CROWDSOURCED TIME-SYNC VIDEO TAGGING USING SEMANTIC ASSOCIATION GRAPH

Anonymous ICME submission

ABSTRACT

Time-sync comments reveal a new way for extracting the online video tags. However, such time-sync comments have lots of noises due to users' diverse comments, introducing great challenges for accurate and fast video tag extractions. In this paper, we propose a novel unsupervised video tag extraction algorithm named Semantic Weight-Inverse Document Frequency (SW-IDF). SW-IDF first generates corresponding semantic association graph (SAG) using semantic similarities and timestamps of the time-sync comments. Then it clusters the comments into sub-graphs of different topics and assigns weight to each comment based on SAG, which can clearly differentiate the meaningful comments with the noises. In this way, the noises can be identified, and effectively eliminated. Extensive experiments have shown that SW-IDF can achieve 0.3045 precision and 0.6530 recall in high-density comments; 0.3800 precision and 0.4460 recall in low-density comments, which is the best performance among the existing unsupervised algorithms.

Index Terms— video tagging, crowdsourced time-sync comments, semantic association graph, keywords extraction

1. INTRODUCTION

Recently, many people spend their time on on-line video sites for news and entertainment. The wide applications of online (or live) videos raise great challenges in fast and accurate videos searching techniques. There are many efforts made on automatic video tagging techniques based on text retrieval [1, 2], but they can only provide video-level tags [3]. In other words, the description of the video tags are not specific enough and timestamps are not matched with tags. Fortunately, a new type of comments, i.e., time-sync video comments appear on video websites like Youku (www.youku.com), AcFun (www.acfun.tv) and BiliBili (www.bilibili.com) in China, and NicoNico (www.nicovideo.jp) in Japan.

In this paper, we focus on the problem of extracting video tags from time-sync comments. Time-sync video comment is an interactive comment form, where users can make their comments synchronized to a video's playback time. These comments are saved and will be displayed on the video screen for other users in the future. Also,they are crowdsourced data, which conveys information involving the content of the current video, feelings of users or replies to other comments. Therefore, time-sync video comments mining can be served as a new way for video tag extraction.

As far as we know, Wu *et al.* [3] are the only ones to extract video tags from time-sync comments, which inspire our work. They use statistics and topic model to build Temporal and Personalized Topic Modeling (TPTM). However, their method need the use id of each comment, which is difficult to obtain accurately because of the privacy-protecting policy. What is more, their approach does not consider the semantic of comments, leading some noises of comments which are not related to the video involved.

To tackle these problems, we propose a graph-based algorithm named Semantic Weight-Inverse Document Frequency (SW-IDF) to generate time-sync video tags automatically. The time-sync video comments have some features distinguished from the common short text, so the main idea in our algorithm is based on the these features. To specify, time-sync video comments have the following features: *(1) Semantic relevance.* Abundant video semantic information is contained that describes both local and global video contents by selecting the time interval of the timestamp. *(2) Real-time.* Time-sync video comment is synchronous to the real-time content of the videos. Users may produce different topics when it comes to the same video contents. *(3) Interdependence.* Latter comments usually depend on the former ones, which means the latter comments have a semantic association with the foregoing. *(4) Noise.* Plenty of noises and internet slang are involved in comments, which makes trouble for tag extraction. Therefore, how to distinguish the weight of each comment and thus identify high-impact comments and noise is a major challenge.

According to our observed results, the noises have few semantic relationships with other time-sync comments while some high impact comments have many in a period of time. Based on this, we can reduce the impact of noises by clustering the semantic similar and time-related comments, and identify high impact comments by their semantic relationships. Specifically, we first generate corresponding semantic association graph (SAG) using semantic similarities and timestamps of the time-sync comments. Then we treat the time-sync comments as vertices in the graph, cluster them into different topics by community detection theory [4], and assign weight to each comment based on the degrees of comment in SAG, which can clearly differentiate the meaningful comments with noises. Moreover, we gain the weight of each word by combining semantic weight and inverse document frequency which is similar to TF-IDF algorithm, and then video tags are extracted in an automatic way.

The main contributions of our paper are as follows:

1) We propose a novel graph-based Semantic Weight-Inverse Document Frequency (SW-IDF) algorithm, which can extract both local and global video tags in an unsupervised way by mining time-sync comments.

