Reading the Videos: Temporal Labeling for Crowdsourced Time-Sync Videos Based on Semantic Embedding

Guangyi Lv[†], **Tong Xu**[†], **Enhong Chen**[†*], **Qi Liu**[†], **Yi Zheng**[‡]
[†]School of Computer Science and Technology, University of Science and Technology of China
gylv@mail.ustc.edu.cn, tongxu@mail.ustc.edu.cn, cheneh@ustc.edu.cn, qiliuql@ustc.edu.cn
[‡]Ant Financial Services Group
aliguagua.zhengy@alipay.com

Abstract

Recent years have witnessed the boom of online sharing media contents, which raise significant challenges in effective management and retrieval. Though a large amount of efforts have been made, precise *retrieval on video shots* with certain topics has been largely ignored. At the same time, due to the popularity of novel time-sync comments, or so-called **"bullet-screen comments"**, video semantics could be now combined with timestamps to support further research on temporal video labeling. In this paper, we propose a novel video understanding framework to assign temporal labels on highlighted video shots. To be specific, due to the informal expression of bullet-screen comments, we first propose a *temporal deep structured semantic* model (T-DSSM) to represent comments into semantic vectors by taking advantage of their temporal correlation. Then, video highlights are recognized and labeled via semantic vectors in a supervised way. Extensive experiments on a real-world dataset prove that our framework could effectively label video highlights with a significant margin compared with baselines, which clearly validates the potential of our framework on video understanding, as well as bullet-screen comments interpretation.

1 Introduction

Recently, the booming of online video-sharing websites raises significant challenges in effective management and retrieval of videos. Though a large amount of efforts have been made on automatic labeling to enrich the metadata, usually they focus on the whole video, while precise *labeling with timestamp* on video shots has been largely ignored. Considering the situation that sometimes users tend to view only parts of video shots on certain topics, e.g., shots about a certain style or a certain movie star, however, they have to browse the whole video if without assistance of temporal labels. Thus, video labeling with both semantics and timestamps is urgently required.

At the same time, thanks to the emergence of the novel time-sync comments, or so-called "**bullet-screen comments**", real-time comments on video shots are now available, e.g., niconico in Japan or Bilibili in China. The "bullet-screen" is named after the scene of comments flying across the screen, which is similar with the barrage of bullets. Intuitively, users will send a bullet-screen comment according to the current video shot, thus new opportunities occur for temporal video labeling, as video contents could now be accurately located with timestamps.

Though some prior arts have been carried out on bullet-screen comments analysis, e.g., (Lin, Ito, and Hirokawa 2014) and (Wu and Ito 2014), usually they just focus on simple statistics that could hardly solve practical problems. Some other works like (Wu et al. 2014) attempt to semantically understand comments, which may fail due to informal expressions. For instance, "2333" presents laughing while "high-energy" means climax, which cannot be interpreted by literal meanings. To that end, in this paper, we propose a novel video understanding framework to better interpret the bullet-screen comments, and then assign temporal label to highlight video shots.

Specially, to deal with the informal expression of bullet-screen comments, we design a *temporal deep structured semantic* model (T-DSSM) to represent comments into semantic vectors, where the model is trained via EM algorithm by harnessing the *temporal correlation* between comments. Further, we build the mapping from semantic vectors to the pre-defined labels in a supervised way, while video highlights will then be recognized and labeled. To the best of our knowledge, we are among the first ones who utilize bullet-screen comments to recognize and label videos in a supervised way. Moreover, we design an EM based learning algorithm for training T-DSSM to achieve semantic embedding, which is proven effective.

The extensive experiments on real-world dataset prove that our framework could effectively label video highlights with a significant margin compared with baselines, which validates the potential of our framework on video understanding, as well as bullet-screen comments interpretation.

2 Problem Definition and Framework

In this paper, we target at finding and labeling video "**highlights**", i.e., video shots focusing on certain topics (labels). Intuitively, this task could be solved in a supervised way, which is defined mathematically as follow:

Definition 1 *Given the training set of videos with bullet-screen comments* $\mathbf{C_{train}} = \{< text, time >\}$*, as well*

*Corresponding author.

Table 1: Mathematical Notations.

