Sentiment Classification of Time-Sync Comments: A Semi-Supervised Hierarchical Deep Learning Method

Renzhi Gao[a], Xiaoyu Yao[a], Zhao Wang[a*], Mohammad Zoynul Abedin[b]

[a] School of Management, Hefei University of Technology, Hefei, Anhui, P.R. China

[b] School of Management, Swansea University, Bay Campus, Fabian Way, SA1 8EN Swansea, UK

Abstract

Time-sync comment (TSC) has emerged as a new type of textual comment for real-time user interactions on online video platforms. The sentiment classification of TSCs provides considerable potential for platforms to optimize operation strategies but inevitably faces great challenges due to the TSCs' often uninformative and informal text. Considering the contextual dependency among TSCs posted within the same video clip, this study posits that contextual TSCs may benefit the sentiment classification of a target TSC. To address the challenges of leveraging contextual TSCs, such as their semantic representation and fusion, we propose a semi-supervised hierarchical deep learning method for the sentiment classification of TSCs. We design a hierarchical architecture to capture the semantics of TSCs at the word, comment, and context levels. Considering the varying importance of words and comments, we also design attention mechanisms to focus on important sentiment information and fuse semantic representations. Empirical evaluation shows that the proposed method outperforms benchmarked sentiment classification methods. This study advances our knowledge of contextual information indicative of TSC sentiment, and contributes to improving the service operation of online video platforms.

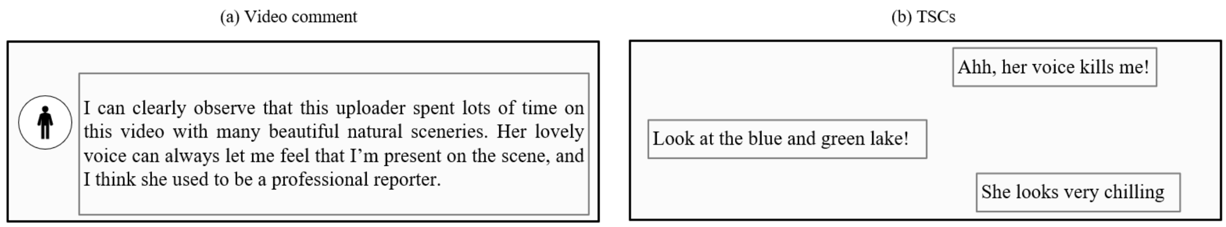
Keywords: OR in marketing; Time-sync comment; Sentiment classification; Contextual dependency; Semi-supervised deep learning

[*] Corresponding author. Address: School of Management, Hefei University of Technology, No.193, Tunxi Road, Hefei, Anhui 230009, P.R. China.
E-mail addresses: 2019210405@mail.hfut.edu.cn (Renzhi Gao); 2019210398@mail.hfut.edu.cn (Xiaoyu Yao); xcwangzhao@163.com (Zhao Wang); m.z.abedin@swansea.ac.uk (Mohammad Zoynul Abedin)

1. Introduction

With the proliferation of the internet and mobile devices, digital video markets and online video platforms have witnessed rapid growth (Chakraborty et al., 2021; Wu & Chiu, 2023). Time-sync comment (TSC), also known as danmaku, is the outcome of these online video platforms (Li & Guo, 2021b; Xu & Zhang, 2017). This new comment type is the latest innovation in the rapid progression of features that marketing analytics must adapt to. TSC provides a real-time interaction mechanism that allows video viewers to express their ideas and emotions about specific video content, with the posted TSCs appearing immediately alongside the video (Zhou et al., 2019). Compared to other evaluative information such as a satisfaction rating that reflects viewers' evaluation of the whole video, TSCs reflect viewers' attitudes toward certain video clips (i.e., certain sections of a video), and exist as more fine-grained feedback information. Hence, the sentiment classification of TSCs is crucial for refining the operation management of online video platforms, and thus creating added value for various stakeholders. For video viewers, identifying the sentiment of TSCs can help online video platforms understand viewer preferences and implement personalized video clip recommendations, which are critical to improving user experience (Jiang et al., 2020). Viewers are directly pushed to potentially interesting video clips and can quickly find their favorite people and scenes without watching the entire video. For video creators, identifying the TSC sentiments allows them to better understand current market demands, which is a key strategy for content providers (Chong et al., 2016). In a video, viewers' sentiments toward video clips are important guidance as to which clips may be outdated and need to be improved, and which styles should be continued. Moreover, viewer sentiment acts as a quality indicator of video clips and provides an effective foundation for subdividing the management and control of video quality (Tarí et al., 2007). Video platforms can introduce flexible incentive measures to motivate excellent video creators based on video clip quality. The video platform itself can benefit by implementing membership or fee systems for certain high-quality video clips instead of the entire video, which could attract more viewers and expand profitability.

Given its distinctive characteristics, the sentiment classification of TSCs is nontrivial and even more difficult than that of other types of online comments. First, TSCs are generally posted by viewers quickly, and thus the text length is much shorter than that of traditional comments, which means that the semantic information from a certain TSC is limited or even ambiguous (He et al., 2018). Second, TSCs usually contain popular phrases and Internet slang related to particular fields, the informal expression of TSCs makes semantic representation intractable (Chen et al., 2022). For example, Figure 1 gives one video comment and three TSCs from the same online video. The figure shows that, compared with the video comment, each TSC contains insufficient semantic information to understand the sentiment.



(a) Video comment

I can clearly observe that this uploader spent lots of time on this video with many beautiful natural sceneries. Her lovely voice can always let me feel that I'm present on the scene, and I think she used to be a professional reporter.

(b) TSCs

Ahh, her voice kills me!

Look at the blue and green lake!

She looks very chilling

Figure 1. A real-life example of a comment and TSCs from the same video

The characteristic contextual dependency of TSCs provides a new pathway for sentiment classification. As a comment on real-time video content, the TSC content is highly correlated to the content of the corresponding video clip (Yang et al., 2019a). In this case, TSCs posted in the same period of time (i.e., neighboring TSCs) comment on similar objects, and thus may contain similar sentiment polarities. Moreover, viewers who post TSCs can concurrently browse other TSCs synchronously with the video, and thus their opinions and emotions are inevitably influenced by the TSCs already posted (Liao et al., 2020). Conversely, a new TSC may affect subsequent TSCs. Under the influence of nearby TSCs, especially those that strike a responsive chord, viewers may feel similar opinions and post a new TSC promptly in response to sympathetic ones. Take the following five successive TSCs from an online music variety show as an example:

Oh my God! The song is sure to be straight fire.

Alas, but the lyrics do not form a whole.

Awesome! The girls' singing sounds unexpected this time.

Yes! All the girls sing very well!

Why the lyrics are completely irrelevant!

Taking the third TSC as the target TSC, the other TSCs can be regarded as contextual TSCs. While the target TSC expresses strong sentiment regarding the aspect of singing using the words "awesome" and "unexpected," its sentiment polarity may be ambiguous; that is, it could express a positive sentiment in a normal manner or a negative sentiment in a sarcastic manner. Fortunately, the first and fourth TSCs talk about similar topics as the target TSC (i.e., the singing) and show positive sentiments. This background information provides clearer evidence to infer that the target TSC has a positive sentiment. In this regard, contextual TSCs have considerable potential to enhance the performance of sentiment classification of the target TSC as a type of auxiliary information.

However, the unique text characteristics and diverse posting scenarios of TSCs bring great challenges for using contextual TSCs, among which we identify two essential ones. First, incorporating contextual information requires not only extracting the semantics of each TSC, but also capturing complex contextual relationships among multiple neighboring TSCs. The challenge lies in capturing various syntactic and semantic relationships from different levels for TSC text representation. Second, with the particular and individual preferences of viewers, different viewers may post TSCs to comment on different objects in the same video clip, and thus the sentiments of contextual TSCs are not always the same as that of the target TSC. This scenario is reflected in the above example, in which the second and fifth TSCs discuss the lyrics and express negative sentiments. Therefore, the challenge also lies in focusing on useful contextual TSCs that enrich our understanding with complementary semantics, and suppressing irrelevant contextual TSCs.

Existing research provides rich insight into machine learning and deep learning methods for the sentiment classification of a TSC (Onan et al., 2016; Wang et al., 2021b). However, while the existing sentiment classification methods can be useful, they rarely specifically address TSCs. In addition, most existing methods focus on learning the features of target comment samples, without the ability to adaptively capture contextual TSC information, and cannot address the characteristics of TSCs (Chen et al., 2019).

To fill this gap, this study proposes a semi-supervised hierarchical deep learning method (called SHDL here) for the sentiment classification of TSCs. SHDL is an innovative operations research application in the marketing domain. Specifically, to capture contextual semantic information, we propose a hierarchical deep learning architecture for the semantic representation of TSCs at the word, comment, and context levels. The word-level and comment-level representations are developed to extract semantics from a single TSC, and the context-level representation is further developed to extract effective contextual semantics from multiple contextual TSCs. Moreover, considering the heterogeneous significance of different words and contextual TSCs in sentiment classification, we design two attention mechanisms for adaptive semantic fusion. The word-level attention mechanism is designed to focus on important words during the comment-level representation, and the comment-level attention mechanism focuses on key contextual information during the context-level representation.

We have evaluated the proposed method using a real TSC dataset collected from a major TSC-enhanced online video platform (i.e., Bilibili). We compared SHDL with eight representative sentiment classification methods from the families of both machine learning and deep learning. Empirical results show that SHDL significantly outperformed all benchmarked methods in terms of all performance metrics. The ablation study also shows that all design artifacts in SHDL have performance-enhancing effects on the sentiment classification of TSCs, and that capturing the temporal correlation among TSCs is particularly significant. Moreover, the representation performance analysis illustrates how SHDL improves the performance of TSC sentiment classification with the ability to effectively use contextual information.

The contributions of this study are fourfold. First, to the best of our knowledge, this is the first study that leverages contextual information (i.e., contextual TSCs) to classify the sentiment of a TSC. In contrast with the existing literature, which merely considers a single TSC sample, we use multiple samples and utilize contextual information in a semi-supervised way. Second, we propose a deep learning–based hierarchical representation architecture for generating the semantics of TSCs at the word, comment, and context levels, which can effectively capture heterogeneous semantics at different levels (i.e., a single word, a single TSC, and a group of neighboring TSCs). Compared to existing structures that extract semantic levels only within comments, we extend the extracted semantic levels beyond the comments. Third, we design two attention mechanisms to adaptively focus on important words and comments for the fusion of the semantic representation at different levels, and the weighting process of the two attention mechanisms is different. Fourth, the proposed TSC sentiment classification method provides significant practical implications for video platforms to refine their operation strategies. The accurate identification of TSC sentiment enables platforms to rapidly show viewers better-suited video clips, enables video creators to achieve more sophisticated video production, and can improve the profit model of the video platforms.

The remainder of this paper is organized as follows. In the next section, we summarize the recent literature on TSC research and sentiment classification. Then in Section 3, we provide a description of the modeling approach. We describe the empirical evaluation in Section 4 before presenting the results in Section 5. Finally, we conclude the study by summarizing our contributions and discussing future research directions in Section 6.