2) We build Semantic Association Graph (SAG) to cluster the comments into sub-graphs of different topics. The method takes the features of time-sync comments into account, and effectively reduce the impact of noises.

3) We evaluate our proposed algorithm with real-world datasets on mainstream video-sharing websites and compare it with classical keyword extraction methods. The results show that SW-IDF outperforms baselines in both precision and recall of video tag extraction.

2. RELATED WORK

Time-sync video comment has an increasing number of researches emphasize text mining on it. In addition to Wu *et al.*'s work [3], there are also some other applications based on time-sync comments. Xian *et al.* extract highlight of video clips by mining the time-sync comments [5] in a simplified model. Lv *et al.* propose a T-DSSM [6] to represent comments into semantic vectors, and video highlights are recognized by semantic vectors in a supervised way. Wu and Ito [7] investigate the correlation between emotional comments and popularity of videos and He *et al.* [8] propose a model to predict the popularity of videos.

Tag/keyword extraction is another work related to ours. At present, three unsupervised keyword extraction methods are available. The first one is based on word frequency statistics, such as TF-IDF. The second method depends on the relationship between the sentences, such as textrank [9], which a graph-based ranking model. And the last one is according to topic model. It brings document-topic and topic-word distribution together by simulating document generation process. Blei *et al.* propose the Latent Dirichlet Allocation(LDA) model [10], the most representative model. To better deal with short text situation, Yan *et al.* propose the BTM [11], which models the generation of word co-occurrence patterns (i.e. biterms) in the whole corpus directly. Yin and Wang propose the Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model [12, 13] for short text clustering and keyword extraction. However, above methods cannot deal with the texts that are full of noises.

Semantic similarity of time-sync comments is a critical issue in our paper. There are mainly two kinds of approaches to measure the similarity of documents. One is based the similarity of the words in sentences. The representations of this approach are propose by Li *et at.* [14] on unsupervised learning and Socher *et at.* [15, 16] on supervised learning. Considering that time-sync comments contain mass newborn internet slang, it is hard to obtain accurate results in this way. The other one is based on the sentence vector. Texts are converted to vector as first, and distance of the vectors is calculated as the similarity between sentences.The topic model such as LDA, and embedding model such as word2vec [17] are the representations of this method.

3. OUR ALGORITHM

In this section, we will first introduce how to construct Semantic Association Graph (SAG) for time-sync comments with their semantic similarity. Then we cluster the comments into subgraphs of different topics, generating the weight of each comment by an out-in degree iterative algorithm based on SAG. Finally, keywords are extracted as video tags from time-sync comments automatically.

3.1. Preliminaries

In this paper, we choose word2vec to calculate the semantic similarity between each pair of comments. To simplify the algorithm, we calculate the mean vector of each word in the sentence as the sentence vector and dimensionality of a vector is 300. Therefore, the semantic similarity between comment a and b is calculated by the cosine angle between vectors:

$$Sim(a,b) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{1}$$

Besides, in our approach, each comment i has a timestamp t_i, denoting the post time of the time-sync comment. The greater the timestamp interval between two comments, the less likely they are in the same topic. So we use the exponential function to express the decay of comment associations:

$$Delay(a,b) = exp^{-\gamma_t \cdot (t_b - t_a)} \tag{2}$$

where γ_t is the attenuation coefficient.

3.2. Semantic Association Graph Construction

Noises in time-sync comments have less semantic associations with the other comments. To decrease the influence of noises, we put the comments with high semantic similarity and close timestamp into a topic and provide a higher weight to topics which contain more comments.

We use a graph to describe the relationships between comments and construct the semantic association graph (SAG) where the vertices are comments and the edges reflect their pairwise semantic. Since comments appear in chronological order, they can only affect the upcoming comments rather

