC_i. According to word-level attentive encoder, we obtain the sentence vector of former neighbors $NSEQ = \{Nseq_1, Nseq_2, ... , Nseq_R\}$ and keyframes $KSEQ = \{Kseq_1, Kseq_2, ..., Kseq_P\}$ based on certain TSCs which represent the semantic of keyframes, and take them as the input to the following SBN.

4.2 Similarity-Based Network

In this section, we introduce the SBN to obtain the likelihood of spoiler based on the semantic similarity between the target and its neighbors and the target and the keyframes.

Fig. 4 shows our integrated SBN. As mentioned in Section 4.2, after the process of word-level attentive encoder, we obtain R consecutive sequence vectors of former neighbors $NSEQ=\{Nseq_1, Nseq_2, ... , Nseq_R\}$ with their corresponding timestamps $t_{F_1} \leq t_{F_2} \leq ... \leq t_{F_R}$. For keyframes, we also use the word-level attentive encoder to get the semantic vector of each TSC, average the semantic vectors of all the TSCs in one keyframe and get the semantic sequence of P keyframes $KSEQ = \{Kseq_1, ..., Kseq_P\}$.

We first compute neighbor similarity $Nsim_r$ between the target TSC_i and each of its former neighbors F_r by

$$Nsim_r = sim(Nseq_r, Tseq_i) \tag{8}$$

where $sim(x, y) = \frac{x \cdot y}{|x| \cdot |y|}$.

Then, we take the weighted average similarity between the target TSC_i and its former neighbor by

$$G_{Nsim_i} = \sum_{r=1}^{R} (Nsim_r \cdot decay(t_{F_r}, t_i)) \tag{9}$$

where

$$decay(t_{F_r}, t_i) = \frac{exp(-\beta(t_{F_r} - t_i))}{\sum_{k=1}^{R} exp(-\beta(t_{F_k} - t_i))} \tag{10}$$

is the time decay function expressing that the interactive property between two TSCs decreases with the increase of time interval. If a former neighbor has a long time interval with the target TSC, it has a small effect on the overall neighbor similarity. Time decay rate β is a hyperparameter that will be discussed in Section 5.

Meanwhile, we compute keyframe similarity $Ksim_p$ between target TSC_i and each keyframe KEY_p. To be noticed, we do not introduce time decay here because we do not care the time interval

between keyframe and the target comment. Also, as long as the target TSC is similar to any keyframe, it is likely to be a spoiler. Therefore, we take the maximum keyframe similarity of the target TSC_i as the overall keyframe similarity G_{Ksim_i}.

$$Ksim_p = sim(\boldsymbol{Kseq_p}, \boldsymbol{Tseq_i}) \quad (11)$$

$$G_{Ksim_i} = Max.\{Ksim_1, Ksim_2, ..., Ksim_P\} \quad (12)$$

where $Max.$ denotes the maximum operation.

In our work, we take the prediction of the likelihood of spoiler as a binary classification problem, where 1 means the target TSC is a spoiler, and 0 otherwise. Intuitively, if the semantic of target TSC is closer to each of its former neighbors, it is less likely to be a spoiler because its topic describes current video content. Moreover, if the semantics of the target TSC is closer to any keyframe, it has more possibilities to be a spoiler because it is more likely to talk about important plots rather than following the playback time of the video. Therefore, we compute the difference between the overall neighbor similarity and overall keyframe similarity as a result. We determine the prediction result as

$$\hat{y}_i = sigmoid(G_{Ksim_i} - G_{Nsim_i}) \quad (13)$$

where $sigmoid(x) = \frac{1}{1+e^{-x}}$.

Finally, we use binary cross-entropy as our loss function:

$$L = y_i \cdot ln\hat{y}_i + (1 - y_i) \cdot ln(1 - \hat{y}_i) \quad (14)$$

where y_i is the ground truth that would be 1 if the TSC is a spoiler, and 0 otherwise.