2. Literature review

2.1. Time-sync comments in online videos

TSCs originated from the Japanese video website Niconico, which initially represented the Animation, Comic, and Game (ACG) subculture (Xi et al., 2021). In contrast with regular comments, TSCs synchronously moved over the ACG videos in the form of subtitles as soon as viewers posted them. This novel and timely comment mechanism became popular with ACG audiences and was quickly accepted and enjoyed by mainstream culture. At present, numerous worldwide mainstream video platforms support the TSC function, and the introduction of TSCs has had profound impacts on various stakeholders, including video viewers, video creators, and video platforms.

From the perspective of video viewers, TSCs serve as a communication medium between viewers and videos, which improves the degree of viewer involvement (Lv et al., 2019). Posting TSCs helps a viewer concentrate on a video and also improves their retention of the video content and sense of fulfillment after watching it (Li et al., 2021a). TSCs can complement the video content, and video viewers might see diverse TSCs, such as several copies of an alert like “dragons ahead” and explanations provided by the expert viewers, which can greatly improve the viewing experience. Meanwhile, existing TSCs affect the TSC-posting behavior of subsequent viewers, as explained by the “herding” effect wherein viewers who are inclined to post TSCs can be influenced by observed TSCs (He et al., 2018). Furthermore, viewers who often post TSCs to express their attitudes toward videos can enjoy more precise video clip push services through personalized video recommendation algorithms based on the sentiment analysis of their TSCs (Bai et al., 2021).

For video creators, TSCs in their videos allow them to observe viewer feedback and better cater to social hotspots and viewing demands in future videos. Considering that TSCs give timely feedback on video clips, which may be positive or negative, careful creators can absorb the opinions and then specifically improve the video production. Moreover, TSCs can intuitively benefit creators during live broadcasting by influencing the viewers' gifting behavior through stimulation and social density (Zhou et al., 2019).

For video platforms, TSC has brought about a new mechanism for video quality assessment, by which a series of operational strategies can be modified. For example, some unruly viewers may spoil the video content through TSCs, so platforms can use automatic detection technologies to block spoilers and penalize offenders (Yang et al., 2019a). Platforms can encourage and award excellent video creators according to video quality, and the awards can be fine-grained (i.e., awards given to specific clips of a video) with the help of approaches such as highlight detection (Liaw & Dai, 2020). Such auxiliary techniques using TSCs can help platforms upgrade their monetization strategies by implementing membership or fee systems for viewers to see certain high-quality video clips instead of whole videos.

2.2. Sentiment classification of online comments

The popularity of social media has broadened the number and variety of online channels in which Internet users can express themselves with comments. These comments usually contain abundant opinion and sentiment information, and are valuable for service providers to improve service quality (Meire et al., 2016; Xia et al., 2021). Since TSC is a new type of online comment, the sentiment classification of TSCs is an urgent matter to fully exploit its potential value. However, the existing research on the sentiment classification of TSCs is scant, which is an important motivation for this study. In this section, we expand the discussion of sentiment classification methods from TSC to online comments to provide a more comprehensive overview of existing methods. The task of classifying the sentiments of online comments can be considered as a text classification problem because of the involvement of several operations that ultimately classify a given piece of text to show either a positive or negative sentiment. The methods of this classification can be divided into two categories: lexicon-based and model-based sentiment classification.

Lexicon-based sentiment classification usually builds auxiliary lexical resources that link the words to corresponding sentiment polarities by scoring (Cruz et al., 2014). For example, Deng et al. proposed a method to adapt existing sentiment lexicons for domain-specific sentiment classification (Deng et al., 2017). However, given that several words have multiple meanings and senses, building a pervasive lexical resource is difficult. The lexical resources are usually constructed based on specific domains and scenarios, which largely limit the flexibility of applying lexicon-based methods (Han et al., 2020).

Model-based sentiment classification aims to train sentiment classification classifiers by using machine learning models (Agarwal et al., 2019). The traditional machine learning models implemented for sentiment classification include naïve Bayes (NB) (Chan & Im, 2022), logistic regression (LR) (Onan et al., 2016), support vector machine (SVM) (Ye et al., 2009), and random forest (RF) (Parmar et al., 2014). Regarding sentiment analysis as a classification problem, training sets are first built by manually labeling a portion of the comment text, and then learning the features from the training data to construct classification models. Finally, the model obtained from the training period is used to classify the test data with its unknown sentiments. For example, Ghaddar and Naoum-Sawaya (2018) improved the high-dimensional online comment data classification efficiency of SVM, offering benefits for optimal decision-making. However, most of the above methods are based on the bag-of-words model, where words in the comments are independent and their significant sequential dependency is ignored (Tsai & Wang, 2017).

As a mainstream branch of machine learning with powerful representation ability, deep learning models with more complex structures can directly learn abstract features from comment text and implement end-to-end sentiment classification (Yang et al., 2022). A recurrent neural network (RNN) is widely used for text mining because of its advantages in capturing the sequential relationships of words owing to the recursive structure (Lai et al., 2015). In particular, two well-known variants of RNN, long short-term memory (LSTM) and gated recurrent unit (GRU), have attracted much attention for text data processing owing to their capacity to handle long series (Kratzwald et al., 2018; Kriebel & Stitz, 2022). Meanwhile, convolutional neural network (CNN) is another popular deep learning model that is used for text classification. With unique convolutional and pooling operations, CNN has relatively low computational costs while providing an excellent feature representation ability. In particular, TextCNN, proposed by Kim, learns the n-gram feature representation from sentences via 1D convolution, which has shown excellent performance in short-text classification (Kim, 2014). Moreover, CNN can be combined with LSTM to use the advantages of both the recurrent structure and convolutional neural models to improve sentiment classification results (Ankita et al., 2022). By combining the recurrent structure and max-pooling layer, Lai et al. (2015) proposed the recurrent convolutional neural network (RCNN), where a recurrent structure is applied to capture sequence information and a max-pooling layer is used to catch the key components in pieces of text. In addition, the attention mechanism in deep learning is currently a research hotspot for natural language processing tasks, which allows a focus on relevant information and ignores irrelevant information with efficient computing (Wang et al., 2021b). For instance, Xu et al. (2022) combined an attention-based model and transfer learning to enhance the performance of aspect-level sentiment classification, where the attention mechanism was developed to extract important features from the sequence according to their weight distributions.

2.3. Research gaps

As summarized in Table 1, previous studies have pointed out diverse representative methods for the sentiment classification of online comments, including machine learning and deep learning methods. While existing sentiment classification methods have used semantic representations and attention mechanisms, most of them focus on semantic representation and attention at the word and comment levels. It is important to clarify that in our study, the term “context” specifically refers to the TSCs that appear in proximity to a given target TSC within the corpus. Although in other methods, such as Word2Vec, “context” may refer to the nearby words used for model training, we do not adopt this definition in our study. Instead, we focus on the TSCs surrounding the target TSC as our reference for context analysis. As discussed earlier, contextual TSCs have considerable potential to enhance the performance of the sentiment classification of the target TSC, but how to identify and extract effective semantic information from contextual TSCs is still an open and challenging topic. We strive to bridge this research gap by proposing a TSC sentiment classification method (SHDL) based on semi-supervised hierarchical deep learning.

Table 1. Comparison of SHDL with existing studies on sentiment classification of online comments

Study	Method	Semantic representation	Use of attention mechanism	Use of contextual information
Onan et al., 2016	NB, LR	Word-level	No	No
Ghaddar & Naoum-Sawaya, 2018	SVM	Word-level	No	No
Parmar et al., 2014	RF	Word-level	No	No
Kim, 2014	TextCNN	Word-level, comment-level	No	No
Kratzwald et al., 2018	BiRNN	Word-level, comment-level	No	No
Lai et al., 2015	RCNN	Word-level, comment-level	No	No
Wang et al., 2021b	BiGRU-Attention	Word-level, sentence-level, document-level	Sentence-level attention	No
SHDL (this study)	CNN, BiGRU, Attention	Word-level, comment-level, context-level	Word-level attention, comment-level attention	Yes

3. Proposed sentiment classification method

To address the sentiment classification of TSCs, we propose a semi-supervised hierarchical deep learning method named SHDL, whose overall framework is illustrated in Figure 2. The method consists of word-, comment-, and context-level representations, and sentiment polarity classification. In the word-level representation, we augment the target TSC with its contextual TSCs and generate their word embeddings in parallel. In the comment-level representation, we develop concurrent CNN with word-level attention branches to learn the representation vector for each TSC. In the context-level representation, we develop a bidirectional GRU (BiGRU) with comment-level attention to learn the representation vector for the TSC context. Finally, the obtained context vector representation is used to identify the sentiment of the target TSC. The proposed method is an innovative application of operations research in the marketing domain.

The distinctive characteristics of TSCs, including their brevity and informal expression, pose a critical challenge for sentiment analysis; that is, TSCs have insufficient semantic information. The idea behind the proposed method is to augment the semantic representation of the target TSC by incorporating context information from the surrounding TSCs, and accordingly improve the performance of the sentiment analysis of TSCs. Specifically, we propose a semi-supervised hierarchical learning framework to use contextual TSCs and learn the multilevel semantics of the TSC context. Moreover, we design two attention mechanisms to focus on important information for adaptive semantic representation fusion. Those design artifacts help the proposed method obtain additional contextual information for understanding the target TSC and address the problem of insufficient semantic information of one single TSC.

It should be noted that our proposed method introduces new elements to the marketing analytics domain, namely the semi-supervised hierarchical learning framework and two attention mechanisms. These novel components enhance the existing sentiment analysis methodologies in the field. However, certain elements such as word embedding, CNN, and BiGRU are derived from established methods (Mikolov et al., 2013; Kim, 2014; Cho et al., 2014). The incorporation of these existing elements, alongside the introduction of new components, forms the foundation of our proposed method.