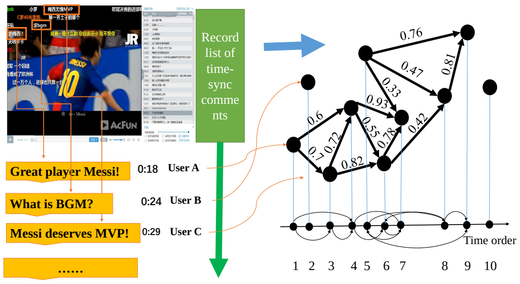

Fig. 1. An example of SAG Construction

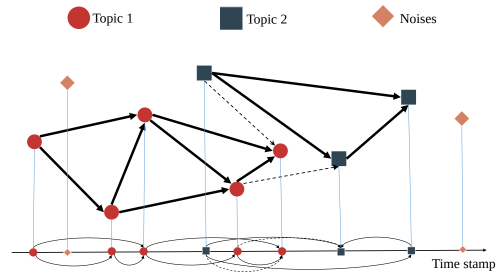

Fig. 2. An example of Topic Partitioning

than prior comments. Thats why we choose directed graph to represent it. Let G denote the directed graph. It is represented by $G = (V, E)$, where V and E are the sets of nodes and edges. Specifically, $V = \{v_1, v_2, ..., v_N\}$, $E = \{e_1, e_2, ..., e_M\}$, where N is the number of nodes in V, and M is the number of edges in E. In our algorithm, $t_{v_1} < t_{v_2} < ... < t_{v_N}$. We use domain to describe the detail of edges. For each edge e_k, $e_k.x$ and $e_k.y$ are two vertices that linked by edge e_k where $t_{e_k.x} < t_{e_k.y}$, and $e_k.w$ is the weight of edge e_k. The weight of edge i that link vertices u and v is defined by

$$e_{i.w} = \begin{cases} Sim(u,v) \cdot Delay(u,v) & t_u < t_v \\ 0 & t_u > t_v \end{cases}$$

Besides, $e_{u,v}$ also expresses the edge that link vertices u and v where $t_u < t_v$. Through experiment, it is proper to consider comment u and v has no association when $e_{u,v}.w < 0.3$, and then we can set $e_{u,v}.w = 0$. Each comment in our model has exact one topic. For vertex v_i, $v_i.S$ is used to describe the set that contain the vertices which have the same topic as v_i and $|S|$ is used to express the number of vertices in set S. An example of SAG Construction is shown in Fig. 1. In a UEFA Champions League video, user A made the comment 1 as"Great player Messi!" when he saw the goal at time $0:18$. Then user B responded with "Messi deserves MVP!" as the comment 3 at time $0:24$. User C makes a comment "What is the BGM ?" as comment 2 to ask the background music, which deviates the video content. So the comment made by user C has no semantic association with other comments, while comments proposed by user A and B has a semantic edge.

3.3. Topic Partitioning

In our algorithm, the time-sync comment that has the similar semantics and similar timestamps should belong to the same topic, so that the mean weight of edges in intra-topic is large while the mean weight of edges that link different topics is small, which satisfies community detection theory.

In the beginning, each comment belong to a unique set that only contains itself to achieve the objective. That is, for each v_i, $v_i.S = \{i\}$. Then edges in set E are sorted by descending order of weight. The new edge set $E' = \{e'_1, e'_2, ..., e'_k, ..., e'_M\}$ are obtained, where $e'_1.w > e'_2.w > ... > e'_M.w$. We process each edge from e'_1 to e'_M. For each edge e'_k, $e'_k.x.S$ and $e'_k.y.S$ are set as S_1 and S_2. We merge S_1 and S_2 when $S_1 \neq S_2$ and

$$\frac{\sum\limits_{e_p.x,e_p.y \in S_1} e_p.w + \sum\limits_{e_q.x,e_q.y \in S_2} e_q.w + \sum\limits_{i \in S_1} \sum\limits_{j \in S_2} e_{i,j}.w}{(|S_1| + |S_2|) \cdot (|S_1| + |S_2| - 1)/2} > \rho$$

where ρ is the threshold of intra-cluster density. In this paper, disjoint-set (union-find set) algorithm [18] is used to merger the sets efficiently. When all the edges are solved, comments with high semantic similarity are merged into a topic, and the intra-cluster density of each subgraph is higher than the threshold. An example is shown in Fig. 2. The SAG constructed in Fig. 1 is partitioned into two topics and several noises. The comment "Great player Messi!" and "Messi deserves MVP!" belong to the same topic, while the comment "What is the BGM ?" is identified as a noise.

3.4. Weight Distribution and Tag Extraction

In this section, we attribute weight to each comment according to the influence of its topic and the relationship in the semantic graph.

The weight of a comment is affected by its topic popularity, so we define the popularity of the comment i,

$$P_i = \frac{|v_i.S|}{\sqrt[K]{|S_1| \cdot |S_2| ... |S_K|}} \tag{3}$$

where S_i is the $i-th$ topic in semantic association graph, and K is the total number of topics in semantic association graph. Obviously that those topics with fewer comments are more likely to be noises and should have less weight. According to the formula above, we can reduce the influence of noises.

