Detecting and Summarizing Video Highlights with Lag-Calibration

Abstract

The increasing popularity of video sharing has led to a growing need for automatic video analysis, including highlight detection. Emerging platforms that feature crowdsourced, time-synchronized video comments offer a valuable resource for identifying video highlights. However, this task presents several challenges: (1) time-synchronized comments often lag behind their corresponding shots; (2) these comments are frequently sparse and contain noise semantically; and (3) determining which shots constitute highlights is inherently subjective. This paper introduces a novel framework designed to address these challenges. The proposed method uses concept-mapped lexical chains to calibrate the lag in comments, models video highlights based on comment intensity and the combined concentration of emotion and concept within each shot, and summarizes detected highlights using an enhanced SumBasic algorithm that incorporates emotion and concept mapping. Experiments conducted on extensive real-world datasets demonstrate that our highlight detection and summarization methods substantially outperform existing benchmark techniques.

1 Introduction

Billions of hours of video content are viewed daily on platforms like YouTube, with mobile devices accounting for half of these views. This surge in video sharing has intensified the demand for efficient video analysis. Consider a scenario where a user wishes to quickly grasp the essence of a lengthy video without manually navigating through it. Automatically generated highlights would enable users to digest the video's key moments in a matter of minutes, aiding their decision on whether to watch the full video later. Furthermore, automated video highlight detection and summarization can significantly enhance video indexing, search, and recommendation systems.

However, extracting highlights from a video is a complex task. Firstly, the perception of a "highlight" can vary significantly among individuals. Secondly, analyzing low-level features such as image, audio, and motion may not always capture the essence of a highlight. The absence of high-level semantic information poses a significant limitation to highlight detection in conventional video processing.

The recent emergence of crowdsourced, time-synchronized video comments, also known as "bullet-screen comments," presents a new avenue for highlight detection. These real-time comments, which appear overlaid on the video screen, are synchronized with the video frames. This phenomenon has gained widespread popularity on platforms like niconico in Japan, Bilibili and Acfun in China, and YouTube Live and Twitch Live in the USA. The prevalence of time-synchronized comments offers a unique opportunity for leveraging natural language processing in video highlight detection.

Nevertheless, using time-synchronized comments for highlight detection and labeling still poses significant challenges. Primarily, there is an almost unavoidable delay between comments and their corresponding shots. As illustrated in Figure 1, discussions about a particular shot may continue into subsequent shots. Highlight detection and labeling without accounting for this lag may yield inaccurate outcomes. Secondly, time-synchronized comments are often semantically sparse, both in terms of the number of comments per shot and the number of words per comment. This sparsity

can hinder the performance of traditional bag-of-words statistical models. Thirdly, determining highlights in an unsupervised manner, without prior knowledge, involves considerable uncertainty. The defining characteristics of highlights must be clearly defined, captured, and modeled to ensure accurate detection.

To our knowledge, limited research has focused on unsupervised highlight detection and labeling using time-synchronized comments. The most relevant work in this area proposes detecting highlights based on the topic concentration derived from semantic vectors of bullet-comments, and labeling each highlight using a pre-trained classifier based on predefined tags. However, we contend that emotion concentration holds greater significance than general topic concentration in highlight detection. Another study suggests extracting highlights based on the frame-by-frame similarity of emotion distributions. However, neither of these approaches addresses the combined challenges of lag calibration, balancing emotion-topic concentration, and unsupervised highlight labeling.

To overcome these challenges, this study proposes the following solutions: (1) employ word-to-concept and word-to-emotion mapping based on global word embedding, enabling the construction of lexical chains for calibrating the lag in bullet-comments; (2) detect highlights based on the emotional and conceptual concentration and intensity of the lag-calibrated bullet-comments; and (3) summarize highlights using a modified Basic Sum algorithm that considers emotions and concepts as fundamental units within a bullet-comment.

The main contributions of this research are as follows: (1) We introduce a completely unsupervised framework for detecting and summarizing video highlights using time-synchronized comments; (2) We introduce a lag-calibration method that uses concept-mapped lexical chains; (3) We have created extensive datasets for bullet-comment word embedding, an emotion lexicon tailored for bullet-comments, and ground-truth data for evaluating highlight detection and labeling based on bullet-comments.