SYMBOL	DESCRIPTION
$\mathbf{C} = \{< text, time >\}$	bullet-screen comments set
$\mathbf{L} = \{< t_s, t_e, lt >\}$	temporal labels
c_i	a piece of bullet-screen comment
$\mathbf{v}_i$	corresponding semantic vector of c_i
s_j	a highlighted video shot
$\mathbf{f}_j$	corresponding feature vector of s_i

as temporal labels $\mathbf{L_{train}} = \{< t_s, t_e, lt >\}$ *in which* $< t_s, t_e >$ *indicates the timestamps (start and end) and* lt *presents the label type, the target is to precisely assign temporal labels* $\mathbf{L_{predict}} = \{< t_s^{'}, t_e^{'}, lt^{'} >\}$ *to the test set* $\mathbf{C_{test}}$*, where each* $< t_s^{'}, t_e^{'} >$ *indicates a video highlight with corresponding label as* $lt^{'}$*.*

To solve this task, we should first deal with the informal expression of bullet-screen comments by representing them as semantic vectors, and then build the mapping from semantic vectors to temporal labels for recognizing and labeling video highlights. The two-stage framework will be formulated as follows:

Semantic embedding stage. In this stage, we design the "Temporal Deep Structured Semantic Model" (T-DSSM) to represent each bullet-screen comment c_i as corresponding semantic vector $\mathbf{v}_i$ via semantic embedding.

Highlight understanding stage. In this stage, we focus on highlight recognizing and labeling in a supervised way. To be specific, we: 1) Split video stream into *slides* in equal length as $\mathbf{S}$, which will be treated as the basic unit in our framework. 2) $\mathbf{S_{train}}$ in training samples with corresponding labels will be trained to building the mapping ξ from semantic vector $\mathbf{v}_i$ to semantic label lt_k. 3) For each slide $s_i \in \mathbf{S_{test}}$, we determine whether the slide is concentrated in certain topics, then label it via ξ to achieve corresponding lk_i. 4) We merge them according to the same lt_i to achieve the final $\mathbf{L_{predict}}$.

Related technical details will be discussed in Section 3 and Section 4. Also, some relevant notations are summarized in Table 1.

3 Comment Semantic Embedding

In this section, we will discuss how to achieve semantic presentation for bullet-screen comments via Temporal Deep Structured Semantic Models (T-DSSM). To be specific, we will first introduce the overall T-DSSM architecture, and then explain the details of model learning.

First of all, we focus on the architecture of T-DSSM model. As shown in Figure. 1, our model is based on DSSM, which is a typical DNN (Huang et al. 2013). In this architecture, all text of a comment c will be first encoded into raw bag-of-words feature, and then put into the input layer. We choose to put the entire sentence into the network, since bullet-screen comments are usually short phrases or fixed allusions, which is difficult to be split. Also, there are informal expressions exist, which could hardly be solved by traditional NLP techniques. Finally, we have the output as

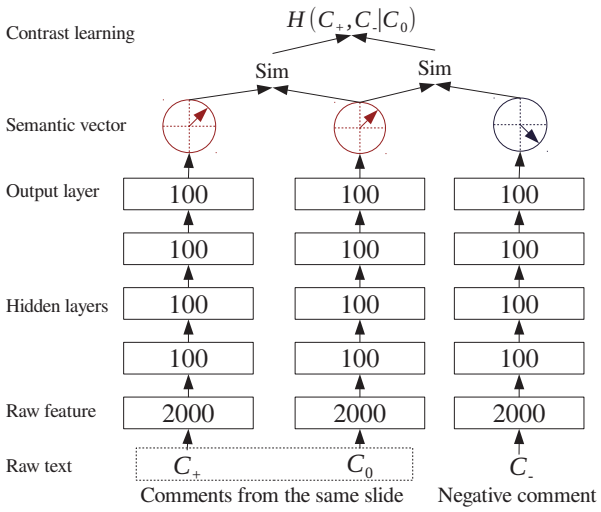


Figure 1: The architecture of T-DSSM with 3 hidden layers.

semantic vector $\mathbf{v}$ which indicate the comment's representation in semantic space.

In detail, we have a DNN with 3 hidden layers in which each of them has 100 neurons. The input vector, also known as the raw features of comment, contain 2000 dimensions in total, while for the semantic embedding output, we have 100-dimensional vectors. Besides, we have $\tanh$ used as activation function for all layers.