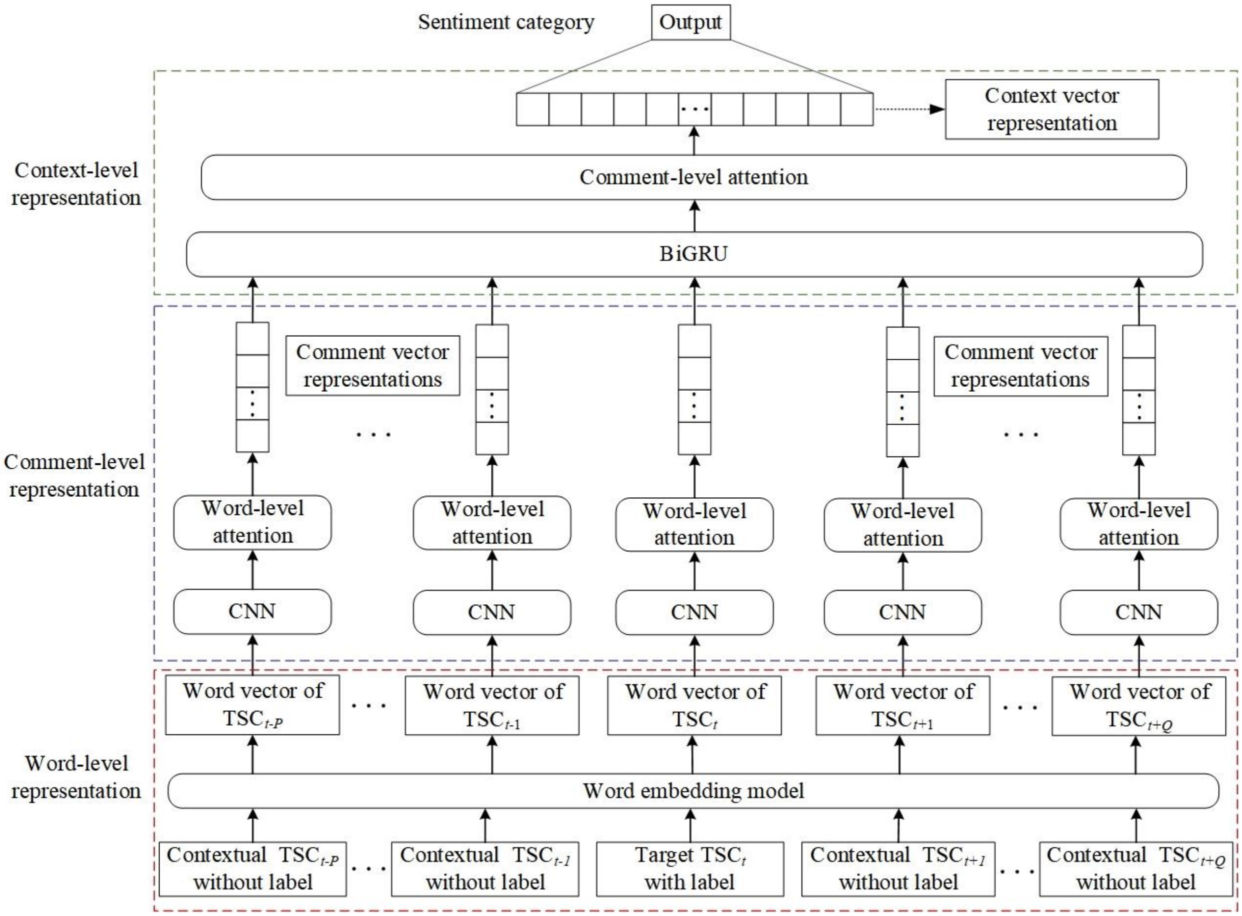

Sentiment category
Output
Context vector representation
Context-level representation
Comment-level attention
BiGRU
Comment vector representations
Comment vector representations
Comment-level representation
Word-level attention
CNN
Word vector of TSC$_{t-P}$
Word vector of TSC$_{t-1}$
Word vector of TSC$_t$
Word vector of TSC$_{t+1}$
Word vector of TSC$_{t+Q}$
Word-level representation
Word embedding model
Contextual TSC$_{t-P}$ without label
Contextual TSC$_{t-1}$ without label
Target TSC$_t$ with label
Contextual TSC$_{t+1}$ without label
Contextual TSC$_{t+Q}$ without label

Figure 2. Overall SHDL framework

3.1. Word-level representation

In practice, video viewers who tend to post TSCs are influenced by both specific video clips and neighboring TSCs. The content of a TSC is correlated with nearby TSCs that occur within the same timeframe. Thus, incorporating the semantic information of contextual TSCs is conducive to identifying the sentiment of the target TSC. In the word-level representation of SHDL, we first augment the target TSC with its contextual TSCs by a contextual window; that is, the original single TSC is augmented into a context sample composed of multiple neighboring TSCs.

Assume that TSC data $D = [d_1, d_2, d_3, \dots, d_t, \dots, d_T]$ appear in the video for a continuous period of time, where d_t represents the *t*th TSC sorted by occurrence time and *T* is the total number of TSCs. To classify the sentiment of the *t*th TSC, the *t*th TSC is augmented by its contextual TSCs within the contextual window, and the augmented input sample is a TSC matrix, which can be denoted as $x_t = [d_{t-P}, \dots, d_{t-1}, d_t, d_{t+1}, \dots, d_{t+Q}]$. Note that the augmented TSC matrix maintains the sequence of the various TSCs as shown in Figure 2. P and Q represent the numbers of contextual TSCs that are posted before and after the target TSC, respectively, and are adjustable parameters that can be different, and $P + Q + 1$ is the width of the contextual window. In this way, to identify the sentiment of a TSC, a total of $P + Q + 1$ TSCs are used as the model input. The target and contextual TSCs are fed into the model to carry out collaborative semantic representation, and ultimately, the representation vector of the augmented TSC context is obtained for sentiment classification.

In practical applications, constructing the sentiment classification model would require labeling massive amounts of TSC data, which is time-consuming, laborious, and costly. In fact, we focus on the semantic information of the contextual TSCs, instead of their sentiment labels. Although multiple TSCs are used as input, only the label of the target TSC is required for our model training. For the augmented TSC context sample, the proposed deep learning model does not require labels from each TSC for supervised learning, but only that of the target TSC without those of its contextual TSCs. Rather, unlabeled contextual TSCs are combined with the labeled target TSC for cooperative optimization in a semi-supervised learning process. The contextual TSCs and target TSC cooperate to complete the forward propagation and jointly obtain the feature embedding that represents the entire input for the sentiment classification of the target TSC. At this time, the classification error is calculated through the predicted value and the label of the target TSC. Through this semi-supervised learning approach, extensive unlabeled data can be used for modeling, and the huge workload caused by marking massive amounts of data is alleviated.

After the sample augmentation, the TSCs are all input into the word embedding model to generate the word vectors that map words onto a real-valued vector space. Because TSC text has the characteristics of domain relevance and contains specific expressions, directly using pre-trained word embedding models trained on existing corpora to generate word embeddings is not applicable. There are two potential solutions to address this issue: one is to use a TSC corpus to train conventional word embedding models (such as Word2Vec (Mikolov et al., 2013)), and the other is to use TSC data for fine-tuning on large-scale pre-trained language models (such as BERT (Kenton et al., 2019)). Considering the significant domain differences that exist between TSC data and a pre-trained corpus, as well as potential computational challenges associated with fine-tuning large-scale pre-trained models. Hence, we opted for the former method, in which we train the Word2Vec model using our collected TSC corpus. Word2Vec uses local semantic relationships and is relatively easier to train, but it ignores the relationships between words inside the local window and those outside it, which can be addressed in the subsequent structure of our method. For the training algorithm, we choose the skip-gram of Word2Vec, which uses a central word as the input of a classifier with a continuous projection layer to predict the words in a certain range before and after it (Mikolov et al., 2013). Given a sequence of training words $w_1, w_2, w_3, \dots, w_L$ and window size k, the skip-gram model aims to maximize the probabilities of generating all background words for any central word by minimizing the following loss functions:

$$Loss_w = -\frac{1}{L}\prod_{l=1}^{L}\prod_{-k\leq j\leq k, j\neq 0}\log p(w^{l+j}|w^{(l)}) \quad (1)$$

Given that TSCs contain numerous words in special fields and Internet slang, we construct a specific dictionary for segmenting the TSC text with a word segmentation tool. Specifically, we enhance the functionality of the Jieba word segmentation tool by supplementing its built-in dictionary with 372 domain-specific words. These additional words are curated based on the collected data, including names of individuals, unique appellations, and Internet catchphrases. Following the expansion of the dictionary, we employ Jieba for word segmentation, using its enhanced capabilities for our analysis.

Then, the segmented text is used to train word embeddings by using Word2Vec, the length of the word sequence is fixed, and positions without words are padded with zero. In this way, the word-level representation vectors of TSCs are obtained, and the word vectors of the *t*th TSC context sample can be denoted as $W_t = [w_{t-P}, \dots, w_{t-1}, w_t, w_{t+1}, \dots, w_{t+Q}]$, where $w_t = [w_{t,1}, w_{t,2}, w_{t,3}, \dots, w_{t,L}] \in R^{L\times S}$, L is the length of the word vectors, and *S* is the dimension size of the word vectors.

3.2. Comment-level representation

Given the fast response time of those posting TSCs, the TSC text length is typically short. A TSC is a concise sentence composed of several words and thus can be suitably addressed by CNN, a widely used deep learning model for short text (Chen et al., 2019; Kim, 2014). CNN has powerful modeling capabilities in automatically extracting abstract feature representations from text data with fewer parameters. Moreover, with the convolution filters that are applied to local features, CNN can capture short-distance dependencies in comments. However, ordinary CNN treats every word equally and ignores their different values. In TSC text, words in different positions have different effects on the TSC sentiment; several keywords may play a decisive role, while others may matter little. For instance, emotional adjectives and evaluative words are highly related to the sentiment, while words such as a character's name, personal pronouns, and common conjunctions hardly have an influence. Therefore, we develop a CNN with word-level attention to learn comment-level representation vectors for each TSC, where the designed attention can assess the importance of different words and fuse their representations in different positions. Figure 3 illustrates the detailed structure of the comment-level representation in SHDL.

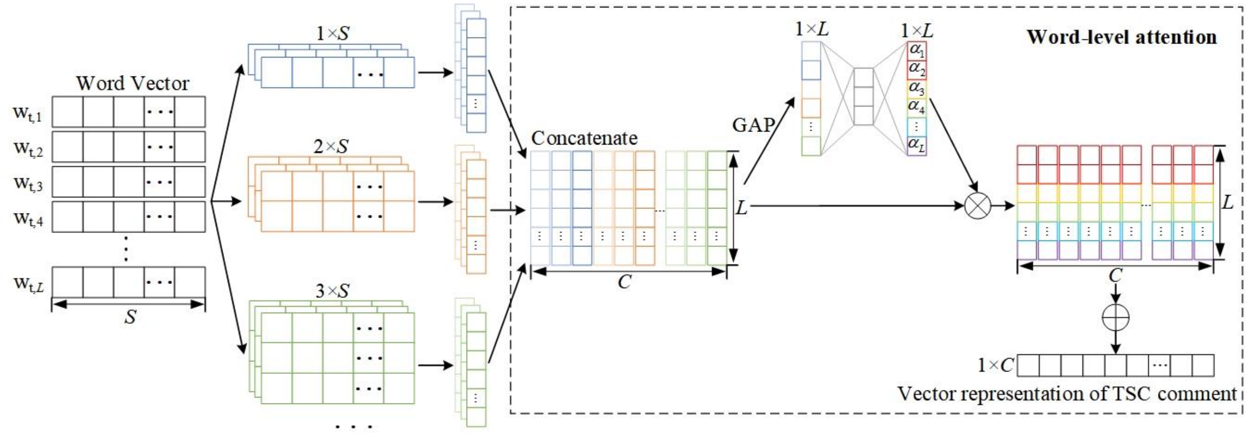


Figure 3. Detailed structure of comment-level representation in SHDL

Given the word vectors of the augmented TSC context $W_t = [w_{t-P}, \dots, w_{t-1}, w_t, w_{t+1}, \dots, w_{t+Q}]$, the word vectors of each TSC are fed into the same CNN with the word-level attention structure to learn their comment-level representation vectors in parallel. As shown in Figure 3, we use multiple convolution kernels with different scales on the word vectors to extract multiscale dependencies within a TSC. Specifically, the width of the convolution kernels is the same as the size of inputs *S*, which ensures that the kernel slides sequentially in the direction of TSC length and can simultaneously process complete information of one or several words. The convolution kernels have various heights, which are recorded as $g \in [1,2,3, \dots, G]$, where *G* is the number of scales. The kernel heights signify the sizes of the word windows. For example, $g = 1$ represents the situation in which the convolution operation maps features for the current word, and is a necessary value because several TSCs contain only one word. Then, $g = 3$ means that the convolution operation maps features for the current word and the previous two words. The kernels with different scales carry out the convolution operation in parallel, and multiple kernels are used for each scale to capture rich semantic features. In this paper, we use three scales, with sizes of 1, 2, and 3. Meanwhile, padding is used to ensure the consistency of feature dimensions from different kernel sizes. For the word vectors w_t, the calculation of CNN can be expressed as follows:

$$f_{t,g} = \text{LeakyReLU}(W_g \otimes w_t + b_g) \tag{2}$$

where $\otimes$ represents the convolution operation, W_g and b_g are the convolution parameters and bias of the *g*th scale, respectively, and $f_{t,g}$ is the obtained features. LeakyReLU is the nonlinear activation function used to filter useful information.

