Journal of Information Hiding and Multimedia Signal Processing ©2009 ISSN 2073-4212
Ubiquitous International Volume **12**, Number **3**, September **2021**

Clustering Web Videos to Improve User Experience on Website

Phuc Nguyen[1,2]

[1]Faculty of Information Systems, University of Economics and Law, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
phucnq@uel.edu.vn

Received March 2021; revised June 2021

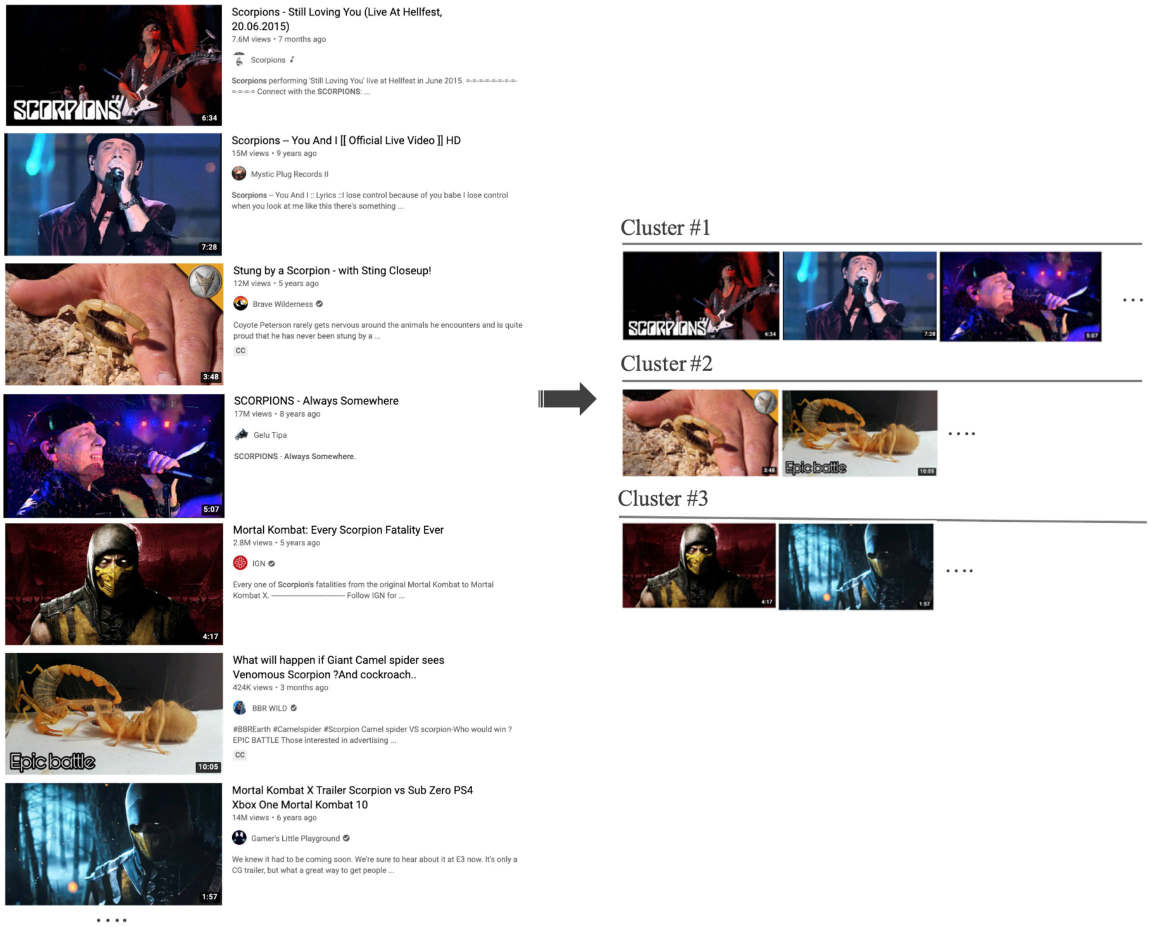
Abstract. *In this paper, we propose a solution to cluster video search results by each topic. We hope this helps to improve user experience on video search engines such as: YouTube, Google videos, etc. The returned search results of these systems are presented as a flat list with many videos of different categories mixed together, and as a result, users find it difficult to locate video clips of interest. Therefore, clustering web video search results is necessary in order to help users quickly find videos on demand. This paper aims to develop our previous researches on clustering web video search results which reported in* [9, 10]. *The main idea based on analyzing and combining the features extracted from video to evaluate the role of each feature and find the set of appropriate features to improve the clustering quality of web video search results*