2 Related Work

2.1 Highlight detection by video processing

Following the definition from previous research, we define highlights as the most memorable shots in a video characterized by high emotional intensity. It's important to note that highlight detection differs from video summarization. While video summarization aims to provide a condensed representation of a video's storyline, highlight detection focuses on extracting its emotionally impactful content.

In the realm of highlight detection, some researchers have proposed representing video emotions as a curve on the arousal-valence plane, utilizing low-level features such as motion, vocal effects, shot length, and audio pitch, or color, along with mid-level features like laughter and subtitles. However, due to the semantic gap between low-level features and high-level semantics, the accuracy of highlight detection based solely on video processing is limited.

2.2 Temporal text summarization

Research on temporal text summarization shares similarities with the present study but also exhibits key distinctions. Several works have approached temporal text summarization as a constrained multi-objective optimization problem, a graph optimization problem, a supervised learning-to-rank problem, and as an online clustering problem.

This study models highlight detection as a simpler two-objective optimization problem with specific constraints. However, the features employed to assess the "highlightness" of a shot diverge from those used in the aforementioned studies. Given that highlight shots are observed to correlate with high emotional intensity and topic concentration, coverage and non-redundancy are not primary optimization goals, as they are in temporal text summarization. Instead, our focus is on modeling emotional and topic concentration within the context of this study.

2.3 Crowdsourced time-sync comment mining

Several studies have explored the use of crowdsourced time-synchronized comments for tagging videos on a shot-by-shot basis. These approaches involve manual labeling and supervised training,

temporal and personalized topic modeling, or tagging the video as a whole. One work proposes generating a summarization for each shot through data reconstruction that jointly considers textual and topic levels.

One work proposed a centroid-diffusion algorithm to identify highlights. Shots are represented by latent topics found through Latent Dirichlet Allocation (LDA). Another method suggests using pre-trained semantic vectors of comments to cluster them into topics and subsequently identify highlights based on topic concentration. Additionally, they utilize predefined labels to train a classifier for highlight labeling. The current study differs from these two studies in several ways. First, before performing highlight detection, we apply a lag-calibration step to mitigate inaccuracies caused by comment delays. Second, we represent each scene using a combination of topic and emotion concentration. Third, we perform both highlight detection and labeling in an unsupervised manner.

2.4 Lexical chain

Lexical chains represent sequences of words that exhibit a cohesive relationship spanning multiple sentences. Early work on lexical chains used syntactic relationships of words from Roget's Thesaurus, without considering word sense disambiguation. Subsequent research expanded lexical chains by incorporating WordNet relations and word sense disambiguation. Lexical chains are also built utilizing word-embedded relations for disambiguating multi-word expressions. This study constructs lexical chains for accurate lag calibration, leveraging global word embedding.

3 Problem Formulation

The problem addressed in this paper can be formulated as follows: The input consists of a set of time-synchronized comments, denoted as $\mathcal{C} = \{c_1, c_2, c_3, \ldots, c_n\}$, along with their corresponding timestamps $\mathcal{T} = \{t_1, t_2, t_3, \ldots, t_n\}$ for a given video v. We are also given a compression ratio $\rho_{highlight}$ that determines the number of highlights to be generated, and a compression ratio $\rho_{summary}$ that specifies the number of comments to be included in each highlight summary. Our objective is twofold: (1) to generate a set of highlight shots $\mathcal{S}(v) = \{s_1, s_2, s_3, \ldots, s_m\}$, and (2) to produce highlight summaries $\Sigma(v) = \{C_1, C_2, C_3, \ldots, C_m\}$ that closely align with the ground truth. Each highlight summary C_i comprises a subset of the comments associated with that shot: $C_i = \{c_1, c_2, c_3, \ldots, c_k\}$. The number of highlight shots m and the number of comments in each summary k are determined by $\rho_{highlight}$ and $\rho_{summary}$, respectively.

4 Video Highlight Detection

This section introduces our proposed framework for detecting video highlights. We also describe two preliminary tasks: constructing a global word embedding for time-synchronized comments and building an emotion lexicon.

4.1 Preliminaries

Word-Embedding of Time-Sync Comments

As previously highlighted, a key challenge in analyzing time-synchronized comments is their semantic sparsity, stemming from the limited number of comments and their brevity. Two semantically related words might not appear related if they don't co-occur frequently within a single video. To address this, we construct a global word embedding based on a large collection of time-synchronized comments.