4.3 Sentence-Level Interactive Variance Attention

In section 4.2, we have calculated the overall neighbor similarity based on the weighted average similarity between the target TSC and its former neighbors. However, the TSC data is high-noise. If the noise is contained in former neighbors, it makes trouble to generate accurate overall neighbor similarity. Therefore, we implement the sentence-level IVA mechanism to effectively eliminate the impact of noise. IVA mechanism aims to detect noise though similarity variance and assign a lower weight to the noise.

The IVA mechanism is formalized into a deep learning framework shown in Fig. 5. For each target TSC_i, we feed its neighbor sequence $\{Nseq_1, Nseq_2, ..., Nseq_R\}$ as the input of the framework.

Since noise has weak relevances with their surroundings [34], we compute the similarity of any two TSCs in the sequence. For each $Nseq_r$, we compute its similarity with each TSC in the whole input sequence as the similarity vector $\boldsymbol{S_r} = \{S_{r,1}, S_{r,2}, ..., S_{r,R}\}$ by

$$S_{i,j} = sim(\boldsymbol{Nseq_i}, \boldsymbol{Nseq_j}) \quad (15)$$

and normalize it by:

$$\overline{S}_{i,j} = softmax(S_{i,j}) \quad (16)$$

where $softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)}$.

The noise has weak relevance to its surroundings and has a high similarity to itself, so the similarity distribution of noise is very concentrated and have relatively high similarity variance. On the contrary, non-noise TSCs have relatively placid similarity distribution and low similarity variance. Based on this point, we can
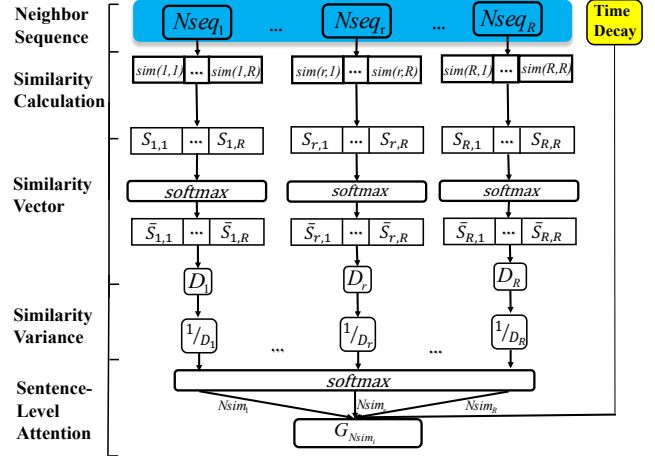

Figure 5: Sentence-Level Interactive Variance Attention (IVA) Framework.

distinguish noise comments from non-noise comments through their difference in similarity variance.

For $Nseq_r$, we compute the similarity variance D_r of its normalized similarity vector $\overline{\boldsymbol{S}}_{\boldsymbol{r}} = \{\overline{S}_{r,1}, \overline{S}_{r,2}, ..., \overline{S}_{r,R}\}$ as:

$$D_r = \frac{1}{R} \sum_{i=1}^{R} (\overline{S}_{r,i} - Ave.(\overline{\boldsymbol{S}}_{\boldsymbol{r}}))^2 \quad (17)$$

where $Ave.(\overline{\boldsymbol{S}}_{\boldsymbol{r}})$ is the average of normalized similarity vector $\overline{\boldsymbol{S}}_{\boldsymbol{r}}$.

We normalize $\frac{1}{D_r}$ through softmax function:

$$\overline{D}_r = softmax(\frac{1}{D_r}) \quad (18)$$

as the attention scores because higher variance means weaker relevance with surrounding comments.

Finally, we compute the overall neighbor similarity G_{Nsim_r} by the weighted sum of neighbor similarities:

$$G_{Nsim_i} = \sum_{r=1}^{R} \overline{D}_r \cdot Nsim_r \cdot decay(t_{F_r}, t_i) \quad (19)$$

Through sentence-level IVA, we make use of the interactive and real-time properties of TSC data to reduce the impact of noise effectively, and acquire the overall neighbor similarity more accurately.