As the overall architecture has been proposed, we then turn to introduce how to learn the T-DSSM model. To initialize the model, we first randomly set the weights of neural network following the uniform distribution in the range between $-\sqrt{6/(nin + nout)}$ and $\sqrt{6/(nin + nout)}$ as suggested in (Orr and Müller 2003), where nin and $nout$ separately refers to the number of input and output neurons. Then, the T-DSSM will be trained via optimizing a loss function with respect to its parameters, which is well-known as back-propagation (BP) (Hecht-Nielsen 1989).

Our loss function is designed based on contrast learning, i.e., for each comment c_0, we target at maximizing the similarities between semantic vectors of c_0 and those positive samples presented as c_+. On the contrary, the similarities between c_0 and negative samples, labeled as c_-, should be minimized. So, the objective function could be formulated as below:

$$H(c_+, c_-|c_0) = \frac{e^{Sim(c_+, c_0)}}{e^{Sim(c_+, c_0)} + e^{Sim(c_-, c_0)}}, \qquad (1)$$

where $Sim(c_1, c_2)$ means the cosine similarity between two their semantic vectors $\mathbf{v_1}, \mathbf{v_2}$ obtained from the DNN' s feed forward stage.

Intuitively, as we realize that bullet-screen comments contain so called "**temporal correlation**", i.e., when users send a bullet-screen comment, they may refer to the current video shot as well as *previous comments*, thus, semantic vectors of adjacent comments could be reasonably similar. Based on this phenomenon, we set an area $\mathbf{A_0}$ around c_0 on the video

stream, where $\mathbf{A_0}$ has n seconds in length, and comments within $\mathbf{A_0}$ will be treated as semantically similar. Then, we can choose c_+ from $\mathbf{A_0}$.

But for c_-, there is no guarantee that comments outside $\mathbf{A_0}$ are not similar to c_0 and to our pilot study, simply selecting comments outside $\mathbf{A_0}$ to train the model may result in difficulty in convergence. One better choice is to regard c_- as latent variable, and instead of maximizing $H(c_+, c_-|c_0)$ directly, parameters are estimated by maximum likelihood estimate (MLE) determined by the marginal likelihood of the observable c_+ through EM algorithm (Dempster, Laird, and Rubin 1977):

$$L(\theta) = P(c_+|c_0) = \sum_{c_-} P(c_+, c_-|c_0), \tag{2}$$

in which

$$P(c_+, c_-|c_0) = \frac{e^{H(c_+, c_-|c_0)}}{\sum_{c'_+} \sum_{c'_-} e^{H(c'_+, c'_-|c_0)}}. \tag{3}$$

At last, the training procedure can be described as: For every comment c_0, we repeat the following steps until the model converges.

Sampling Step. Find the positive set $\mathbf{C}_+ = \{c_+\}$ which contains all possible comments in $\mathbf{A_0}$. Then, sample negative set $\mathbf{C}_- = \{c_-\}$ randomly outside $\mathbf{A_0}$ but ensure that $\mathbf{C}_-$ has the same size with $\mathbf{C}_+$.

E Step. Calculate the expected value of the log likelihood function, with respect to the conditional distribution of c_- given c_+ under the current estimate of the parameters $\theta^{(t)}$, where

$$Q(\theta|\theta^{(t)}) = E_{c_-|c_+}\left[logP(c_+, c_-|c_0)\right], \tag{4}$$

while the distribution of c_- given c_+ is defined as:

$$P(c_-|c_+) = \frac{e^{Sim(c_+, c_-)}}{\sum_{c'_- \in \mathbf{C}_-} e^{Sim(c_+, c'_-)}}. \tag{5}$$

M Step. Update the parameter which maximizes $Q(\theta|\theta^{(t)})$ via Stochastic Gradient Ascent (SGA).

Based on EM steps, our likelihood function can be maximized through iteratively maximizing its lower bound, which ensures the convergence of our model and the performance of semantic embedding. Furthermore, since the number of comments in $\mathbf{A_0}$ is relatively small, sizes of both $\mathbf{C}_+$ and $\mathbf{C}_-$ are also limited, which means the computational cost of training process is acceptable. Even though we sample $\mathbf{C}_-$ to approximate $P(c_-|c_+)$, to our observation, sampling $\mathbf{C}_-$ with the same size as $\mathbf{C}_+$ can already gain a good performance compared with optimizing function $H(c_+, c_-|c_0)$ directly.