This word-embedding dictionary can be represented as: $\mathcal{D} = \{(w_1 : v_1), (w_2 : v_2), \ldots, (w_n : v_n)\}$, where w_i is a word, v_i is its corresponding word vector, and n is the vocabulary size of the corpus.

Emotion Lexicon Construction

Extracting emotions from time-synchronized comments is crucial for highlight detection, as emphasized earlier. However, traditional emotion lexicons are not directly applicable in this context due to the prevalence of internet slang specific to these platforms. For example, "23333" signifies laughter ("ha ha ha"), and "6666" expresses admiration ("really awesome"). Therefore, we construct an emotion lexicon tailored for time-synchronized comments, derived from the word-embedding

dictionary generated in the previous step. We begin by manually labeling words corresponding to the five basic emotion categories (happiness, sadness, fear, anger, and surprise) as seeds, selecting from the most frequent words in the corpus. The sixth emotion category, "disgust," is omitted due to its rarity in the dataset but can be easily incorporated for other datasets. We then expand this emotion lexicon by identifying the top N neighbors of each seed word in the word-embedding space. A neighbor is added to the seeds if it meets a minimum percentage of overlap $\theta_{overlap}$ with all seeds, with a minimum similarity score of sim_{min}. Neighbors are determined based on cosine similarity within the word-embedding space.

4.2 Lag-Calibration

This section details our method for lag calibration, which involves concept mapping, constructing word-embedded lexical chains, and performing the actual calibration.

Concept Mapping

To tackle semantic sparsity in time-synchronized comments and build lexical chains of semantically related words, we first map words with similar meanings to the same concept. Given a set of comments $\mathcal{C}$ for a video v, we define a mapping $\mathcal{F}$ from the vocabulary $\mathcal{V}_\mathcal{C}$ of comments $\mathcal{C}$ to a set of concepts $\mathcal{K}_\mathcal{C}$:

$$\mathcal{F} : \mathcal{V}_\mathcal{C} \to \mathcal{K}_\mathcal{C} \quad (|\mathcal{V}_\mathcal{C}| \geq |\mathcal{K}_\mathcal{C}|)$$

Specifically, the mapping $\mathcal{F}$ assigns each word w_i to a concept $k = \mathcal{F}(w_i)$ as follows:

$$\mathcal{F}(w_i) = \mathcal{F}(w_1) = \mathcal{F}(w_2) = \ldots = \mathcal{F}(w_{top\_n}) = k, \quad \exists k \in \mathcal{K}_\mathcal{C}$$

$$s.t. \quad \{w | w \in top\_n(w_i) \wedge \mathcal{F}(w) = k\} / |top\_n(w_i)| \geq \theta_{overlap}$$

$top\_n(w_i)$ returns the n nearest neighbors of word w_i based on cosine similarity. For each word w_i in the comments $\mathcal{C}$, we examine the percentage of its neighbors that have already been mapped to a concept k. If this percentage exceeds the threshold $\theta_{overlap}$, then word w_i and its neighbors are mapped to concept k. Otherwise, they are assigned to a new concept, represented by w_i itself.

Lexical Chain Construction

The next step involves constructing all lexical chains present in the time-synchronized comments for video v. This enables the calibration of lagged comments based on these chains. A lexical chain l_{ik} consists of a set of triples $l_{ik} = \{(w, t, c)\}$, where w is the actual word mentioned for concept k in comment c, and t is the timestamp of comment c. We create a lexical chain dictionary $\mathcal{L}_\mathcal{C}$ for the time-synchronized comments $\mathcal{C}$ of video v:

$$\mathcal{L}_\mathcal{C} = \{k_1 : (l_{11}, l_{12}, l_{13}, \ldots), k_2 : (l_{21}, l_{22}, l_{23}, \ldots), \ldots, k_n : (l_{n1}, l_{n2}, l_{n3}, \ldots)\}$$

where $k_i \in \mathcal{K}_\mathcal{C}$ represents a concept, and l_{ik} is the i-th lexical chain associated with concept k. The procedure for constructing these lexical chains is detailed in Algorithm 1.

Specifically, each comment in $\mathcal{C}$ can either be appended to an existing lexical chain or added to a new, empty chain. This decision is based on the comment's temporal distance from existing chains, controlled by the maximum silence parameter $t_{silence}$.