5 EXPERIMENT

In this section, we first introduce dataset we used in this work in Section 5.1 and the dataset process and evaluation metric in Section 5.2. Then, we evaluate the effectiveness of our proposed method by comparing with 4 baseline methods in Section 5.3. Finally, we visualize the weight of the attention layer to validate the performance of the attention mechanism both in word and sentence level in Section 5.4.

5.1 Data Preparation

The TSC data used in this paper is crawled from Chinese video website Youku [1] and Bilibili [2]. We choose TV-series, movies, and sports as the categories of our data set, since the video plots of these three categories are very likely to be spoiled. These spoilers are relatively simple to be detected and labeled though certain spoiler keywords related to the plots. We gather the data from November 2017 to March 2018, where each TSC has been labeled as a spoiler or not by man-selected keyword and manual inspection.

To label the data, we first summarize the possible spoiler keywords specific to each video's tags and filter out the TSCs that most likely to be spoilers based on keywords. Then three human evaluators check the high-likely spoilers and assign labels to real spoilers according to whether the TSC is related to the latter plots of the video. We treat a certain TSC as a spoiler if two or more evaluators regard it related to the crucial plot. To better understand the insights of this dataset, we conduct preliminary statistic analyses listed as Table 2. The average density of TSCs means the average number of TSCs appear per second.

Table 2: Statistics of spoilers in the time-synchronized comment (TSC) dataset

	TV-series	Movies	Sports
Number of videos	32	480	501
Number of TSCs	337568	591254	588036
Ave. TSCs per video	10549.0	1231.8	1173.7
Number of spoilers	102677	99868	98204
Proportion of spoiler	0.3044	0.1689	0.1670
Ave. Spoilers per video	3208.7	208.1	196.0
Ave. Density of TSCs	4.14	0.9223	0.8687

5.2 Dataset Preprocess and Evaluation Metric

The raw TSC text is full of noise, so we manually remove meaningless TSCs and establish a set of mapping rules for network slang, which will be substituted by their real meaning in the text. For instance, 233... (2 followed by several 3) means laughter, 666... (several 6) means playing games very well, and "front high energy" means terrible plots will happen next in the video. After that, we segment the words and remove the anomaly symbol in TSCs by an open-source Chinese-language processing toolbox Jieba.

Also, timestamps are recorded when users send their TSCs at the video's playback time. To avoid a long time interval between consecutive TSCs and take better advantage of the inner relationships among TSCs, we discard the videos with fewer than 300 TSCs or its density of TSCs is lower than 0.1 comment per second. The final statistics of the data set have been introduced in Table 2. We randomly select 70% of the dataset as the training set, 20% as the test set, and 10% as the validation set.

In our model, we set $R = 5$ and $P = 3$, i.e, in each video, we sample 5 former neighbors of each target TSC and select 3 keyframes. Moreover, the time decay rate β is hyper-parameter need to be

Table 3: F1-score, Precision, and Recall of each method

(a) TV-series

Baselines	Precision	Recall	F1-score
KM	0.443	**0.892**	0.577
LDA	0.563	0.656	0.606
LI-NPP	0.677	0.713	0.695
DN-GAA	0.730	0.798	0.747
SBN	0.782	0.791	0.779
SBN-WT	0.762	0.743	0.754
SBN-IVA	**0.843**	0.856	**0.850**

(b) Movies

Baselines	Precision	Recall	F1-score
KM	0.398	**0.912**	0.545
LDA	0.497	0.604	0.562
LI-NPP	0.654	0.727	0.688
DN-GAA	0.714	0.792	0.751
SBN	0.722	0.833	0.785
SBN-WT	0.708	0.776	0.732
SBN-IVA	**0.753**	0.856	**0.811**

(c) Sports

Baselines	Precision	Recall	F1-score
KM	0.373	0.797	0.509
LDA	0.525	0.589	0.555
LI-NPP	0.668	0.778	0.719
DN-GAA	0.718	0.758	0.738
SBN	0.782	0.809	0.789
SBN-WT	0.722	0.749	0.741
SBN-IVA	**0.810**	**0.841**	**0.825**

decided. In the training progress, we adjust the value of β by the validation set. The initial learning rate of Adam is 0.001, and the word vector dimension is 128. We get the best results when $\beta = 0.15$, and will discuss them in details in Section 5.3.