The feature vectors obtained by CNN with multiple scales and channels are concatenated to form the feature matrix $F \in R^{C \times L}$, where C is the total number of feature channels, each containing semantic features of all words in a TSC. Each word has multiple channels for feature representation. To fuse the features of different words and obtain the representation vector of the entire TSC, the feature matrix *F* is input to the word-level attention mechanism, with the goal of determining which words need promotion or suppression. The word-level attention mechanism first extracts the global information of each TSC, and then conducts a “squeeze and excitation” operation to obtain the weight of each word. The weights directly work on the word representation for feature fusion, and the weights will be automatically updated in the training process of SHDL. A word with richer semantic information will adaptively get a larger weight; thus, the word-level attention mechanism highlights important emotional words that determine the semantics of the corresponding TSC. Specifically, since we pay attention to the importance of words, global average pooling is first performed along the channel axis to obtain global information of all words, which can be formalized as follows:

$$v_l = \mathrm{GAP}(F_l) = \frac{1}{C}\sum_{c=1}^{C} F_l^c \tag{3}$$

where *l* represents the index of words, F_l^c represents the feature point of the *l*th word of the *c*th channel of the feature matrix, and v_l represents the squeezed point of the *l*th word. Then, the squeezed points of all words are concatenated to obtain a word descriptor $V \in R^{L \times 1}$ that contains global information of a TSC. Based on the word descriptor *V*, two fully connected layers are used as the excitation operation to generate the word attention weights, which can be calculated as follows:

$$\alpha = \mathrm{Softmax}(W_{\alpha,2}\sigma(W_{\alpha,1}V)) \tag{4}$$

where $W_{\alpha,1} \in R^{\frac{L}{r} \times L}$ and $W_{\alpha,2} \in R^{L \times \frac{L}{r}}$ are weight matrixes, *r* is the dimension reduction ratio, and σ is the ReLU activation function. The Softmax function is used to normalize the attention weights, which can be denoted as $\alpha = [\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_L]$, which indicate the importance of words. The feature fusion of different words can be expressed as follows:

$$m = \sum_{l=1}^{L} \alpha_l F_l \tag{5}$$

where m is the representation vector for the corresponding TSC. Thus, the word-level attention mechanism determines where to highlight in the word-level feature vectors and adaptively fuses word representations according to their informativeness, and the comment-level vector representation of each TSC is obtained.

3.3. Context-level representation

Through the above comment-level representation, the obtained vector representation of each TSC in the augmented context can be denoted as $M = [m_{t-P}, \ldots, m_{t-1}, m_t, m_{t+1}, \ldots, m_{t+Q}]$. Then, the representation sequence is input into the context-level representation to capture the contextual dependency among multiple TSCs. Considering that TSCs appear in sequence according to their posting in the video timeline, capturing the temporal correlation of contextual TSCs is important. Moreover, given the viewers' unique ideas and diverse topics, viewers may express comments and opinions that differ from mainstream views, which indicates that contextual TSCs have different effects on judging the sentiment of the target TSC. Therefore, we propose a BiGRU with comment-level attention to extract the dependency and quantify the contributions of contextual TSCs for obtaining contextual semantic representations. Figure 4 illustrates the detailed structure of the context-level representation in SHDL.

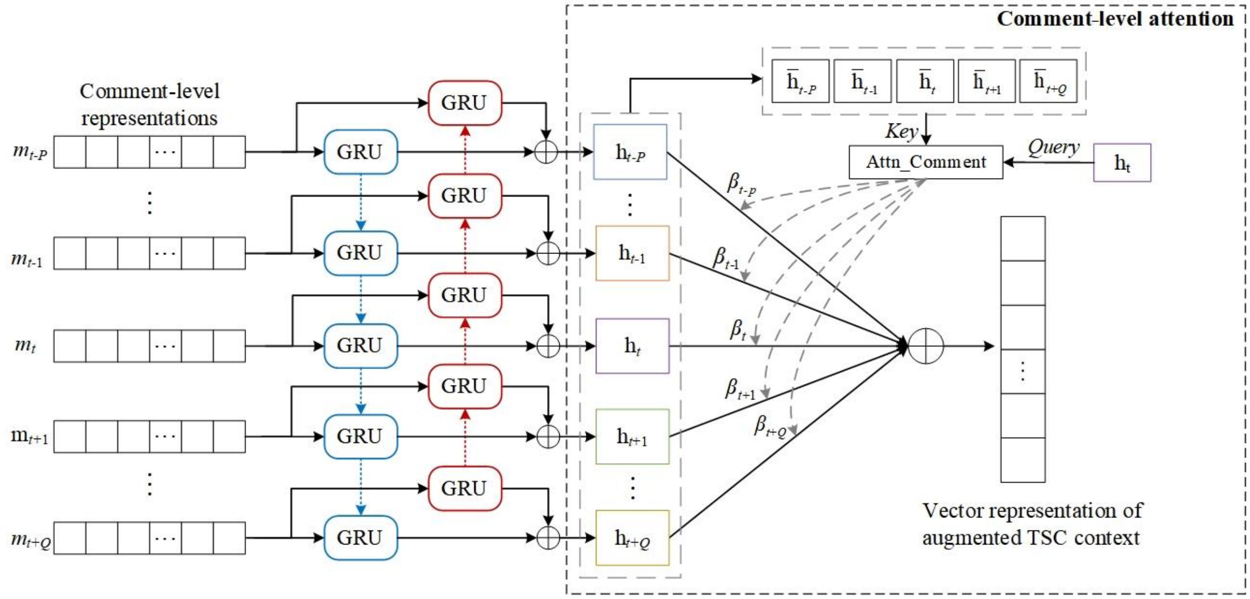


Figure 4. Detailed structure of context-level representation in SHDL

Considering the temporal correlations among TSCs, RNN and its variants LSTM and GRU, which can memorize historical information, are typically used for temporal data. However, standard RNN may face the problems of gradient vanishing or gradient explosion (Chung et al., 2014). Compared with LSTM, GRU, which introduces the gate mechanism, solves gradient problems with fewer parameters (Chung et al., 2014). In addition, BiGRU with its bidirectional structure can learn both the forward and backward dependencies of contextual TSCs. Hence, as shown in Figure 4, we employ BiGRU to capture the temporal correlation of contextual TSCs. Taking a sequence of obtained comment-level vector representations as input, BiGRU is used to obtain the hidden state features for each comment-level representation. The conversion functions of the GRU cell are as follows:

$$z_t = \sigma(U_z m_t + W_z h_{t-1} + b_z) \tag{6}$$

$$r_t = \sigma(U_r m_t + W_r h_{t-1} + b_r) \tag{7}$$

$$\tilde{h}_t = \tanh\,(U_h m_t + r_t \cdot W_h h_{t-1} + b_h) \tag{8}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{9}$$

where z_t and r_t are the update and reset gates, respectively, which are responsible for controlling the selective flow of information. U_z, W_z, U_r, W_r, U_h, and W_h are the weight parameters; b_z, b_r, and b_h

are the biases; σ is the Sigmoid function; $\tilde{h}_t$ represents the candidate state of the *t*th TSC; and h_t represents the final hidden state of a GRU cell. Based on the GRU cell, BiGRU carries out a bidirectional calculation and can capture the contextual dependencies among TSCs from both directions. For the *t*th TSC, the representation vectors are simultaneously input into the forward and backward GRUs, and the forward $\overrightarrow{h}_t$ and backward $\overleftarrow{h}_t$ semantic features are obtained, respectively, which are concatenated as the output. The concatenated feature vector of the *t*th TSC can be presented as follows:

$$h_t{=}\overrightarrow{h}_t \oplus \overleftarrow{h}_t = \mathrm{GRU}(h_{t-1}, m_t) \oplus \mathrm{GRU}(h_{t+1}, m_t) \tag{10}$$

Through the temporal feature representation by the above-described BiGRU, the obtained representation sequence of the TSC context is denoted as $H = [h_{t-P}, \dots, h_{t-1}, h_t, h_{t+1}, \dots, h_{t+Q}]$. Among the contextual TSCs used, the ones with higher semantic similarity to the target TSC contribute more to identifying the target sentiment, while the irrelevant ones are useless and may even become interference. To distinguish the contributions of different contextual TSCs, comment-level attention is applied on the contextual feature representations. The comment-level attention aims to analyze the semantic correlation between the target and contextual TSCs, and assigns weights according to their semantics. The key to the attention mechanism is defined as $\bar{h} = [\bar{h}_{t-P}, \dots, \bar{h}_{t-1}, \bar{h}_t, \bar{h}_{t+1}, \dots, \bar{h}_{t+Q}]$, which is transformed from the comment-level contextual features by a dense layer and calculated as follows:

$$\bar{h}_i = \tanh\left(W_{\beta,i} h_i + b_{\beta,i}\right), t - P \le i \le t + Q \tag{11}$$

where W_β and b_β are the weight and bias of the dense layer, respectively. Considering that the target TSC must be dominant to identify its sentiment, we distinguish the target TSC by defining the query of the developed comment-level attention as the state vector of BiGRU in the time step of the target TSC $\bar{h}_t$. Thus, the attention weight is calculated as follows:

$$\beta_i = \frac{\exp\,(\bar{h}_i h_t^T)}{\sum_{i=t-P}^{t+Q} \exp\,(\bar{h}_i h_t^T)} \tag{12}$$

where β_i is the attention weight of the *ith* TSC in the augmented sample. In this way, we can compute the similarity of the target and its contextual TSCs, and a more relevant TSC is distributed with a larger weight. Consequently, if the semantics of all contextual TSCs are mainly consistent with that of the target TSC, the target TSC then tends to be assigned a smaller weight. Conversely, if the semantics of most contextual TSCs are uncorrelated to the target TSC, then the weight of the latter becomes larger. Then, the attention weights are applied to the contextual representation sequence, which is expressed as follows:

$$d_t = \sum_{i=t-P}^{t+Q} \beta_i\, h_i \tag{13}$$

where d_t is the obtained representation vector. Through the comment-level attention mechanism, we reasonably take advantage of the contextual dependency of TSC data by strengthening the role of relevant contextual TSCs and reducing the role of irrelevant ones. In addition, the contextual semantics are adaptively fused. The ultimately fused context vector representation covers the contextual semantics of the whole augmented TSC sample.