4 Highlight Understanding

In this section, we will focus on recognizing and labeling video highlights based on comments' semantic vectors.

Preparation

As temporal label not only contains type information, but also time range in the video, we first set a m seconds time-window to split the video stream into slides as illustrated in Figure. 2. For each slide, we treat it as the basic unit and extract its feature for labeling.

In our case, the feature is presented as latent topics revealed from clustering semantic vectors, where the latent topics will benefit for further analysis: 1) The reduction in dimensionality of semantic vectors with reserving discriminating parts will further improve the effectiveness of classifiers. 2) It provides an intuitive explanation of similarities between bullet-screen comments, which will benefit the discussion on semantic understanding and labeling. 3) The latent factors could better indicate the concentration of topics within a single slide.

For simplicity, we conduct the classical DBSCAN algorithm to cluster semantic vectors. We tune its parameters by keeping the cluster number k which is also known as the number of latent topics ,while the strategy to select k will be discussed in experimental part.

After clustering, we will turn to introduce the recognizing, labeling and merging steps.

Recognizing Step

As all preparatory steps are done, we now turn to recognize the highlight shots in videos, since compared with those all-inclusive shots, we tend to highlight shots which concentrate on fewer topics for labeling as more information could be gained. So, we label each comment with the corresponding cluster (topic) and for each slide, we could simply calculate comment frequency on each topic and denote it as feature $\mathbf{f}$. Specially, if we have a semantically concentrating slide, for the topic frequencies, they may have higher variance and lower information entropy, thus, we formulate the *concentrating rating* as follow:

$$rating = \frac{\frac{\sum_i^k (f_i - \bar{f})}{k}}{\sum_{\mathbf{p}} -p\log(p)}, \tag{6}$$

in which $\mathbf{p}$ is normalized form of $\mathbf{f}$, i.e., $p_i = \frac{f_i}{\sum_j f_j}(f_i \neq 0)$, and $\sum_{\mathbf{p}} -p\log(p)$ here is indeed the entropy of $\mathbf{p}$. Then, slides with their concentrating ratings larger then a threshold will be recognized as highlight slides $\mathbf{S_{highlight}} = \{< t_s, t_e, \mathbf{f} >\}$, where t_s, t_e indicates the time range of slide, $\mathbf{f}$ means the comment frequency on topics.

Note that the threshold here is set dynamically in different videos. For each video, we can calculate a series of ratings for the slides and then find the max and min. The threshold is set as $\alpha * min + (1 - \alpha) * max$, $(0 \leq \alpha \leq 1)$, where α is called *pass rate* and the sensitiveness of α will also be discussed in experimental part.

Labeling Step

The highlight slides $\mathbf{S_{highlight}} = \{< t_s, t_e, \mathbf{f} >\}$ are now obtained, and we turn to label them with our preset highlight types in a supervised way. First, we will train a classifier.

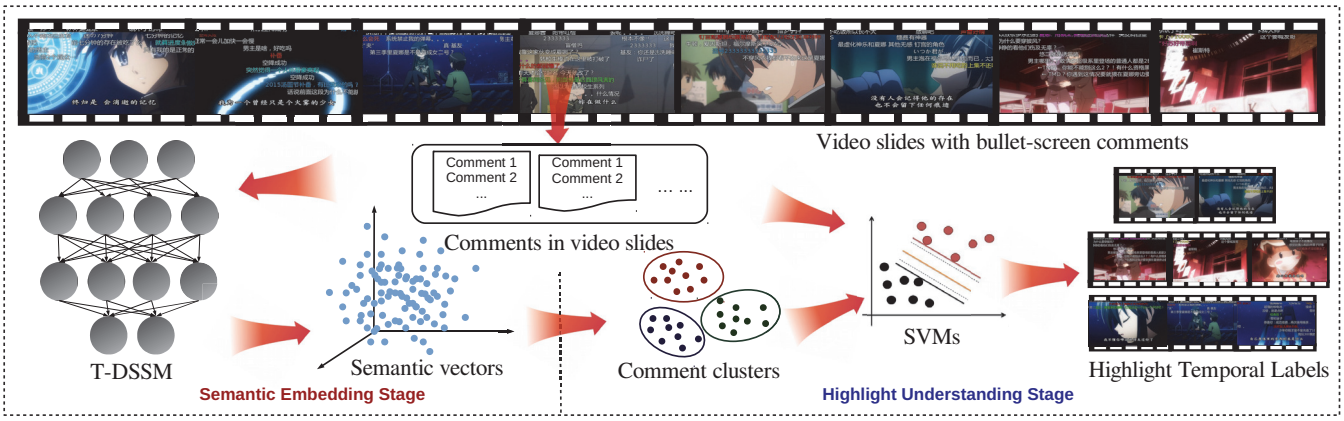


Figure 2: Illustration of the labeling framework.