The prediction of the likelihood of spoilers is a binary classification problem and the positive sample ratio (spoiler ratio) only accounts for 19.83% of the total samples. Therefore, only considering the prediction performance of positive samples can better test the performance of the models. Thus, we use Precision, Recall and F1-score instead of Accuracy-score to measure the performance.

5.3 Comparison of Baseline Methods

To evaluate the performance of our proposed model, we compare our models with the following methods as baselines:

- **KM:** Keyword-matching method is the simplest method, which filters out spoilers according to the actor name in a drama or the match score at a sporting game [8, 22]. We set the prediction result as 1, if a TSC matches spoiler keywords, and 0 otherwise.
- **LDA:** This is a machine-learning method based on Latent Dirichlet Allocation (LDA) used by Guo *et al.* [9] in their

Table 4: F1-score, Precision, and Recall with different IVA parameters

Parameters	R=1			R=2			R=3			R=4		
Metrics	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
P=1	0.489	0.472	0.480	0.525	0.566	0.525	0.537	0.589	0.540	0.545	0.607	0.555
P=2	0.583	0.563	0.561	0.638	0.662	0.649	0.642	0.682	0.652	0.628	0.695	0.673
P=3	0.688	0.733	0.712	0.759	0.762	0.740	0.817	0.797	0.807	0.805	0.811	0.807
P=4	0.730	0.762	0.744	0.770	0.824	0.797	0.822	0.833	0.819	0.829	0.828	0.814
P=5	0.777	0.781	0.769	0.817	0.845	0.841	0.825	0.860	0.846	0.819	0.869	0.850
P=6	0.779	0.808	0.773	0.820	0.853	0.839	0.830	0.854	0.845	0.827	0.855	0.844

work to filter out spoiler using predictive perplexity. We use perplexity to predict the appropriate number of topics, and set it as 20.

- **LI-NPP:** This method introduces location information (LI) and neighborhood plot probability (NPP) to help identify spoilers with contextual information through SVM [11]. We set a linear kernel and complexity constant $C = 1$ in the SVM.
- **DN-GAA:** This is a Deep Neural spoiler detection model (DN) that uses a Genre-Aware Attention mechanism (GAA) [3].
- **SBN:** Our Similarity-Based Network that proposed in Section 4.2.
- **SBN-WT:** Our Similarity-Based Network Without Timestamps. In this model, we don't consider the timestamp and set the decay function (Eq.(10)) equal to 1, which is used to evaluate the contribution of timestamp and decay function.
- **SBN-IVA:** Our Similarity-Based Network works with Interactive Variance Attention mechanism that proposed in Section 4.3.

The baseline methods may reach their best performance with different latent factors. Therefore, we tested various parameters in the range of their work and chose the best prediction performance as final results for each baseline method.

The experiments of all the models are repeated for 10 times, and we use the average values as the final results. The results of F1-score, Precision, and Recall in the categories of TV-series, movies and sports are shown in table 3.

From Table 3, we can conclude that SBN-IVA achieves the best performance on Precision, Recall and F1-Score in the categories of TV-series, movies and sports. Compared to the-state-of-art DN-GAA method proposed by Chang *et al* [3], SBN-IVA enhance the performance 13.8%, 7.99%, 11.8% (average 11.2%) upon F1-Score. In other baselines, keyword-matching (KM) method achieves high Recall and low Precision, since it treats many non-spoiler TSCs as spoilers. Latent Dirichlet Allocation (LDA) based method does not perform well because the LDA model is not suitable to process short texts like TSC data. LI-NPP is an SVM-based method and has better performance than the unsupervised learning methods. DN-GAA is the state-of-the-art method and has the highest performance among the baselines. The results show that the neural network has powerful feature extraction capabilities when there is sufficient data.