Finally, the obtained context representation is used to output the sentiment probability by a fully connected layer and activation function.

3.4. Objective function

Among viewers who watch online videos and post TSCs, many are interested in the video content or already love the people or objects of the video, whereas those who are not interested in the content or are disgusted by the content seldom watch it. As such, TSCs are normally dominated by positive sentiments, whereas negative sentiments are relatively fewer. This causes an imbalance between the two sentiment classes in TSC data. This problem is alleviated by using a focal loss as the objective function to train our proposed TSC sentiment classification model (Lin et al., 2020). The focal loss has been developed in the fields of image analysis and object detection, which have proven its effectiveness in compensating for class imbalance. By increasing the weight of the minority class and reducing that of the majority class, the focal loss allows the model to focus more on samples that are difficult to classify.

The focal loss is constructed on the basis of the traditional binary cross entropy loss function and introduces the weighting factor α_t and the tunable focusing parameter γ to account for the class imbalance. Specifically, the loss is computed as follows:

$$p_t = \begin{cases} p, if\ y = 1 \\ 1 - p, otherwise \end{cases} \tag{14}$$

$$\text{Loss}_{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{15}$$

where P is the calculated probability for the class and y is the label of the classified sample. By utilizing the focal loss, the proposed model is penalized for overconfidence in predicting certain values and pays more attention to the training for difficult negative samples.

4. Empirical evaluation

4.1. Data

We evaluated our proposed sentiment classification method using a real-world TSC dataset collected from a series of video programs on Bilibili, which is one of the largest TSC-enhanced online video platforms. The dataset contains the text content of TSCs and their posting time in the video timeline. For TSC annotation, we randomly selected 15,000 candidate TSCs from the collected TSC set for labeling. Three domain experts were solicited to label the candidate TSCs, with each TSC being assessed by all three experts. Initially, the annotators performed the labeling independently according to the predetermined annotation guideline (available in Appendix B), in which TSCs with sentiment tendencies were labeled with positive or negative labels, and the neutral TSCs were filtered out. Following the initial labeling process, the inter-rater agreement, as measured by Fleiss's kappa value (Fleiss, 1971), reached 0.79, indicating a substantial level of agreement. In the event of initial labeling inconsistencies, the experts engaged in discussions and conducted relabeling to ensure accuracy and consistency. To determine the final sentiment labels, we employed the majority vote mechanism. Consequently, we obtained a labeled dataset consisting of 13,135 target TSCs for empirical evaluation. Among these, 11,090 TSCs were labeled as positive sentiments, while 2,045 TSCs were labeled as negative sentiments. For each labeled TSC, the four closest TSCs before it and four closest TSCs after it (in terms of posting time) were considered as the possible contextual TSCs of the labeled (or target) TSC for contextual information, resulting in 105,080 unlabeled TSCs that were used as contextual TSCs. Overall, our experimental data comprised a total of 118,215 TSCs.

4.2. Experimental design

The representative methods of sentiment classification (as summarized in Table 1) were selected as benchmarked methods. Benchmarked machine learning methods include NB, LR, SVM, and RF, which have been commonly used for text mining (Onan et al., 2016; Ghaddar & Naoum-Sawaya, 2018; Parmar et al., 2014). Benchmarked deep learning methods include TextCNN, BiRNN, RCNN, and BiGRU with Attention (BiGRU-A) (Kim, 2014; Kratzwald et al., 2018; Lai et al., 2015; Wang et al., 2021b). For machine learning methods, we calculated word frequency vectors on the basis of the TF-IDF algorithm for document feature extraction, which were used as the input. For deep learning methods, the embedding word vectors of TSCs as the input were acquired using the skip-gram technique of Word2Vec, with dimension set to 100 and sequence length set to 16. For training all the deep learning methods, the Adam optimizer was used to train the models with the learning rate of 0.005 and batch size of 32. All the hyperparameters mentioned above were tuned using a validation set in our experiments. Meanwhile, the early stopping criteria and dropout were applied to avoid overfitting.

To measure the performance of the sentiment classification of our experimental methods, we adopted three performance metrics, including recall, precision, and F1-score. Recall reflects the ability of the model to detect target categories, precision reflects the accurate proportion of all samples predicted to be of this category, and F1-score reflects the trade-off between recall and precision. We calculated the above three performance metrics for each category separately. Typically, a desired classification method is expected to have high values for each performance metric.

We evaluated the sentiment classification performance using repeated cross-validation. Specifically, we conducted 10 independent five-fold cross-validations with different random seeds, resulting in 50 performance estimates. Such a way can effectively alleviate the impact of randomness in the training-testing split, and thus has been used in many extant studies (Chen et al., 2023; Jiang et al., 2019; Rodriguez et al., 2009). During each cross-validation, the dataset was divided into five equal-sized subsets (folds), and each fold was used to estimate the performance of the classifier trained on the other four folds. For fairness, the fold splitting was kept identical across all classification methods. Performance results (averages and the 95 percent confidence interval) reported later are all based on the 50 estimates. Moreover, all experiments were implemented by Python 3.8 based on Xeon(R) Gold 5218R CPU and NVIDIA-Tesla-TU104GL GPU.

5. Results

5.1. Sentiment classification performance

Table 2 summarizes the performance comparisons of the above-mentioned methods on the sentiment classification of TSCs in terms of recall, precision, and F1-score. The results show that the proposed method outperforms the traditional machine learning and deep learning methods in terms of all performance metrics. Overall, compared with traditional machine learning methods, deep learning methods achieve better classification performances. The chosen deep learning methods can handle the sequence relationship of words in the text, which enables them to capture richer semantic information. Compared to the deep learning methods (TextCNN, BiRNN, RCNN, and BiGRU-A), SHDL shows a better sentiment classification performance in terms of each performance metric for both positive and negative classes, indicating its superiority and the robustness of the results.

Table 2. Sentiment classification performance

Model	Recall (%)		Precision (%)		F1-score (%)	
	Positive	Negative	Positive	Negative	Positive	Negative
NB	90.26(90.05–90.46)	74.85(74.31–75.39)	95.09(94.99–95.18)	58.80(58.30–59.30)	92.60(92.49–92.71)	65.84(65.45–66.22)
LR	97.11(97.00–97.22)	65.66(65.10–66.23)	93.85(93.75–93.94)	80.87(80.28–81.46)	95.45(95.38–95.52)	72.45(72.01–72.89)
SVM	96.54(96.42–96.65)	66.05(65.46–66.62)	93.88(93.78–93.97)	78.03(77.47–78.59)	95.19(95.12–95.26)	71.51(71.08–71.94)
RF	97.68(97.57–97.79)	62.43(61.91–62.95)	93.34(93.26–93.43)	83.37(82.73–84.01)	95.46(95.39–95.52)	71.37(70.95–71.79)
TextCNN	96.53(96.36–96.69)	79.79(78.72–80.86)	96.17(95.91–96.43)	81.65(80.85–82.44)	96.33(96.27–96.39)	80.36(80.17–80.56)
BiRNN	96.94(96.66–97.22)	74.64(73.56–75.72)	94.59(94.12–95.06)	83.73(82.62–84.84)	95.74(95.54–95.94)	78.63(77.76–79.49)
RCNN	96.38(95.95–96.82)	82.39(81.72–83.07)	96.66(96.36–96.95)	81.37(80.56–82.19)	96.54(96.37–96.72)	81.68(80.94–82.42)
BiGRU–A	96.39(95.84–96.94)	82.30(80.94–83.66)	96.71(96.32–97.09)	83.58(82.68–84.48)	96.55(96.02-97.09)	82.75(81.69–83.81)
SHDL	**97.81**(97.68–97.95)	**87.33**(86.79–87.87)	**97.67**(97.57–97.76)	**88.16**(87.55–88.76)	**97.74**(97.70–97.78)	**87.71**(87.50–87.90)

We tested the statistical significance of the comparisons between SHDL and benchmarked methods using both non-parametric and parametric tests. For the non-parametric test, we used the Friedman test with a post-hoc procedure (Demšar 2006). Table 3 summarizes the results of the pairwise comparisons adjusted by Bonferroni correction of the nine sentiment classification methods in terms of F1-score for the negative class. The actual p-values in terms of all performance metrics for both positive and negative classes are available in Appendix C. Overall, the differences across the nine sentiment classification methods were statistically significant (χ^2=377.58, p<0.001). Further pairwise comparisons verify that SHDL significantly outperformed all benchmarked methods.

Table 3. Results of the Friedman test in terms of F1-score for the negative class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	9.00								
LR	6.00	<0.001							
SVM	7.24	<0.05	1.00						
RF	7.76	<0.05	1.00	1.00					
TextCNN	3.72	<0.001	<0.001	<0.001	<0.001				
BiRNN	4.56	<0.001	<0.001	<0.001	<0.001	1.00			
RCNN	3.12	<0.001	<0.001	<0.001	<0.001	1.00	0.56		
BiGRU-A	2.56	<0.001	<0.001	<0.001	<0.001	1.00	0.07	1.00	
SHDL	1.04	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.05	<0.05
Friedman χ^2	377.58 (<0.001)								

For the parametric test, we used the repeated measures ANOVA, with the method (SHDL vs. one of the benchmarked methods, respectively) as the main factor. Figure 5 illustrates the effect size (i.e., partial η^2) of using SHDL in lieu of each benchmarked method in terms of performance improvement. To comprehensively reflect the performance improvement, for each performance metric, we calculated the averages of the two target classes as comparative data. The results show that except for the partial η^2 in terms of recall when SHDL and BiGRU-A are the main factors, all the others are over 0.4, proving that using the proposed method accounts for the conspicuous performance improvement in sentiment classification. The comparison results verify that the proposed method can better identify the sentiment polarity of TSCs and improve sentiment classification performance.