Within the topic, a comment affecting more comments and affected by less comments should have higher weight. A proposed graph iterative algorithm is designed to gain the comment weight within the topic. An influence matrix

Algorithm 1 EXTRACTING TAGS BY SW-IDF

Input Semantic Association Graph
Output Tags of video

sort E by descending order of $e_i.w$, obtain $E^{'}$
for i = 1 to M **do**
  set $e_i^{'}.x.S$ as S_1, $e_i^{'}.y.S$ as S_2
  if $\frac{\sum\limits_{e_p.x,e_p.y\in S_1} e_p.w+\sum\limits_{e_q.x,e_q.y\in S_2} e_q.w+\sum\limits_{i\in S_1}\sum\limits_{j\in S_2} e_{i,j}.w}{(|S_1|+|S_2|)\cdot(|S_1|+|S_2|-1)/2} > \rho$ **then**
    merge S_1 and S_2
  end if
end for
Calculate the influence matrix $\mathbb{M}_{N\times N}$ by Eq.(4)
for i = 1 to N **do**
  $I_i^0 = 1$
  Calculate the popularity of comment i using Eq.(3)
end for
for k = 1 to T **do**
  for i = N downto 1 **do**
    Calculate I_i^{2k-1} using Eq.(5)
  end for
  for i = 1 to N **do**
    Calculate I_i^{2k} using Eq.(6)
  end for
end for
Calculate the SW-IDF of each word using Eq.(7)
Select maximum SW-IDF words as video tags

$\mathbb{M}_{N\times N}$ is established at first. For the elements in the matrix,

$$m_{i,j} = \begin{cases} e_{i,j}.w & v_i.S = v_j.S \\ 0 & v_i.S \neq v_j.S \end{cases} \tag{4}$$

We use $I_{i,k}$ to express the influence value of $i-th$ comment after k iterations. For each comment i, $I_{i,0} = 1$ in the initial. Then on the $k-th$ turn of iteration, there two steps as follows:

$$I_{i.2k-1} = I_{i,2k-2} + \sum_{j=i+1}^{n} m_{i,j} \cdot I_{j,2k-1} \tag{5}$$

$$I_{i,2k} = \frac{I_{i,2k-1}}{I_{i,2k-1} + \sum\limits_{j=1}^{i-1} m_{j,i} \cdot I_{j,2k}} \tag{6}$$

In the $(2k-1)-th$ iteration, we increase the influence value of comment i based on the influential value of its affecting comments. We know that a comment will only affect the comments lagging behind it, so the comments can be processed from v_N down to v_1. That is, before we process comment i, all the comments j which $t_j > t_i$ have been processed. In the $(2k)-th$ iteration, we reduce the influence value of comment i based on the influence value of the comments which affect it. Contrary to the $(2k-1)-th$ iteration, we process the comments from v_1 to v_N in the $(2k)-th$ iteration. The influential value of the comments in Fig. 1 is shown in Fig. 3. After 20 turns of iterations, all comments converge to the interval $[0,1]$. In topic 1, the influence value of comment 3 and comment 4 is bigger than comment 6 and comment 7, meting our expectation.

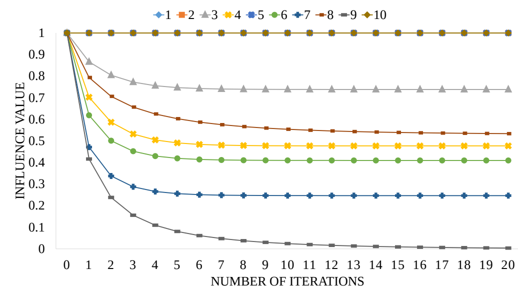

Fig. 3. The influence value of comments in Fig. 1

To combine the popularity and the influence value, the comment weight i is obtained by $W_i = P_i \cdot I_i^T$ where T is the number of turns of iterations and depends on the number of nonzero elements in matrix $\mathbb{M}_{N\times N}$. So, weight of each word is formulated as below:

$$SW-IDF_i = \sum_j W_j \cdot IDF_i \tag{7}$$

where j expresses the comment that word i appears and IDF_i is the inverse document frequency as defined in TF-IDF method. We extract words with the highest weight as video tags. The complete algorithm in shown in Algorithm 1.