Although LI-NPP and DN-GAA have achieved better results than unsupervised learning methods, they do not consider the specific properties of TSC. Our SBN framework takes the interactive and real-time properties of TSCs into considering, and performs better than LI-NPP and DN-GAA. To further evaluate the effect of the real-time property of TSCs, we remove the decay function in SBN (set the decay function Eq.(10) equal to 1), and obtain SBN-WT. Results show that compared with SBN-WT, SBN enhance the performance 3.31%, 7.24%, 6.47% upon F1-Score. This indicates that it is necessary to provide a low weight to those TSCs that has a long time interval from the target TSC because they maybe discuss different topics. Moreover, to evaluate the contribution of IVA mechanism, we compare the SBN with SBN-IVA. Results show that SBN-IVA enhances the performance 9.11%, 3.31%, 4.56% upon F1-Score than SBN, which indicates that the IVA mechanism can reduce the impact of noise efficiently. Our SBN-IVA achieves the best Precision, Recall, and F1-Score, because our model effectively makes use of the interactive and real-time properties of TSCs and reduce the impact of noise.
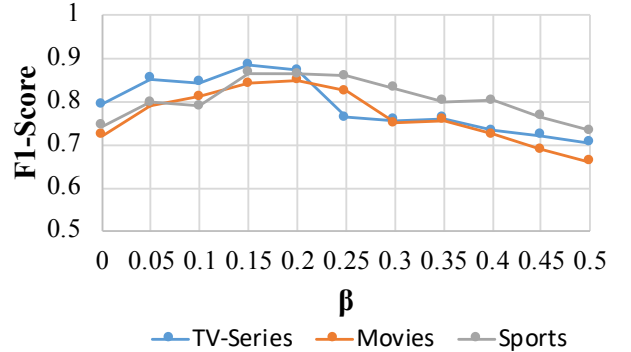

Figure 6: The influence of time decay rate β.

Then, we discuss the influence of time decay rate β on the experimental results. $\beta = 0$ means we do not consider the time decay. When time decay rate β grows, the interactive property of TSCs becomes weaker with the time interval. We change the value of β from 0 to 0.5 with the step size of 0.05 and calculate the F1-score in the categories of TV-series, movies, and sports. As shown in Figure 6, the decay rate β reaches its best performance when $\beta = 0.15$.

Finally, we change the number of former neighbors R and keyframe P to see their influence on the experiment result in the validation

set. The results of F1-score, Precision, and Recall with different IVA parameters are shown in Table 4. We can find the F1-score increase with the number of former neighbors R and keyframe P. This result proves that our application of the interactive and real-time properties of TSCs is correct. When we increase former neighbors, TSCs establish semantics association with more surrounding neighbors. When we increase keyframes, TSCs can be compared with more important plots in the video. According to table 4, we choose $P = 5$ and $R = 3$, and do not continue increasing P and R because this leads to a decline in time efficiency and there is no significant growth in Precision, Recall, and F1-score.

5.4 Visualization of Attention

To validate the performance of our attention mechanism both in word and sentence level, we visualize the weight of the attention layer in Fig. 7. We take a group of consecutive TSCs in a criminal investigation TV Series as an example. The scene of the example is shown in Fig. 1.
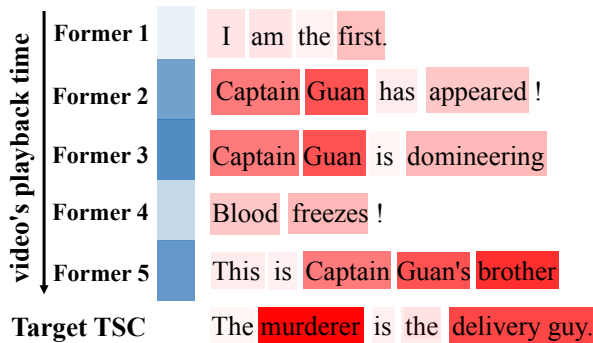

Figure 7: Visualization of attention in word and sentence level.