It's important to note that word senses within the constructed lexical chains are not disambiguated, unlike in most traditional algorithms. However, we argue that these lexical chains remain useful because our concept mapping is built from time-synchronized comments in their natural order. This progressive semantic continuity naturally reinforces similar word senses for temporally close comments. This continuity, combined with global word embedding, ensures the validity of our concept mapping in most scenarios.

Comment Lag-Calibration

With the lexical chain dictionary $\mathcal{L}_\mathcal{C}$ constructed, we can now calibrate the comments in $\mathcal{C}$ based on their respective lexical chains. Our observations indicate that the initial comment pertaining to a shot typically occurs within that shot, while subsequent comments may not. Therefore, we adjust the timestamp of each comment to match the timestamp of the first element within its corresponding lexical chain. If a comment belongs to multiple lexical chains (concepts), we select the chain with the highest score $score_{chain}$. The $score_{chain}$ is calculated as the sum of the frequencies of each word

in the chain, weighted by the logarithm of their global frequencies, denoted as $\log(\mathcal{D}(w).count)$. Consequently, each comment will be assigned to its most semantically significant lexical chain (concept) for calibration. The calibration algorithm is presented in Algorithm 2.

It's worth noting that if multiple consecutive shots, $\{s_1, s_2, \ldots, s_n\}$, contain comments with similar content, our lag-calibration method might shift many comments from shots $s_2, s_3, \ldots, s_n$ to the timestamp of the first shot, s_1, if these comments are connected through lexical chains originating from s_1. This is not necessarily a drawback, as it helps us avoid selecting redundant consecutive highlight shots and allows for the inclusion of other potential highlights, given a fixed compression ratio.

4.3 Shot Importance Scoring

In this section, we first segment comments into shots of equal temporal length, denoted as t_{shot}. We then model the importance of each shot, enabling highlight detection based on these importance scores.

A shot's importance is modeled as a function of two factors: comment concentration and commenting intensity. Regarding comment concentration, as mentioned earlier, both concept and emotional concentration contribute to highlight detection. For instance, a cluster of concept-concentrated comments like "the background music/bgm/soundtrack of this shot is classic/inspiring/the best" could indicate a highlight related to memorable background music. Similarly, comments such as "this plot is so funny/hilarious/lmao/lol/2333" might suggest a highlight characterized by a single concentrated emotion. Therefore, our model combines these two types of concentration. We define the emotional concentration $C_{emotion}(\mathcal{C}_s)$ of shot s based on time-synchronized comments $\mathcal{C}_s$ and the emotion lexicon $\mathcal{E}$ as follows:

$$C_{emotion}(\mathcal{C}_s) = 1 - \sum_{e=1}^{|\mathcal{E}|} p_e \log(p_e)$$

$$p_e = \frac{|\{w|w \in \mathcal{C}_s \wedge w \in \mathcal{E}(e)\}|}{|\mathcal{C}_s|}$$

Here, we calculate the inverse of the entropy of probabilities for the five emotions within a shot to represent emotion concentration. Next, we define topical concentration C_{topic} as:

$$C_{topic}(\mathcal{C}_s) = \frac{1}{J} \sum_{j=1}^{J} p_j \log(p_j)$$

$$p_j = \frac{\sum_{w \in \mathcal{C}_s \cap \mathcal{F}(k_j)} \frac{1}{\log(\mathcal{D}(w))}}{\sum_{w \in \mathcal{C}_s} \frac{1}{\log(\mathcal{D}(w))}}$$

where we calculate the inverse of the entropy of all concepts within a shot to represent topic concentration. The probability of each concept k is determined by the sum of the frequencies of its mentioned words, weighted by their global frequencies, and then divided by the sum of these weighted frequencies for all words in the shot.

Now, the comment importance $I_{comment}(\mathcal{C}_s, s)$ of shot s can be defined as:

$$I_{comment}(\mathcal{C}_s, s) = \lambda \cdot C_{emotion}(\mathcal{C}_s, s) + (1 - \lambda) \cdot C_{topic}(\mathcal{C}_s, s)$$

where λ is a hyperparameter that controls the balance between emotion and concept concentration.