4. EXPERIMENTAL STUDY

In this section, we verify the effectiveness of our proposed method by comparing with four unsupervised methods of keyword extraction. The datasets are collected from AcFun. We provide necessary parameters for our model at first and then analyze the performance of our method on video tag extraction.

4.1. Experimental Setup and Datasets

We crawl time-sync comments from a Chinese time-sync comments video website AcFun. To specify, totally 227,780 comments are collected randomly from music, sports, and movie, 126,146 comments for the training set and 101,634 comments for the test set. The test set is divided into two parts: high-density comments and low-density comments by density (the number of comments per second). We manually make video tags about 120 videos as the standard. More details included Length (second), Number of comments, Density (comments per second) and the number of videos about test set is shown in the Table 1.

Table 1. Data Description Table

	Length	Comments number	Density	Video number
High density	7475	41,556	5.5593	89
Low density	79008	60,078	0.7604	31

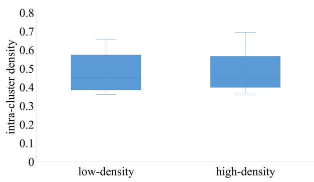

Fig. 4. Box diagram of intra-cluster density

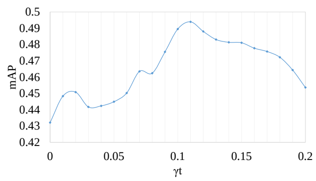

Fig. 5. The effect of attenuation coefficient γ_t

In our algorithm, two parameters need to be determined, the threshold of intra-cluster density ρ, and the attenuation coefficient γ_t. The ρ controls the accuracy of topic clustering. Distances between different topics increase with increasing of ρ. To determine ρ, we randomly choose some comments from both low-density and high-density comments in the test set. Then we manually partition the comments that should belong to the same topic and calculate the intra-cluster density of them. The box diagram of them is show in Fig. 4. The minimum value of intra-cluster density in both low-density and high-density is nearly 0.36, so ρ is initiated to 0.36 in our algorithm. The γ_t is the attenuation coefficient of the interval between time-sync comments, which controls the value of the edge weights in the graph. To determine γ_t, we randomly choose 30 tagged videos from the test set, adjusting γ_t and observe the mAP of video tags generated by our model. Results are shown in the Fig. 5. γ_t gains better performance in the range of 0.1 to 0.15 and gets optimal performance at 0.11. In fact, when $\gamma_t = 0$, the semantic association graph is independent of time; when $a = +\infty$, all weights of edge equal to 0, and our model is equivalent to TF-IDF. Besides, the number of iterations T also need to be determined. We find in the experiment that graph with hundreds of vertices is convergence after nearly 20 interactions and with thousand vertices is convergence after approximately 50 interactions.

4.2. Results

To evaluate the performance of the proposed video tag extraction algorithm, we compare our method with 4 unsupervised keyword extraction methods in baseline including:

(1) A classical keyword extraction algorithm TF-IDF.

(2) A graph-based text ranking model, textrank [9], which is inspired by PageRank. The method is denoted as "TX".

(3) A topic model based algorhtim, Biterm Topic Model [11], which is the improvement of LDA [10] for shot texts. The method is denoted as "BTM".

Table 2. Comparison of different methods on video tag extraction of the top 10 candidate tags with high density comments.

Method	Prec	Recall	F-score	mAP
TF-IDF	0.2674	0.5735	0.3648	0.4224
TX	0.2427	0.5205	0.3310	0.3696
BTM	0.2337	0.5012	0.3188	0.3094
GSDPMM	0.2445	0.5094	0.3302	0.3374
SW-IDF	0.3045	0.6530	0.4153	0.4853

Table 3. Comparison of different methods on video tag extraction of the top 10 candidate tags with low density comments.

Method	Prec	Recall	F-score	mAP
TF-IDF	0.3411	0.4028	0.3694	0.3098
TX	0.3224	0.3709	0.3450	0.3147
BTM	0.3210	0.3662	0.3369	0.2927
GSDPMM	0.3440	0.4038	0.3715	0.3202
SW-IDF	0.3800	0.4460	0.4104	0.3518

(4) A collapsed Gibbs Sampling algorithm for the Dirichlet Process Multinomial Mixture model [12, 13], which has good performance when dealing with shot texts. This method is denoted as "GSDPMM"

Regrettably, we cannot choose Wu et al.'s [3] as baseline, because both video sites AcFun and Bilibili have already hidden the ID of users when posting comments to protect privacy, which is necessary in Wu et al.'s method. To increase the reduce the effect of noises, we build a stop word list, and delete all stop words in the original comments.