The TSC in the last line is the target TSC, and the others are its former neighbors in order of timestamps. The blue color denotes the weight of sentences in sentence-level IVA mechanism. The darker the color is, the higher the IVA score the TSC has. Meanwhile, the red color denotes the weight of words in the word-level attentive encoder. The darker the color is, the more likely the word to be a keyword to define a spoiler.

Fig. 7 shows the effectiveness of our attention model. We can see that these TSCs mainly discuss something about Captain Guan (the protagonist in the TV-series). The weight of sentence-level IVA is calculated by the product of semantic similarity and time decay. The Former 2, Former 3, and Former 5 have high IVA scores, which means they are related to the main topic. In contrast, Former 1 and Former 4 mainly express the users' emotion deviated to the central topic, therefore they should be classified as noise and have a small weight.

In the word-level, Fig 7 shows that our attentive word encoder can select the words carrying strong spoiler tendencies such as "Captain", "Guan", "brother", "murderer", and "delivery guy". Other unimportant words such as "am", "is", "this", "has", are discarded. That is, the word attentive encoder selects spoiler keyword when calculating the sequence vector through the weighted sum.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel similarity-based spoiler detection model with the dataset of TSCs. To extract spoiler keyword accurately based on the short-text, interactive, real-time and noise properties of TSCs, we introduce attention mechanism both in word encoder and sentence invariance level in our model. The Word-Level Attentive Encoder filters the spoiler keyword out, and Interactive Variance Attention (IVA) calculate the similarity invariance of comments and assign a low weight to noise comments. In this way, we integrate the short-text, interactive and real-time properties of TSCs in the deep framework, and effectively weaken the impact of noise comments. Then the likelihood of TSCs to be spoilers is predicted accurately through the difference of neighbor similarity and keyframe similarity. Extensive experiments on real-world dataset proved that our model with IVA outperformed the existing spoiler detection method in F1-score, Precision, and Recall.

Our pioneer work in this paper points out the promising future directions in online spoiler detection, and this is the first step towards our goal in online spoiler detection. In the future, we will seek to explore the following directions: (1) We will take other information about the online video, such as visual and audio features to improve the performance of the model. (2) In most cases, the spoilers are sparse. However, some extreme users will continue to post the spoilers to occupy the screen. In this extreme case, our model cannot solve it well (since all the neighbors are the same and post by extreme users). Therefore, we will incorporate user ID into the model to identify extreme spoilers in the next step. (3) We will design an online spoiler detection model to detect video spoilers in real time.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *arXiv preprint arXiv:1409.0473*.

[2] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*. 169–214.

[3] Buru Chang, Hyunjae Kim, Raehyun Kim, Deahan Kim, and Jaewoo Kang. 2018. A Deep Neural Spoiler Detection Model Using a Genre-Aware Attention Mechanism. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 183–195.

[4] Xinpeng Chen, Jingyuan Chen, Lin Ma, Jian Yao, Wei Liu, Jiebo Luo, and Tong Zhang. 2018. Fine-grained video attractiveness prediction using multimodal deep learning on a large real-world dataset. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 671–678.

[5] Xu Chen, Yongfeng Zhang, Qingyao Ai, Hongteng Xu, Junchi Yan, and Zheng Qin. 2017. Personalized key frame recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 315–324.

[6] Yixin Chen. 2017. Automatic Annotation of Online Multimedia Data with Similarity Relations. *McGill University* (2017).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[8] Jennifer Golbeck. 2012. The twitter mute button: a web filtering challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2755–2758.

[9] Sheng Guo and Naren Ramakrishnan. 2010. Finding the storyteller: automatic spoiler tagging using linguistic cues. In *Proceedings of the 23rd International Conference on Computational Linguistics*. ACL, 412–420.

[10] Ming He, Yong Ge, Le Wu, Enhong Chen, and Chang Tan. 2016. Predicting the popularity of danmu-enabled videos: A multi-factor view. In *International Conference on Database Systems for Advanced Applications*. Springer, 351–366.