Considering there is a training set with bullet-screen comments $\mathbf{C_{train}}$ and a series of existing labels $\mathbf{L_{train}} = \{< t_s, t_e, lt >\}$, we split $\mathbf{C_{train}}$ into slides and then calculate $\mathbf{f}$ for each slide. After that, $\mathbf{f}$ is used as feature and a classifier $\xi(\mathbf{f}) \rightarrow lt$ is learnt mapping $\mathbf{f}$ to lt.

When we get the classifier, every slide $< t_s, t_e, \mathbf{f} >$ in $\mathbf{S_{highlight}}$ can be labeled with a human understandable lt through ξ, and finally we get $\mathbf{S_{labeled}} = \{< t_s, t_e, lt >\}$. In our case, SVMs is chosen to perform robust and efficient multi-classification.

Merging Step

Finally, we design an easy heuristic method to merge the adjacent slides in $\mathbf{S_{labeled}}$. For any adjacent slide pair $< t_{s1}, t_{e1}, lt_1 >$ and $< t_{s2}, t_{e2}, lt_2 >$, i.e., $t_{s2} = t_{e1}$, if we have $lt_1 = lt_2$, we will merge these two slides into a new one $< t_{s1}, t_{e2}, lt_1 >$. After all slides in $\mathbf{S_{labeled}}$ are merged, we finally get temporal labels $\mathbf{L_{predict}}$.

5 Experiments

To validate our framework, in this section, we will conduct experiments compared with several baselines. Then we further discuss the parameter sensitiveness and some phenomenons observed in our experiments.

Dataset Preparation

We choose to illustrate our work on a real-world data set extracted from Bilibili [1], which is one of the largest video-sharing platforms focusing on animation, TV series and game videos. Specially, totally 133,250 comments are extracted, corresponds to 1,600 minutes long videos of different types of animation. For each bullet-screen comment, we obtain the text and time of occurrence, i.e., in the form of $\mathbf{C} = \{< text, time >\}$.

For data pre-processing, we first filter those comments that are extremely long. Based on our observation, the average length of bullet-screen comments is 8.57 and most of comments contain less than 20 characters. Thus, comments that longer than 40 characters will be regarded as noise and then deleted. Also, the invisible comment that only contains space characters is also removed. Finally, we merge those comments with only differences in suffix with repeated words. For instance, "23333" and "23333333", or "yoooo" and "yooooooo" will be merged.

We randomly select half of commented videos as training set $\mathbf{C_{train}}$ and others as testing set $\mathbf{C_{test}}$.

Experimental Setup

In our framework, parameter n which is the size of $\mathbf{A_0}$ in T-DSSM is set as 5 seconds since it might be the most suitable length in majority of cases according to our observation, and also, it is the time for comments to fly across the screen. In highlight understanding stage, the window size m to split the video stream is set as twice as n, and we respectively set the number of clusters k and the pass rate α as 26 and 0.3. We will explain how to determine parameters k and α in detail later. As few works about highlight labeling has been done before, to evaluate the performance of our framework, we consider 2 straightforward baseline models to compare with: Word based and LDA (Blei, Ng, and Jordan 2003) based.

Word based. In this model, we generate a distribution of words instead of latent topics for each window-slide. Thus, the highlight recognizing and labeling can be performed based on the topics via steps mentioned in Section 4. The pass rate α is set to be 0.3 as it achieves the best performance for this model.

LDA based. Each slide of comments is regarded as a document in topic model. LDA is used to obtain the distribution of topics for each slide. The number of topics is set as 26 at first which is identical to the number of clusters in our framework, and then tuned for acquiring best performance. After at least 1000 loops of iteration, distribution on topics is generated for each slide and then we obtain temporal labels through the approach as described in Section 4.