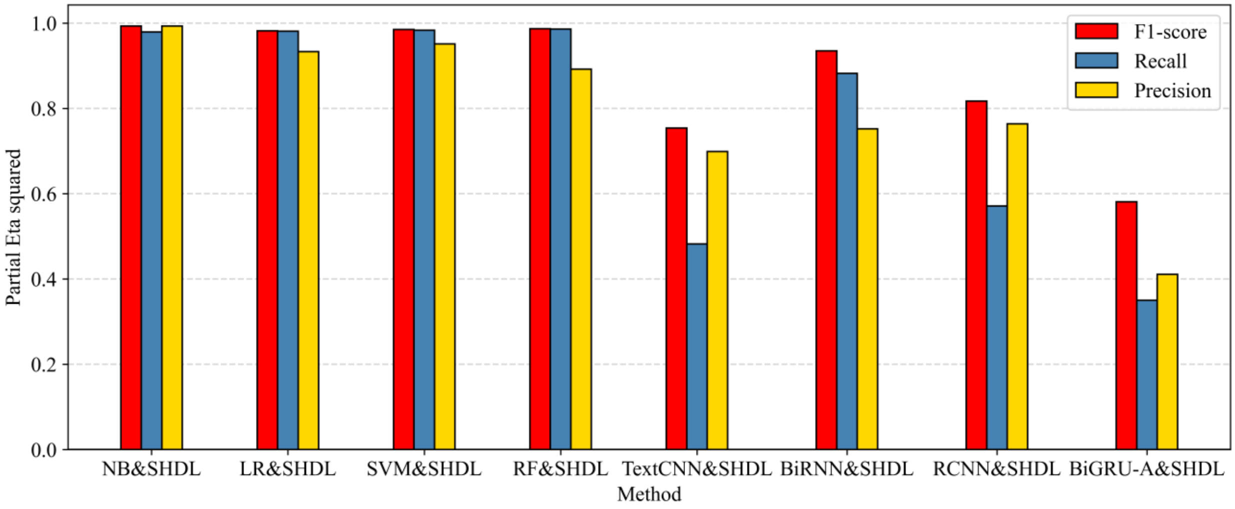


Figure 5. Partial η^2 of repeated measures ANOVA

5.2. Equal feature space

Considering that the proposed method is fed with information from several TSCs, to verify whether the superiority of our method stems from the additional data that is considered for each observation or from the method being able to better capture the information from the available data, we conducted extra feature space experiments. In this section, the input of the benchmark methods was also contextual TSCs, as in SHDL, to ensure a consistent feature space. Specifically, in view of the fact that the benchmarks do not have hierarchical feature learning capabilities, we combined the target TSC with its contextual TSCs into a TSC document as the input of the benchmarks, and the sequence length of the combined TSC document is equivalent to the sequence length of a single TSC multiplied by the number of TSCs included. The remaining settings were the same as in the above experiments. As summarized in Table 4, in the feature space experiments, RF achieved the best performance in terms of recall for positive sentiment and precision for negative sentiment, while SHDL achieved the best performance in terms of F1-score. Although the benchmark methods were fed additional data, the performance of most benchmarks did not improve and even decreased, which indicates that their structures cannot extract valuable contextual information and that the added TSC data may be a form of interference. The experimental results verify that SHDL outperforms most other methods due to its excellent information extraction capability.

Table 4. Results of feature space experiments

Model	Recall (%)		Precision (%)		F1-score (%)	
	Positive	Negative	Positive	Negative	Positive	Negative
NB	88.09(87.92–88.26)	66.56(65.87–67.24)	93.43(93.30–93.55)	50.91(50.47–51.34)	90.68(90.57–90.79)	57.68(57.21–58.15)
LR	95.10(94.97–95.22)	65.57(65.02–66.11)	93.71(93.62–93.80)	71.31(70.81–71.81)	94.40(94.32–94.47)	68.29(67.90–68.68)
SVM	93.39(93.23–93.55)	66.30(65.74–66.86)	93.73(93.63–93.83)	65.07(64.53–65.61)	93.56(93.47–93.65)	65.65(65.23–66.06)
RF	**98.71**(98.62–98.79)	59.58(59.04–60.12)	92.94(92.85–93.03)	**89.62**(89.01–90.23)	95.74(95.68–95.79)	71.54(71.12–71.96)
TextCNN	97.25(97.05–97.45)	69.46(68.64–70.26)	94.53(94.39–94.66)	82.54(81.53–83.55)	95.87(95.77–95.97)	75.35(74.81–75.88)
BiRNN	97.10(96.85–97.35)	69.08(67.28–70.88)	94.07(93.77–94.38)	82.60(81.49–83.72)	95.62(95.46–95.78)	73.45(72.35–74.55)
RCNN	97.17(96.94–97.41)	68.46(67.66–69.27)	94.35(94.22–94.48)	82.00(80.85–83.15)	95.74(95.63–95.85)	74.52(73.96–75.08)
BiGRU–A	97.11(96.73–97.48)	77.56(76.39–78.72)	95.93(95.62–96.23)	83.72(82.11–85.34)	96.50(96.30–96.70)	80.23(79.14–81.31)
SHDL	97.81(97.68–97.95)	**87.33**(86.79–87.87)	**97.67**(97.57–97.76)	88.16(87.55–88.76)	**97.74**(97.70–97.78)	**87.71**(87.50–87.90)

5.3. Ablation study

To verify the contribution of each artifact in SHDL, we also carried out an ablation study. For comparison with the complete model, we respectively removed one of the components in SHDL to construct five reduced models (i.e., M1–M5). Specifically, we respectively removed CNN (M1), word-level attention (M2), BiGRU (M3), and comment-level attention (M4), and used the standard binary cross entropy loss function as the objective function for model training (M5). Table 5 shows the results of the ablation experiments, which indicate that removing any component leads to the overall classification performance degradation of SHDL and demonstrates the effectiveness of these key artifacts in SHDL for the semantic representation and sentiment classification of TSCs. Moreover, Figure 6 shows the performance decrease percentage of the above reduced methods (M1–M5) with the SHDL as a benchmark in terms of F1-score. The figure shows that the largest performance decrease is in M3 when BiGRU is removed, with the F1-score for positive and negative classes decreasing by 1.44% and 6.64%, respectively. These results indicate that BiGRU, which captures the temporal correlation of a TSC, is significant for contextual semantic representation.

Table 5. Results of ablation experiments

Model	Recall (%)		Precision (%)		F1-score (%)	
	Positive	Negative	Positive	Negative	Positive	Negative
M1	97.46(97.31–97.61)	81.52(80.82–82.22)	96.60(96.47–96.72)	85.63(84.94–86.31)	97.02(96.96–97.08)	83.43(83.08–83.78)
M2	97.34(97.15–97.53)	85.87(84.83–86.91)	97.16(96.91–97.41)	84.75(84.03–85.47)	97.25(97.16–97.34)	85.21(84.63–85.79)
M3	96.11(95.86–96.36)	80.73(79.61–81.85)	96.57(96.36–96.78)	83.89(83.05–84.73)	96.33(96.24–96.42)	81.89(81.35–82.43)
M4	97.32(97.12–97.52)	84.80(83.85–85.75)	97.41(97.19–97.63)	85.67(84.94–86.40)	97.36(97.29–97.43)	85.18(84.62–85.74)
M5	**97.87**(97.68–98.06)	83.59(82.49–84.69)	97.56(97.37–97.75)	85.27(84.48–86.06)	97.71(97.64–97.78)	84.29(83.86–84.72)
SHDL	97.81(97.68–97.95)	**87.33**(86.79–87.87)	**97.67**(97.57–97.76)	**88.16**(87.55–88.76)	**97.74**(97.70–97.78)	**87.71**(87.50–87.90)

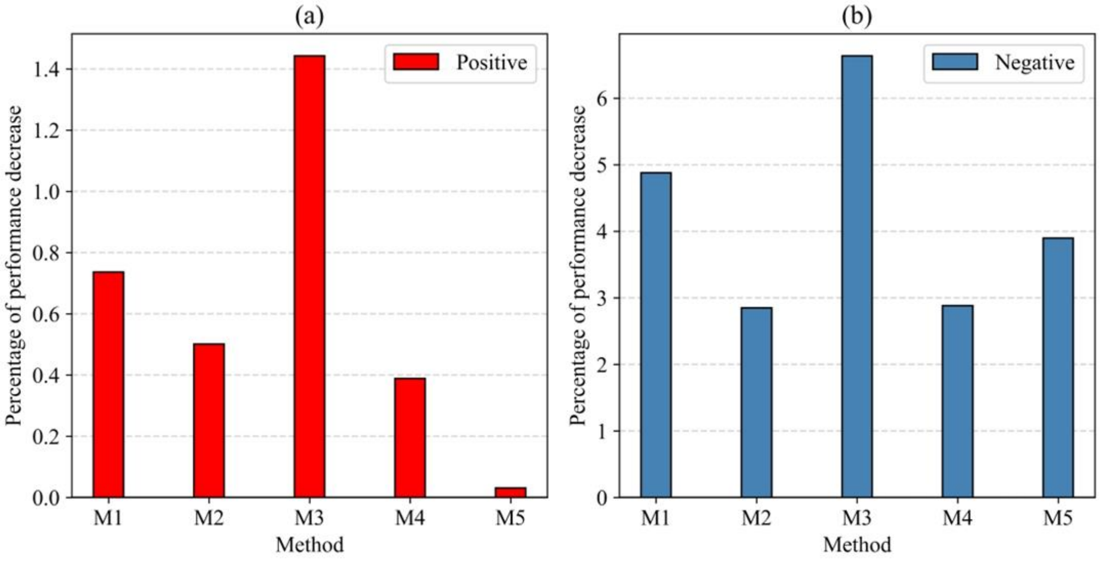


Figure 6. Percentage of performance decrease of the comparative methods in terms of F1-score

5.4. Effect of context size

In SHDL, we use a contextual window to locate contextual TSCs and augment the target TSC for its sentiment classification. The context size, which reflects the number of contextual TSCs used (i.e., the window size), acts as a key hyperparameter of the proposed method, and its effect on the classification performance was analyzed. Considering that the former and latter posted TSCs of target TSC included in contextual TSCs may have different effects, we evaluated the performance of the proposed method with varying numbers of former and latter TSCs. Figure 7 reports the overall performance of SHDL with 1, 2, 3, and 4 former and latter TSCs in terms of average recall, precision, and F1-score. In Figure 7, the horizontal and vertical axes represent the number of former and latter TSCs used. The results show that SHDL achieves optimal performance when the number of former and latter TSCs are both 3. Theoretically, as the number of contextual TSCs used increases, SHDL will be able to capture longer distance dependencies among TSCs. In addition, with an increase in the number of contextual TSCs used, more unlabeled TSC data can be used to provide more abundant semantic information. However, in general, a contextual TSC that is farther from the target TSC in terms of posting time means a weaker dependency, and two TSCs with too long a distance between them may have no correlation. Hence, increasing the number of contextual TSCs used does not necessarily improve the sentiment classification performance of the model. Figure 7 shows that enforcing a proper number of former and latter TSCs (3 in our study) improves the model performance. Given this number, the context size of the size of the window is 7.