[11] Yoshinori Hijikata, Hidenari Iwai, and Shogo Nishida. 2016. Context-Based Plot Detection from Online Review Comments for Preventing Spoilers. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 57–65.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Hidenari Iwai, Yoshinori Hijikata, Kaori Ikeda, and Shogo Nishida. 2014. Sentence-based plot classification for online review comments. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1. IEEE, 245–253.

[14] Sungho Jeon, Sungchul Kim, and Hwanjo Yu. 2013. DonâĂŹt be spoiled by your friends: spoiler detection in TV program tweets. In *Seventh International AAAI Conference on Weblogs and Social Media*.

[15] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*. 1378–1387.

[16] Jiangfeng Li, Zhenyu Liao, Chenxi Zhang, and Jing Wang. 2016. Event detection on online videos using crowdsourced time-sync comment. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*. IEEE, 52–57.

[17] Zhenyu Liao, Yikun Xian, Xiao Yang, Qinpei Zhao, Chenxi Zhang, and Jiangfeng Li. 2018. TSCSet: A Crowdsourced Time-Sync Comment Dataset for Exploration of User Experience Improvement. In *23rd International Conference on Intelligent User Interfaces*. ACM, 641–652.

[18] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *arXiv preprint arXiv:1508.04025*.

[19] Guangyi Lv, Tong Xu, Enhong Chen, Qi Liu, and Yi Zheng. [n. d.]. Reading the Videos: Temporal Labeling for Crowdsourced Time-Sync Videos Based on Semantic Embedding. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[20] Kyosuke Maeda, Yoshinori Hijikata, and Satoshi Nakamura. 2016. A basic study on spoiler detection from review comments using story documents. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 572–577.

[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[22] Satoshi Nakamura and Katsumi Tanaka. 2007. Temporal filtering system to reduce the risk of spoiling a user's enjoyment. In *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 345–348.

[23] Qing Ping. 2018. Video recommendation using crowdsourced time-sync comments. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 568–572.

[24] Qing Ping and Chaomei Chen. 2017. Video Highlights Detection and Summarization with Lag-Calibration based on Concept-Emotion Mapping of Crowd-sourced Time-Sync Comments. *arXiv preprint arXiv:1708.02210* (2017).

[25] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* (2018).

[26] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. 2440–2448.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[28] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in neural information processing systems*. 2773–2781.

[29] Bin Wu, Erheng Zhong, Ben Tan, Andrew Horner, and Qiang Yang. 2014. Crowd-sourced time-sync video tagging using temporal and personalized topic modeling. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 721–730.

[30] Yikun Xian, Jiangfeng Li, Chenxi Zhang, and Zhenyu Liao. 2015. Video highlight shot extraction with time-sync comment. In *Proceedings of the 7th International Workshop on Hot Topics in Planet-scale mObile computing and online Social neTworking*. ACM, 31–36.

[31] Linli Xu and Chao Zhang. [n. d.]. Bridging Video Content and Comments: Synchronized Video Description with Temporal Summarization of Crowdsourced Time-Sync Comments. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[32] Wenmian Yang, Wenyuan Gao, Xiaojie Zhou, Weijia Jia, Shaohua Zhang, and Yutao Luo. 2019. Herding Effect based Attention for Personalized Time-Sync Video Recommendation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE.

[33] Wenmian Yang, Weijia Jia, XIaojie Zhou, and Yutao Luo. 2019. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

[34] Wenmian Yang, Na Ruan, Wenyuan Gao, Kun Wang, Wensheng Ran, and Weijia Jia. 2017. Crowdsourced time-sync video tagging using semantic association graph. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 547–552.

[35] Wenmian Yang, Kun Wang, Na Ruan, Wenyuan Gao, Weijia Jia, Wei Zhao, Nan Liu, and Yunyong Zhang. 2019. Time-sync Video Tag Extraction Using Semantic Association Graph. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13(4), 37 (2019).

[36] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Video description generation incorporating spatio-temporal features and a soft-attention mechanism. *arXiv preprint arXiv:1502.08029* (2015).

[37] Yongjian You, Weijia Jia, Tianyi Liu, and Wenmian Yang. 2019. Improving Abstractive Document Summarization with Salient Information Modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 2132–2141.