To evaluate the performance we have three experts who are professional in Japanese anime to label the training samples in fold ways. In detail, they were required to watch the episodes that were randomly selected from videos, and at the same time, they were also given some pre-defined labels.

[1] http://www.bilibili.com/

Due to the limitation of extracted samples, we only select 10 types here as ground truth to generally validate the potential of our framework, while more labels could be easily added when more videos extracted. These labels contain types describe scenes, e.g., "funny", "moving", "surprising", "sad", "magnificent fighting" and types describe characters, e.g., "cool", "lovely girl", "sexy shot" and even type about music ,e.g., "OP", "BGM". Once they considered a plot is meaningful that matches the description of one from the given labels, they would take a record of the start and end time in the form of $< t_s^+, t_e^+, lt^+ >$. Thus, we can obtain the series of labels $\mathbf{L}^+ = \{< t_{s1}^+, t_{e1}^+, lt_1^+ >\}$ as the ground truth for further evaluation.

In order to measure the overall performance, we define the metric as hit_{time} as the overlap of labeling result with ground truth as follow:

$$hit_{time} = \sum_{L_i.lt=L_j^+.lt} L_i \cap L_j^+, \tag{7}$$

where $L_i \cap L_j^+$ represents the overlapped time of the two labels. Then precision, recall and F1 score are available. Moreover, if only length is focused, when a long shot is correctly labeled, some short but significant shots might be ignored. Thus, we further design other two sets of metrics, i.e., $Precision$ / $Recall$ of labels, which measures the performance of semantic label discovery, and also the $Recall$ of shots, which weight the effectiveness of highlight mining.

In detail, $Precision$ / $Recall$ of labels is based on hit_{label} where hit_{label} is defined as the number of different types in **hit shots** which has the same highlight type with the ground truth label L^+ with their overlapped time longer than 80% the length of L^+. $Recall$ of shots is defined through hit_{shot} which is the number of hit shots.

Overall Results

The overall experimental results of these models are summarized in Table 2. We can see that our T-DSSM based framework outperforms the other models in all of four metrics. No surprisingly, word based method achieves the lowest precision since it totally ignores the comment content. LDA shows a little improvement, and both of them infer comments' topic based on words. As words can not well reflect the meaning of topics, their performances still have a big margin compared to ours. Furthermore, since they could not represent the actual meaning of comments sometimes, both of them are comparatively week in discover labels with rare types, so their $Precision_{label}$ and $Recall_{label}$ is obviously lower than ours. In contrast, our model uses semantic vectors which can be better in discovering topics from comments since they are able to present the internal relationship among comments even though they have no literal association. Thus, it is proved to be reasonable that T-DSSM based framework is more suitable for understanding videos by comments with temporal correlation and latent meaning.

Sensitivity of Parameters

Since the performance of our framework may be affected by the parameters, it is crucial for us to determine the parameters and also necessary to analyze the impacts of parameters on the final performance. Basically, there are two parameters in our model to be determined, i.e., the number of clusters (latent topics) k and the pass rate α used to determine highlight time range.

Table 2: Performance of these models.

Model	Word based	LDA	**T-DSSM**
$Precision$	0.3509	0.3695	**0.4575**
$Recall$	0.3885	0.4013	**0.4969**
$F1$	0.3687	0.3847	**0.4764**
$Precision_{label}$	0.4992	0.5139	**0.6103**
$Recall_{label}$	0.4452	0.4547	**0.5738**
$Recall_{shot}$	0.3486	0.3669	**0.4770**

For the former, we followed the previous studies and adopted the average distance within clusters to measure the quality of clustering. As shown in Figure. 3(a), we evaluate the average distance for different k. When the number of clusters k grows from 2 to 20, the average distance also considered as the loss of clustering reduces rapidly. After k is greater than 30, this loss tends to be stable, which guides us to further fine tune the parameter k based on the other metrics $Precision$ and $Recall$. As shown in Figure. 3(b), we evaluate our model with different k from 22 to 28 in $Precision$ and $Recall$. For other metrics, we omit their figures due to the similarity of the results.