Finally, the overall importance of a shot is defined as:

$$I(\mathcal{C}_s, s) = I_{comment}(\mathcal{C}_s, s) \cdot \log(|\mathcal{C}_s|)$$

where $|\mathcal{C}_s|$ represents the total length of all time-synchronized comments within shot s, serving as a straightforward yet effective indicator of comment intensity per shot.

The problem of highlight detection can now be formulated as a maximization problem:

Maximize $\sum_{s \in \mathcal{S}} I(\mathcal{C}_s, s)$

Subject to $|\mathcal{S}| \leq \rho_{highlight} \cdot N$

5 Video Highlight Summarization

Given a set of detected highlight shots $\mathcal{S}(v) = \{s_1, s_2, s_3, \ldots, s_m\}$ for video v, each associated with its lag-calibrated comments $\mathcal{C}_s$, our goal is to generate summaries $\Sigma(v) = \{C_1, C_2, C_3, \ldots, C_m\}$ such that $C_i \subset \mathcal{C}_{s_i}$, with a compression ratio of $\rho_{summary}$, and C_i closely resembles the ground truth.

We propose a simple yet highly effective summarization model, building upon SumBasic with enhancements that incorporate emotion and concept mapping, along with a two-level updating mechanism.

In our modified SumBasic, instead of solely down-weighting the probabilities of words in a selected sentence to mitigate redundancy, we down-weight the probabilities of both words and their mapped concepts to re-weight each comment. This two-level updating approach achieves two key objectives: (1) it penalizes the selection of sentences containing semantically similar words, and (2) it allows for the selection of a sentence with a word already present in the summary if that word occurs significantly more frequently. Additionally, we introduce an emotion bias parameter, $b_{emotion}$, to weight words and concepts during probability calculations. This increases the frequencies of emotional words and concepts by a factor of $b_{emotion}$ compared to non-emotional ones.

6 Experiment

This section presents the experiments conducted on large-scale real-world datasets to evaluate highlight detection and summarization. We describe the data collection process, evaluation metrics, benchmark methods, and experimental results.

6.1 Data

This section describes the datasets collected and constructed for our experiments. All datasets and code will be made publicly available on Github.

Crowdsourced Time-sync Comment Corpus

To train the word embedding described earlier, we collected a large corpus of time-synchronized comments from Bilibili, a content-sharing website in China that features such comments. The corpus comprises 2,108,746 comments, 15,179,132 tokens, and 91,745 unique tokens, extracted from 6,368 long videos. On average, each comment contains 7.20 tokens.

Before training, each comment undergoes tokenization using the Chinese word tokenization package Jieba. Repeated characters within words, such as "233333," "66666," and "ǩ54c8ǩ54c8ǩ54c8ǩ54c8," are replaced with two instances of the same character.

The word embedding is trained using word2vec with the skip-gram model. We set the number of embedding dimensions to 300, the window size to 7, and the down-sampling rate to 1e-3. Words with a frequency lower than 3 are discarded.

Emotion Lexicon Construction

After training the word embedding, we manually select emotional words belonging to the five basic emotion categories from the 500 most frequent words in the embedding. We then iteratively expand these emotion seeds using Algorithm 1. After each expansion iteration, we manually review the expanded lexicon, removing any inaccurate words to prevent concept drift. The filtered expanded seeds are then used for further expansion in the next round. The minimum overlap $\theta_{overlap}$ is set to 0.05, and the minimum similarity sim_{min} is set to 0.6. These values are determined through a grid search within the range of [0, 1]. The number of words for each emotion, both initially and after the final expansion, is presented in Table 3.

Video Highlights Data

To evaluate our highlight detection algorithm, we constructed a ground-truth dataset. This dataset leverages user-uploaded mixed-clips related to a specific video on Bilibili. Mixed-clips represent a collection of video highlights chosen according to the user's preferences. We then consider the most frequently selected highlights as the ground truth for a given video.

Table 1: Number of Initial and Expanded Emotion Words

	Happy	Sad	Fear	Anger	Surprise
Seeds	17	13	19	21	14
All	157	235	258	284	226

The dataset consists of 11 videos totaling 1333 minutes in length, with 75,653 time-synchronized comments. For each video, 3-4 video mix-clips are collected from Bilibili. Shots that appear in at least two of these mix-clips are considered ground-truth highlights. These highlights are mapped to the original video timeline, and their start and end times are recorded as ground truth. Mix-clips are selected based on the following criteria: (1) they are found on Bilibili using the search query "video title + mixed clips"; (2) they are sorted by play count in descending order; (3) they primarily focus on video highlights rather than a plot-by-plot summary or gist; (4) they are under 10 minutes in length; and (5) they contain a mix of several highlight shots instead of just one.