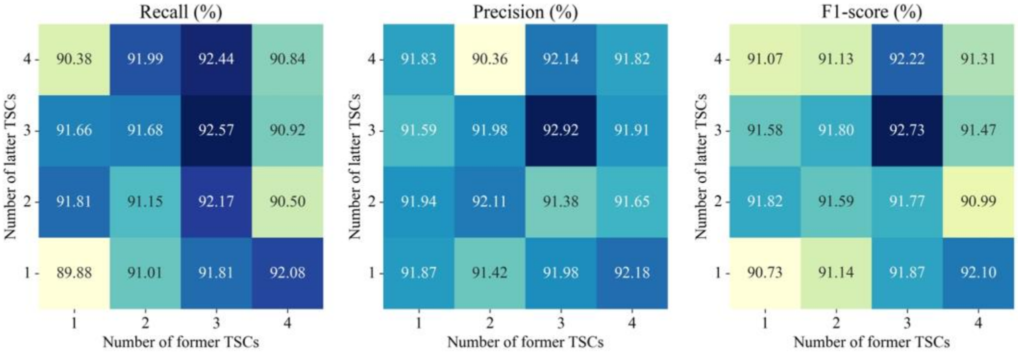


Figure 7. Sentiment classification performance with different context sizes

5.5. Representation performance analysis

The proposed method hierarchically generates semantic representations of TSC text at word, comment, and context levels. The word- and comment-level representations are both generated from one single TSC, while the context-level representations are generated from the augmented TSC consisting of a target TSC and its contextual TSCs. To intuitively show the superiority of context-level representations on sentiment classification, we used the t-SNE technique to graphically illustrate the learned comment- and context-level representations, with dimensions reduced into two for visualization. For this comparison, Figure 8 shows the reduced 2D features of comment- and context-level representations, where the different colors represent different sentiment categories. Figure 8(a) provides the 2D visual features of comment-level representations extracted from target TSCs. We can observe that different sentiment categories heavily overlap, which indicates that the feature information of a single TSC is hardly differentiable. Figure 8(b) provides the 2D visual features of context-level representations extracted from target TSCs and their contextual TSCs. The features of different sentiment categories are much more separable while those of the same sentiment category display better cluster performance, and thus this method enables easier classification of different sentiments. The visualization results prove that using contextual TSCs to generate context-level semantic representations highly improves sentiment classification.

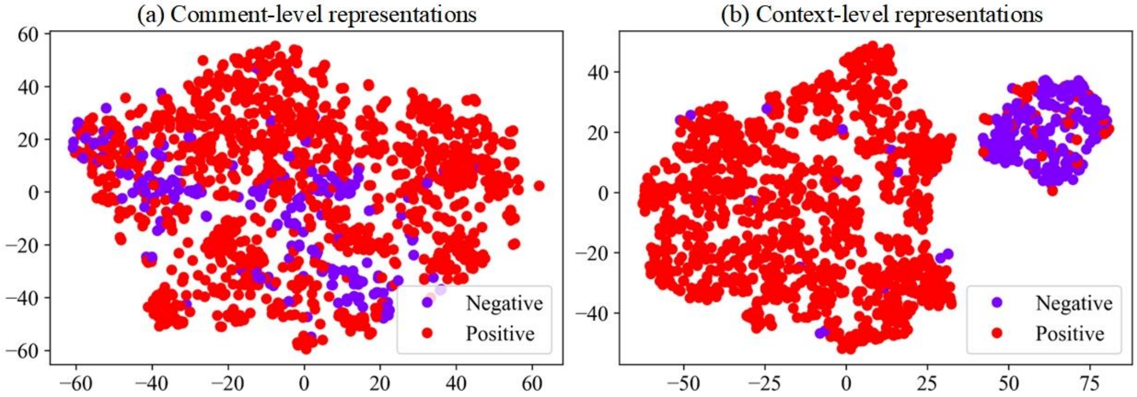


Figure 8. Feature visualization of the hierarchical representations by t-SNE

6. Conclusions

As a new type of online comment, TSCs contain significant sentiment information with considerable potential value for the success of online video platforms. The contextual dependency of TSCs provides opportunities for using contextual TSCs to assist in identifying the sentiment of a target TSC. In this study, we identified the challenges in capturing contextual information from neighboring TSCs and fusing contextual semantics for sentiment classification. To address these challenges, we proposed a semi-supervised hierarchical deep learning method for sentiment classification of TSCs with the reasonable usage of contextual TSCs. Specifically, we designed a hierarchical architecture to capture the multilevel semantics of TSCs and developed two attention mechanisms for semantic fusion at different levels. We evaluated our method using a TSC dataset from a popular online video platform. The empirical results show that our SHDL method effectively improved the performance of TSC sentiment classification compared with benchmark methods. The results advance our knowledge of contextual information indicative of TSC sentiment.

This study contributes to both research and practice. First, we propose a novel and effective sentiment classification method, and managers and operators of online video platforms could use the proposed method to analyze the sentiments of the massive numbers of TSCs on their platforms, so as to grasp user demands and enhance their service performance. Second, we use contextual TSC information in a semi-supervised manner, which could save burdensome data annotation work and reduce the model deployment costs of online video platforms. Third, while the proposed method focuses on TSCs, the prescriptive knowledge advanced in this study (e.g., hierarchical semantic representation architecture and attention mechanisms) may be generalizable to other types of interactive text, such as Q&A text.

This study has several limitations, which may be addressed in future research. First, our proposed method was evaluated on only one dataset with a class imbalance issue. Further research may collect data from various online video platforms to comprehensively evaluate our proposed method. Second, the proposed method used a fixed number of contextual TSCs, whereas the distance of the contextual dependency depends on the density of posted TSCs and is nonstationary. Further research may consider designing a method that could adaptively select context windows to further improve the sentiment classification performance. Third, due to the reliance on TSCs following the target TSC, our proposed method may not be suitable for live scoring. Future research may consider designing artifacts for accommodating live scoring.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [Grant 72101073], and the Anhui Provincial Natural Science Foundation [Grant 2108085MG234].

Appendix A. Performance using CBOW of Word2Vec as word embedding model

To observe the sentiment classification performance under different word embedding models, we also used the CBOW method of Word2Vec (Mikolov et al., 2013) as the word embedding model to build deep learning methods (TextCNN, BiRNN, RCNN, BiGRU-A, and SHDL) in our experiments. The results are shown in Table A.1. It indicates that when using CBOW as the word embedding model, the performance of the proposed method is inferior to that of using skip-gram, although SHDL still outperforms the other, compared methods.

Table A1. Sentiment classification performance using CBOW

Model	Recall (%)		Precision (%)		F1-score (%)	
	Positive	Negative	Positive	Negative	Positive	Negative
TextCNN	95.90(95.59–96.21)	81.11(79.93–82.29)	96.50(96.31–96.70)	78.92 (77.69–80.16)	96.20(96.12–96.27)	79.78 (79.51–80.04)
BiRNN	95.88(95.61–96.14)	79.47(78.21–80.74)	96.21(95.99–96.43)	78.27(77.31–79.23)	96.04(95.92–96.15)	78.72(78.09–79.35)
RCNN	95.96 (95.68–96.25)	82.64 (81.54–83.74)	96.78(96.59–96.97)	79.35(78.40–80.31)	96.37(96.29–96.44)	80.81 (80.51–81.12)
BiGRU–A	96.94 (96.53–97.34)	81.78 (80.32–83.24)	96.68(96.35–97.01)	83.93 (82.76–85.09)	96.79(96.70–96.88)	82.38(81.88–82.88)
SHDL	**97.36**(97.22–97.51)	**86.14**(85.50–86.79)	**97.45**(97.33–97.56)	**85.86**(85.26–86.46)	**97.40** (97.36–97.44)	**85.95**(85.75–86.15)

Appendix B. Some details for data annotation

The annotation guidelines for data annotation are shown in Table B.1. TSCs with obvious sentiment tendencies are labeled with positive or negative sentiment labels by the annotators, while TSCs expressing statements are regarded as neutral sentiments and are skipped during the annotation process.

Table B1. The annotation guideline for data annotation

Sentiment category	Feelings expressed
Positive	Expressing love, praise, satisfaction, or comfort toward the objects in the video or toward the video creator. Describing the joy, excitement, or emotional movement of the viewer.
Negative	Expressing criticism, disgust, anger, fear, or disappointment toward the objects in the video or toward the video creator. Describing the sadness, anxiety, or pain of the viewer.

For the annotation task, the annotators were three graduate students majoring in Management Science and Engineering at Hefei University of Technology. They have long conducted research in the field of online video and social media commentary, and are familiar with the operation of new online video platforms, such as the TSC mechanism.

Appendix C. The actual p-values of non-parametric full pairwise comparisons

Tables C1 to C6 report the actual *p*-values of full pairwise comparisons in terms of F1-score, recall, and precision, for positive and negative classes, respectively.

Table C1. Results of full pairwise comparison in terms of F1-score for the negative class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	9.00								
LR	6.00	2.27e-04							
SVM	7.24	1.23e-02	1.00						
RF	7.76	2.06e-02	1.00	1.00					
TextCNN	3.72	2.95e-23	2.86e-07	7.32e-10	2.84e-10				
BiRNN	4.56	2.49e-17	4.68e-04	4.29e-06	2.01e-06	1.00			
RCNN	3.12	4.91e-28	4.31e-10	4.36e-13	1.48e-13	1.00	5.58e-01		
BiGRU-A	2.56	2.13e-31	3.57e-12	1.97e-15	6.10e-16	1.00	7.41e-02	1.00	
SHDL	1.04	8.62e-51	3.31e-25	8.89e-30	1.77e-30	2.81e-05	7.79e-09	3.07e-03	3.90e-02
Friedman χ^2	377.58 (1.17e-76)								

Table C2. Results of full pairwise comparison in terms of F1-score for the positive class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	8.96								
LR	6.10	5.85e-06							
SVM	7.78	2.70e-02	1.00						
RF	6.00	2.64e-06	1.00	1.00					

TextCNN	3.78	6.45e-22	7.46e-05	1.36e-09	1.51e-04				
BiRNN	5.34	4.96e-10	1.00	2.52e-02	1.00	4.55e-02			
RCNN	3.26	2.63e-26	4.22e-07	1.34e-12	9.81e-07	1.00	1.04e-03		
BiGRU-A	2.78	2.29e-29	9.12e-09	9.28e-15	2.31e-08	1.00	5.64e-05	1.00	
SHDL	1	4.15e-51	3.83e-22	4.15e-31	1.65e-21	4.42e-06	6.10e-16	5.32e-04	7.45e-03
Friedman χ^2	341.20 (6.84e-69)								

Table C3. Results of full pairwise comparison in terms of recall for the negative class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	5.34								
LR	7.54	2.41e-04							
SVM	7.43	7.20e-04	1.00						
RF	8.97	1.03e-09	1.00	6.10e-01					
TextCNN	3.70	4.73e-02	4.32e-13	2.72e-12	2.11e-21				
BiRNN	5.38	1.00	1.50e-04	4.58e-04	5.22e-10	6.67e-02			
RCNN	2.60	1.89e-05	6.23e-20	6.01e-19	6.56e-30	1.00	3.17e-05		
BiGRU-A	2.76	5.44e-05	4.47e-19	4.10e-18	7.30e-29	1.00	8.92e-05	1.00	
SHDL	1.28	5.41e-13	1.26e-32	2.28e-31	4.42e-45	2.76e-04	1.18e-12	2.73e-01	1.44e-01
Friedman χ^2	364.80 (6.27e-74)								