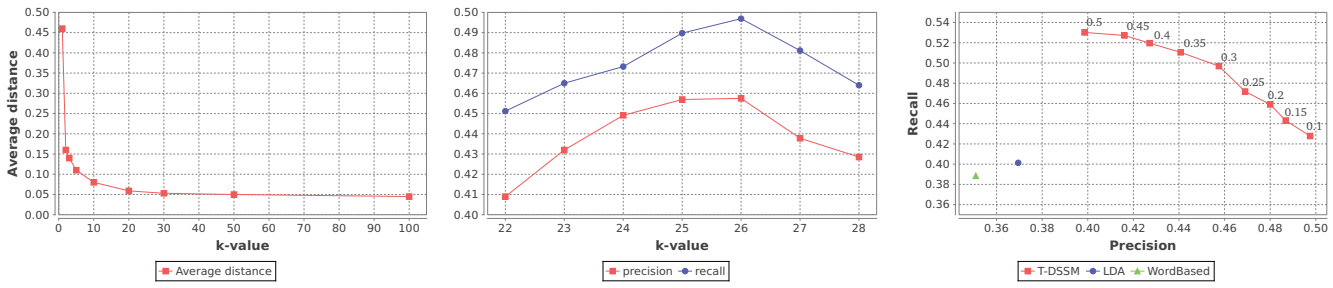
For the pass rate α which is used to recognize highlight slides, we evaluate the $Precision$ and $Recall$ of our model with different α from 0.1 to 0.5, the result of which is shown in Figure. 3(c). We have discussed in Section 4 that α decides the number and the length of highlight slides, hence, it would influence the performance of our model to some extent. From Figure. 3(c), we can see that the $Precision$ decreases and the $Recall$ increases as α grows from 0.1 to 0.5 with a step of 0.05. With relatively small α, our framework can generate more highlight slides, which could improve the $Recall$ and decline the $Precision$ at the same time. We observe that when α is set as 0.3, our model can achieve the best performance in $F1$ score. Moreover, no matter what value we give to α within this interval (i.e. $[0.1, 0.5]$), our T-DSSM based framework is always superior to the other baseline models.

To well capture the parameters for other applications, we have some experience that can reduce the efforts for this. For instance, we can define a threshold to judge whether average distance has been converged, based on which the number of clusters k can be determined easily. Moreover, α can be initialized with 0.3, and it is better to subsample the data and obtain a small groups of data for further tuning.

Discussion

Here, we will discuss two phenomenons found in our result.

First, we have observed that video shots labeled with "magnificent fighting" may have two kinds of comments: those related to "Shana", "wu ru sai" from anime "Shakugan No Shana" and thoes about "Kanade" or "hand sonic" from anime "Angel Beats". In fact, this phenomenon shows



(a) (b) (c)

Figure 3: The influence of parameter k and α. (a) Average Distance w.r.t. k, (b) Precision and Recall w.r.t. k, (c) Precision and Recall w.r.t. α

us that "magnificent fighting" could be further divided into "Shana battle scene" and "angel battle scene", and if so, a user may get more accuracy result when he aim to search one of them. So we notice that available labels in our framework will not be limited within the 10 types about scene, character and music in our experiment. They can be extended to more special use like recognizing some identical people or scenario in a video.

Moreover, we have observed that most of labels in "moving" type stops earlier than the ground truth. To explain this, we should know that "moving" is always relevant to shots with heavy mind and most users prefer to send "QWQ" or "tears eyes" only at the beginning of it. Due to the impressiveness of the shot, viewers may be totally absorbed in the video and have no spare to send comments later. Hence, there will be fewer comments appear near to the end of the shot which is difficult to be recognized as "moving". We observed from the cases that although there are slight but regular errors due to the user habit for different topics, they can be well interpreted.

6 Related Work

Researches on bullet-screen comments are comparatively rare since it is a new interactive mode around video-sharing. Most of the works focused on some statistics for bullet-screen comments and the correlation between comments and videos. (Lin, Ito, and Hirokawa 2014) proposed a statistical method to identify whether a Chinese word is a loanword from Japanese or not based on bullet-screen comments. (Wu and Ito 2014) investigated correlation between emotional comments and popularity of a video. A Temporal and Personalized Topic Model (TPTM) proposed by (Wu et al. 2014) generates time-sync video tags for videos using comments. However, there are few works about temporal labeling and all their works are based on words rather than the understanding in semantics, so it will be difficult in coping with the informal expression of comments.