For each method, we calculate the precision, recall, mAP(mean average precision) and F-score of top 10 tagging results. The result of high density and low density of comments are shown in Table 2 and Table 3 respectively.

The results show that our method performs better than baselines for comments both in high-density and low-density conditions. In high-density condition, both precision and recall are significantly increased because the semantic graph is dense. Meanwhile, noises of comments also increase with the increasing of the density of comments, so the result of topic model based methods, BTM and GSDPMM are poor performance and even worse than classical method TF-IDF. Relatively, in low-density comments, the graph is sparse and noises reduce, so the increase rate of precision and recall of our model reduce slightly, but is still the highest among the others.

To further validate our model, we show the precision and recall of top 5 and top 15 candidate tags in Table 4. The results of each method are similar to the performance of Top 10, which prove that our model has the best performance when extracting video tags from time-sync comments in any situation, compared with the most of existed keyword extraction algorithms.

Table 4. Comparison of different methods on video tag extraction of the top 5 and top 15 candidate tags

Method	H-Top 5		H-Top 15		L-Top 5		L-top 15	
	Prec	Recall	Prec	Recall	Prec	Recall	Prec	Recall
TF-IDF	0.418	0.448	0.187	0.600	0.414	0.243	0.299	0.526
TX	0.301	0.323	0.181	0.583	0.384	0.225	0.281	0.507
BTM	0.272	0.292	0.177	0.569	0.368	0.216	0.261	0.460
GSDPMM	0.281	0.301	0.183	0.593	0.418	0.249	0.307	0.539
SW-IDF	0.488	0.523	0.223	0.718	0.464	0.272	0.349	0.615

5. CONCLUSION

In this paper, we present a graph-based algorithm named Semantic Weight-Inverse Document Frequency (SW-IDF) to extract video tags from time-sync comments. Compared with the previous work, we are pioneers to use Semantic Association Graph (SAG) to build relationship among comments according to the features of time-sync comments. Based on the properties of Semantic Association Graph (SAG), SW-IDF divides comments into subgraphs in different topics and assigns weight to each comment. Through the approaches above, noises can be effectively identified and eliminated. Then keywords are extracted as video tags by the weight of each comment. Experiments with real-world datasets and four unsupervised baseline methods prove the correctness of our model and show that tags extracted by SW-IDF have the best precision and recall rates.

6. REFERENCES

[1] Stefan Siersdorfer, Jose San Pedro, and Mark Sanderson, "Automatic video tagging using content redundancy," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2009, pp. 395–402.

[2] Adrian Ulges, Christian Schulze, Markus Koch, and Thomas M Breuel, "Learning automatic concept detectors from online video," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 429–438, 2010.

[3] Bin Wu, Erheng Zhong, Ben Tan, Andrew Horner, and Qiang Yang, "Crowdsourced time-sync video tagging using temporal and personalized topic modeling," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 721–730.

[4] Santo Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

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

[6] Guangyi Lv, Tong Xu, Enhong Chen, Qi Liu, and Yi Zheng, "Reading the videos: Temporal labeling for crowdsourced time-sync videos based on semantic embedding," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[7] Zechen Wu and Eisuke Ito, "Correlation analysis between user's emotional comments and popularity measures," in *Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on*. IEEE, 2014, pp. 280–283.

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

[9] Rada Mihalcea and Paul Tarau, "Textrank: Bringing order into texts," Association for Computational Linguistics, 2004.

[10] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[11] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 1445–1456.

[12] Jianhua Yin and Jianyong Wang, "A dirichlet multinomial mixture model-based approach for short text clustering," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 233–242.

[13] Jianhua Yin and Jianyong Wang, "A model-based approach for text clustering with outlier detection," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 625–636.

[14] Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1138–1150, 2006.

[15] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 129–136.

[16] Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger, "From word embeddings to document distances," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015, pp. 957–966.

[17] Tomas Mikolov, Quoc V Le, and Ilya Sutskever, "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

[18] Robert Endre Tarjan, "Efficiency of a good but not linear set union algorithm," *Journal of the ACM (JACM)*, vol. 22, no. 2, pp. 215–225, 1975.