On average, each video contains 24.3 highlight shots. The mean duration of these highlight shots is 27.79 seconds, while the mode is 8 and 10 seconds (with a frequency of 19).

Highlights Summarization Data

We also created a highlight summarization (labeling) dataset for the 11 videos. For each highlight shot and its associated comments, we asked annotators to create a summary by selecting as many comments as they deemed necessary. The guiding principles were: (1) comments with identical meanings should not be selected more than once; (2) the most representative comment among similar comments should be chosen; and (3) comments that stand out and are irrelevant to the current discussion should be discarded.

Across the 11 videos and 267 highlights, each highlight has an average of 3.83 comments in its summary.

6.2 Evaluation Metrics

This section introduces the evaluation metrics employed for both highlight detection and summarization.

Video Highlight Detection Evaluation

To evaluate video highlight detection, we need to define a "hit" between a candidate highlight and a reference highlight. A strict definition would require a perfect match between the start and end times of the candidate and reference highlights. However, this criterion is overly stringent for any model. A more lenient definition would consider an overlap between a candidate and a reference highlight. However, this can still underestimate model performance, as users' choices of highlight start and end times can sometimes be arbitrary. Instead, we define a "hit" with a relaxation parameter δ between a candidate h and the reference set R as follows:

$hit(h, R) = \{\, 1\ \exists r \in R : (s_h, e_h) \cap (s_r - \delta, e_r + \delta) \neq \emptyset$
$0 otherwise$

where s_h, e_h represent the start and end times of highlight h, and δ is the relaxation length applied to the reference set R. We can then define precision, recall, and F1-score as:

$Precision(H, R) = \frac{\sum_{h \in H} hit(h,R)}{|H|}$

$Recall(H, R) = \frac{\sum_{r \in R} hit(r,H)}{|R|}$

$F1(H, R) = \frac{2 \cdot Precision(H,R) \cdot Recall(H,R)}{Precision(H,R) + Recall(H,R)}$

In this study, we set the relaxation length δ to 5 seconds. The candidate highlight length is set to 15 seconds.

Video Highlight Summarization Evaluation

We utilize ROUGE-1 and ROUGE-2 as recall metrics for evaluating candidate summaries:

$$ROUGE-n(C,R) = \frac{\sum_{r \in R} \sum_{n-gram \in r} Count_{match}(n-gram)}{\sum_{r \in R} \sum_{n-gram \in r} Count(n-gram)}$$

We employ BLEU-1 and BLEU-2 as precision metrics. BLEU is chosen for two reasons. First, a naive precision metric would be biased towards shorter comments, and BLEU mitigates this with the BP (Brevity Penalty) factor:

$$BLEU-n(C,R) = BP \cdot \frac{\sum_{c \in C} \sum_{n-gram \in c} Count_{clip}(n-gram)}{\sum_{c \in C} \sum_{n-gram \in c} Count(n-gram)}$$

$$BP = \{ 1 \; if |C| > |R| \\ e^{(1-|R|/|C|)} if |C| \leq |R|$$

where C is the candidate summary and R is the reference summary. Second, while the reference summary contains no redundancy, the candidate summary might incorrectly select multiple similar comments that match the same keywords in the reference. In such cases, precision would be significantly overestimated. BLEU addresses this by counting matches one-by-one; the number of matches for a word will be the minimum of its frequencies in the candidate and reference summaries.

Finally, the F1-score is defined as:

$$F1-n(C,R) = \frac{2 \cdot BLEU-n(C,R) \cdot ROUGE-n(C,R)}{BLEU-n(C,R) + ROUGE-n(C,R)}$$

6.3 Benchmark methods

Benchmarks for Video Highlight Detection

For highlight detection, we compare different combinations of our model against three benchmark methods:

* **Random-Selection:** Highlight shots are randomly selected from all shots in a video. * **Uniform-Selection:** Highlight shots are selected at equal intervals. * **Spike-Selection:** Highlight shots are chosen based on the highest number of comments within the shot. * **Spike+E+T:** This is our method, incorporating emotion and topic concentration but without lag calibration. * **Spike+L:** This is our method, including only the lag-calibration step and not considering content concentration. * **Spike+L+E+T:** This represents our full model.