Table C4. Results of full pairwise comparison in terms of recall for the positive class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	8.98								
LR	4.31	2.42e-18							
SVM	6.09	5.99e-07	1.75e-02						
RF	2.22	7.47e-35	1.88e-02	1.25e-10					
TextCNN	5.94	1.09e-07	4.91e-02	1.00	9.14e-10				
BiRNN	4.48	8.56e-17	1.00	7.11e-02	4.02e-03	1.80e-01			
RCNN	5.61	1.01e-09	4.78e-01	1.00	9.94e-08	1.00	1.00		
BiGRU-A	5.34	5.22e-11	1.00	1.00	1.20e-06	1.00	1.00	1.00	
SHDL	2.03	1.85e-37	3.01e-03	4.14e-12	1.00	3.46e-11	5.41e-04	5.24e-09	7.69e-08
Friedman χ^2	238.13 (5.64e-47)								

Table C5. Results of full pairwise comparison in terms of precision for the negative class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	9.00								
LR	5.62	3.56e-09							
SVM	7.54	4.42e-02	4.36e-02						

RF	3.78	1.23e-21	2.01e-02	8.19e-10					
TextCNN	4.94	2.17e-13	1.00	1.75e-04	1.00				
BiRNN	4.00	2.62e-21	2.66e-02	1.37e-09	1.00	1.00			
RCNN	5.14	1.05e-11	1.00	1.73e-03	3.15e-01	1.00	3.92e-01		
BiGRU-A	3.68	4.41e-22	1.37e-02	4.06e-10	1.00	9.55e-01	1.00	2.32e-01	
SHDL	1.30	7.89e-47	1.33e-14	1.80e-28	9.50e-05	3.43e-10	6.54e-05	9.00e-12	1.56e-04
Friedman χ^2	271.87 (3.95e-54)								

Table C6. Results of full pairwise comparison in terms of precision for the positive class

		p-value of Pairwise Comparison Adjusted by Bonferroni Correction							
	Average Rank	NB	LR	SVM	RF	TextCNN	BiRNN	RCNN	BiGRU-A
NB	5.1								
LR	7.1	1.05e-03							
SVM	7.1	2.01e-03	1.00						
RF	8.74	2.54e-09	6.94e-01	4.61e-01					
TextCNN	3.6	1.04e-01	2.91e-11	8.57e-11	7.59e-20				
BiRNN	6.16	1.00	5.68e-01	8.47e-01	7.18e-05	7.49e-05			
RCNN	2.96	1.33e-03	3.57e-15	1.24e-14	6.57e-25	1.00	1.38e-07		
BiGRU-A	2.82	1.02e-03	2.15e-15	7.56e-15	3.45e-25	1.00	9.56e-08	1.00	
SHDL	1.42	4.12e-10	1.99e-26	1.03e-25	7.62e-39	5.06e-03	4.33e-16	2.81e-01	3.35e-01
Friedman χ^2	319.02 (3.66e-64)								

References

Agarwal, A., Gupta, A., Kumar, A., & Tamilselvam, S. G. (2019). Learning risk culture of banks using news analytics. European Journal of Operational Research, 277(2), 770-783.

Ankita, Rani, S., Bashir, A. K., Alhudhaif, A., Koundal, D., & Gunduz, E. S. (2022). An efficient CNN-LSTM model for sentiment detection in #BlackLivesMatter. Expert Systems with Applications, 193, 116256.

Bai, Q., Wei, K., Zhou, J., Xiong, C., Wu, Y., Lin, X., & He, L. (2021). Entity-level sentiment prediction in Danmaku video interaction. The Journal of Supercomputing, 77(9), 9474-9493.

Chakraborty, S., Basu, S., Ray, S., & Sharma, M. (2021). Advertisement revenue management: Determining the optimal mix of skippable and non-skippable ads for online video sharing platforms. European Journal of Operational Research, 292(1), 213-229.

Chan, K. H., & Im, S. K. (2022). Sentiment analysis by using Naive-Bayes classifier with stacked CARU. Electronics Letters, 58(10), 411-413.

Chen, G., Huang, L., Xiao, S., Zhang, C., & Zhao, H. (2023). Attending to Customer Attention: A Novel Deep Learning Method for Leveraging Multimodal Online Reviews to Enhance Sales Prediction.

Information Systems Research.

Chen, S., Li, S., Li, Y., Zhu, J., Long, J., Chen, S., & Yuan, X. (2022). DanmuVis: visualizing danmu content dynamics and associated viewer behaviors in online videos. Computer Graphics Forum, 41(3), 429-440.

Chen, Z., Tang, Y., Zhang, Z., Zhang, C., & Wang, L. (2019). Sentiment-aware short text classification based on convolutional neural network and attention. Paper presented at the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chong, A. Y. L., Li, B., Ngai, E. W., Ch'ng, E., & Lee, F. (2016). Predicting online product sales via online reviews, sentiments, and promotion strategies: A big data architecture and neural network approach. International Journal of Operations & Production Management, 36(4), 358-383.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. Paper presented at the NIPS 2014 Workshop on Deep Learning, December 2014.

Cruz, F. L., Troyano, J. A., Pontes, B., & Ortega, F. J. (2014). Building layered, multilingual sentiment lexicons at synset and lemma levels. Expert Systems with Applications, 41(13), 5984-5994.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7, 1-30.

Deng, S., Sinha, A. P., & Zhao, H. (2017). Adapting sentiment lexicons to domain-specific social media texts. Decision Support Systems, 94, 65-76.

Ghaddar, B., & Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. European Journal of Operational Research, 265(3), 993-1004.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5), 378.

Han, Y., Liu, Y., & Jin, Z. (2020). Sentiment analysis via semi-supervised learning: a model based on dynamic threshold and multi-classifiers. Neural Computing and Applications, 32(9), 5117-5129.

He, M., Ge, Y., Chen, E. H., Liu, Q., & Wang, X. S. (2018). Exploring the emerging type of comment for online videos: DanMu. ACM Transactions on the Web, 12(1), 1-33.

Jiang, C., Wang, Z., & Zhao, H. (2019). A prediction-driven mixture cure model and its application in credit scoring. European Journal of Operational Research, 277(1), 20-31.

Jiang, G., Shang, J., Liu, W., Feng, X., & Lei, J. (2020). Modeling the dynamics of online review life cycle: Role of social and economic moderations. European Journal of Operational Research, 285(1), 360-

379.

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. Paper presented at the Proceedings of NACCL-HLT.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Paper presented at the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

Kratzwald, B., Ilić, S., Kraus, M., Feuerriegel, S., & Prendinger, H. (2018). Deep learning for affective computing: Text-based emotion recognition in decision support. Decision Support Systems, 115, 24-35.

Kriebel, J., & Stitz, L. (2022). Credit default prediction from user-generated text in peer-to-peer lending using deep learning. European Journal of Operational Research, 302(1), 309-323.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. Paper presented at the Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.

Li, R., Lu, Y., Ma, J., & Wang, W. (2021a). Examining gifting behavior on live streaming platforms: An identity-based motivation model. Information & Management, 58(6), 103406.

Li, Y., & Guo, Y. (2021b). Virtual gifting and danmaku: What motivates people to interact in game live streaming? Telematics and Informatics, 62, 101624.

Liao, Z., Xian, Y., Li, J., Zhang, C., & Zhao, S. (2020). Time-sync comments denoising via graph convolutional and contextual encoding. Pattern Recognition Letters, 135, 256-263.

Liaw, C. M., & Dai, B. R. (2020). Live stream highlight detection using chat messages. Paper presented at the 2020 21st IEEE International Conference on Mobile Data Management (MDM).

Lin, T. Y., Goyal, P., Girshick, R., He, K. M., & Dollar, P. (2020). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 318-327.

Lv, G., Zhang, K., Wu, L., Chen, E., Xu, T., Liu, Q., & He, W. (2019). Understanding the Users and Videos by Mining a Novel Danmu Dataset. IEEE Transactions on Big Data, 535-551.

Meire, M., Ballings, M., & Van den Poel, D. (2016). The added value of auxiliary data in sentiment analysis of Facebook posts. Decision Support Systems, 89, 98-112.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, pp. 1-12.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Systems with Applications, 62, 1-16.

Parmar, H., Bhanderi, S., & Shah, G. (2014). Sentiment mining of movie reviews using Random Forest with Tuned Hyperparameters. Paper presented at the International Conference on Information Science.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE transactions on pattern analysis and machine intelligence, 32(3), 569-575.

Tarí, J. J., Molina, J. F., & Castejón, J. L. (2007). The relationship between quality management practices and their effects on quality outcomes. European Journal of Operational Research, 183(2), 483-501.

Tsai, M. F., & Wang, C. J. (2017). On the risk prediction and analysis of soft information in finance reports. European Journal of Operational Research, 257(1), 243-250.

Wang, B., Shan, D., Fan, A., Liu, L., & Gao, J. (2021a). A sentiment classification method of web social media based on multidimensional and multilevel modeling. IEEE Transactions on Industrial Informatics, 18(2), 1240-1249.

Wang, P., Li, J., & Hou, J. (2021b). S2SAN: A sentence-to-sentence attention network for sentiment analysis of online reviews. Decision Support Systems, 149, 113603.

Wu, C. H., & Chiu, Y. Y. (2023). Pricing and content development for online media platforms regarding consumer homing choices. European Journal of Operational Research, 305(1), 312-328.

Xi, D., Xu, W., Chen, R., Zhou, Y., & Yang, Z. (2021). Sending or not? A multimodal framework for Danmaku comment prediction. Information Processing & Management, 58(6), 102687.

Xia L., A., Li, Y., & Xu, S. X. (2021). Assessing the unacquainted: Inferred reviewer personality and review helpfulness. MIS Quarterly, 45(3), 1113-1148.

Xu, G., Zhang, Z., Zhang, T., Yu, S., Meng, Y., & Chen, S. (2022). Aspect-level sentiment classification based on attention-BiLSTM model and transfer learning. Knowledge-Based Systems, 245, 108568.

Xu, L. L., & Zhang, C. (2017). Bridging video content and comments: synchronized video description with temporal summarization of crowdsourced Time-sync comments. Paper presented at the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA.

Yang, K., Lau, R. Y. K., & Abbasi, A. (2022). Getting personal: A deep learning artifact for text-based measurement of personality. Information Systems Research, 24.

Yang, W., Jia, W., Gao, W., Zhou, X., & Luo, Y. (2019a). Interactive variance attention based online spoiler detection for time-sync comments. Paper presented at the Proceedings of the 28th ACM International Conference on Information and Knowledge Management.

Ye, Q., Zhang, Z., & Law, R. (2009). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Systems with Applications, 36(3), 6527-6535.

Zhou, J., Zhou, J., Ding, Y., & Wang, H. (2019). The magic of danmaku: A social interaction perspective of gift sending on live streaming platforms. Electronic Commerce Research and Applications, 34, 100815.