Another work related to ours is word/semantic embedding. Due to the success of deep learning in image and speech recognition area (Hinton, Osindero, and Teh 2006; Hinton et al. 2012; Deng et al. 2013), performing semantic embedding by deep structured models has turned out to be a very successful approach. Word/Semantic embedding technique were used in natural language processing (NLP) to simplify and improve performance (Turian, Ratinov, and Bengio 2010; Collobert et al. 2011; Socher et al. 2013) where words or phrases from the vocabulary are represented as vectors of real numbers in a low dimensional space. A deep structured semantic model (DSSM) was developed in which raw term vector of a query or document is mapped into a common low-dimensional semantic vector through a feed-forward neural network (Huang et al. 2013). However, most of existing methods are suitable for documents with strict syntax which the bullet-screen comments do not have. Based on DSSM, our proposed model is designed to analyze such short text with temporal-correlation.

There are also some works around video highlight extraction, in which segments with highlight are extracted from sport videos (Hanjalic 2005; Kathirvel, Manikandan, and Soman 2011), e.g., baseball (Hu et al. 2011) or soccer (Wang et al. 2014) games. No doubt that their applications are limited within sports videos since their methods are based on identical scene in visual, and they also cannot tell the type of the highlights.

7 Conclusions and Future Work

In this paper, we proposed a novel video understanding framework to assign temporal labels on highlighted video shots. To deal with the informal expression of bullet-screen comments, T-DSSM was designed to represent comments into semantic vectors with taking advantage of their temporal correlation. Then, video highlight shots were recognized and finally temporally labeled via mapping semantic vectors in a supervised way. Experiments on real-world dataset proved that our framework could effectively label video highlights with a significant improvement compared with several baselines, which clearly validates the potential of our framework on video understanding, as well as bullet-screen comments interpretation.

In the future, we will focus on the following two potential directions along this line. First, we will further adapt the raw comment feature as the input of T-DSSM to make it suitable for all languages. Second, we will extend our framework in multi-labeling the highlights.

8 Acknowledgements

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Natural Science Foundation of China (Grant No. 61403358) and the Fundamental Research Funds for the Central Universities of China (Grant No. WK2350000001). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS and acknowledges the support of the CCF-Intel Young Faculty Researcher Program (YFRP). Finally, we are grateful for Hengshu Zhu's help during our discussion.

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3:993–1022.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)* 1–38.

Deng, L.; Li, J.; Huang, J.-T.; Yao, K.; Yu, D.; Seide, F.; Seltzer, M.; Zweig, G.; He, X.; Williams, J.; et al. 2013. Recent advances in deep learning for speech research at microsoft. In *ICASSP*, 8604–8608. IEEE.

Hanjalic, A. 2005. Adaptive extraction of highlights from a sport video based on excitement modeling. *Multimedia, IEEE Transactions on* 7(6):1114–1122.

Hecht-Nielsen, R. 1989. Theory of the backpropagation neural network. In *IJCNN*, 593–605. IEEE.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE* 29(6):82–97.

Hinton, G.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7):1527–1554.

Hu, M.-C.; Chang, M.-H.; Wu, J.-L.; and Chi, L. 2011. Robust camera calibration and player tracking in broadcast basketball video. *Multimedia, IEEE Transactions on* 13(2):266–279.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2333–2338. ACM.

Kathirvel, P.; Manikandan, M. S.; and Soman, K. 2011. Automated referee whistle sound detection for extraction of highlights from sports video. *International Journal of Computer Applications* 12(11):16–21.

Lin, X.; Ito, E.; and Hirokawa, S. 2014. Chinese tag analysis for foreign movie contents. In *ICIS*, 163–166. IEEE.

Orr, G. B., and Müller, K.-R. 2003. *Neural networks: tricks of the trade*. Springer.

Socher, R.; Bauer, J.; Manning, C. D.; and Ng, A. Y. 2013. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, 384–394. Association for Computational Linguistics.

Wang, Z.; Yu, J.; He, Y.; and Guan, T. 2014. Affection arousal based highlight extraction for soccer video. *Multimedia Tools and Applications* 73(1):519–546.

Wu, Z., and Ito, E. 2014. Correlation analysis between user's emotional comments and popularity measures. In *IIAIAAI*, 280–283. IEEE.

Wu, B.; Zhong, E.; Tan, B.; Horner, A.; and Yang, Q. 2014. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *SIGKDD*, 721–730. ACM.