Benchmarks for Video Highlight Summarization

For highlight summarization, we compare our method against five benchmark methods:

* **SumBasic:** Summarization that relies solely on frequency for summary construction. * **Latent Semantic Analysis (LSA):** Text summarization based on singular value decomposition (SVD) for latent topic discovery. * **LexRank:** Graph-based summarization that calculates sentence importance using the concept of eigenvector centrality in a sentence graph. * **KL-Divergence:** Summarization based on minimizing KL-divergence between the summary and the source corpus, employing a greedy search approach. * **Luhn method:** A heuristic summarization method that considers both word frequency and sentence position within an article.

6.4 Experiment Results

This section presents the experimental results for both highlight detection and highlight summarization.

Results of Highlight Detection

In our highlight detection model, the maximum silence threshold for lexical chains, $t_{silence}$, is set to 11 seconds. The threshold for concept mapping, $\theta_{overlap}$, is set to 0.5. The number of neighbors considered for concept mapping, top_n, is set to 15. The parameter λ, which controls the balance between emotion and concept concentration, is set to 0.9. A detailed parameter analysis is provided in Section 7.

Table 4 presents the precision, recall, and F1-scores for different combinations of our method and the benchmark methods. Our full model (Spike+L+E+T) outperforms all other benchmarks across all metrics. Random and uniform selection exhibit low precision and recall, as they don't incorporate structural or content information. Spike-selection shows significant improvement by leveraging

comment intensity. However, not all comment-intensive shots are highlights. For example, comments at the beginning and end of a video are often high-volume greetings or goodbyes, which may not be indicative of highlights. Additionally, spike-selection tends to cluster highlights within consecutive shots with high comment volumes. In contrast, our method can identify less intensive but emotionally or conceptually concentrated shots that might be missed by spike-selection. This is evident in the performance of Spike+E+T.

We also observe that lag calibration alone (Spike+L) considerably enhances the performance of Spike-selection, partially supporting our hypothesis that lag calibration is crucial for tasks involving time-synchronized comments.

Table 2: Comparison of Highlight Detection Methods

Method	Precision	Recall	F1-score
Random-Selection	0.1578	0.1567	0.1587
Uniform-Selection	0.1797	0.1830	0.1775
Spike-Selection	0.2594	0.2167	0.2321
Spike+E+T	0.2796	0.2357	0.2500
Spike+L	0.3125	0.2690	0.2829
Spike+L+E+T	0.3099	0.3071	0.3066

Results of Highlight Summarization

In our highlight summarization model, the emotion bias $b_{emotion}$ is set to 0.3.

Table 5 compares the 1-gram BLEU, ROUGE, and F1-scores of our method and the benchmark methods. Our method outperforms all others, particularly in terms of ROUGE-1. LSA exhibits the lowest BLEU score, primarily because it statistically favors longer, multi-word sentences, which are not representative in time-synchronized comments. The SumBasic method also performs relatively poorly, as it treats semantically related words separately, unlike our method, which uses concepts instead of individual words.

Table 3: Comparison of Highlight Summarization Methods (1-Gram)

Method	BLEU-1	ROUGE-1	F1-1
LSA	0.2382	0.4855	0.3196
SumBasic	0.2854	0.3898	0.3295
KL-divergence	0.3162	0.3848	0.3471
Luhn	0.2770	0.4970	0.3557
LexRank	0.3045	0.4325	0.3574
Our method	0.3333	0.6006	0.4287

7 Conclusion

This work presents a novel unsupervised framework for video highlight detection and summarization, based on crowdsourced time-synchronized comments. We introduce a lag-calibration technique that re-aligns delayed comments to their corresponding video scenes by using concept-mapped lexical chains. Video highlights are identified based on comment intensity and the concentration of concepts and emotions within each shot. For summarization, a two-level SumBasic is proposed which updates word and concept probabilities iteratively when selecting sentences. Future work includes integrating additional data sources such as video meta-data, audience profiles, and low-level multi-modal features.