Keywords: Multiple features, Social web video clustering, User experience, Video representation.

1. **Introduction.** Video is one of the most popular digital content types shared on the Internet. To search videos, users usually use online video search systems such as YouTube, Google videos. In these search engines, the search results are presented as a flat list with videos ranked by their relevance to the query keywords. Given the list, users have to greedily scan over pages (i.e. portions of the list displayed as pages on the site) to locate their videos of interest. When the user submits a short or vague query, the search results are fragmented by irrelevant videos. This makes it difficult for users to determine the desired videos. Clustering video search results is a method to fix this problem. This method gives users a better overview through clustered specific video topics, users can easily ignore inappropriate video clusters and identify videos to look for in a short time instead of browsing the entire list of returned search results. Figure 1 depicts the input and output for clustering web video search results.

To cluster videos, one of the major challenges is to compute the similarity between videos. Video is a complex structured data form with many types of features such as visual features, audio features, and associated textual information. The similarity between videos is often calculated based on their representations. A popular approach is to use visual features to represent video. Following this approach, many studies have exploited information from visual features to represent and match videos reported in [11, 13, 14]. However, to be suitable for effectively considering the similarity between videos, each video needs to be represented by combining many features to fully present information.

The effectiveness of the multi-feature combination approach proven in the image clustering problem [2, 6], and the image classification problem [3, 4, 8]. To improve the efficiency of video matching, the problem of clustering video clips is also solved by the multi-feature fusion approach [1, 5]. Hindle et al. [1] focused on the visual features and



Scorpions - Still Loving You (Live At Hellfest, 20.06.2015)
7.6M views • 7 months ago
Scorpions
Scorpions performing 'Still Loving You' live at Hellfest in June 2015. =-=-=-=-=-=-=-=-=-=-= Connect with the SCORPIONS: ...

Scorpions -- You And I [[Official Live Video]] HD
15M views • 9 years ago
Mystic Plug Records II
Scorpions -- You And I :: Lyrics ::I lose control because of you babe I lose control when you look at me like this there's something ...

Stung by a Scorpion - with Sting Closeup!
12M views • 5 years ago
Brave Wilderness
Coyote Peterson rarely gets nervous around the animals he encounters and is quite proud that he has never been stung by a ...
CC

SCORPIONS - Always Somewhere
17M views • 8 years ago
Gelu Tipa
SCORPIONS - Always Somewhere.

Mortal Kombat: Every Scorpion Fatality Ever
2.8M views • 5 years ago
IGN
Every one of Scorpion's fatalities from the original Mortal Kombat to Mortal Kombat X. ——————————— Follow IGN for ...

What will happen if Giant Camel spider sees Venomous Scorpion ?And cockroach..
424K views • 3 months ago
BBR WILD
#BBREarth #Camelspider #Scorpion Camel spider VS scorpion-Who would win ? EPIC BATTLE Those interested in advertising ...
CC

Mortal Kombat X Trailer Scorpion vs Sub Zero PS4 Xbox One Mortal Kombat 10
14M views • 6 years ago
Gamer's Little Playground
We knew it had to be coming soon. We're sure to hear about it at E3 now. It's only a CG trailer, but what a great way to get people ...

FIGURE 1. The video search results are organized by cluster.

textual information of the video. To represent video, the author proposes the Bounded Coordinate System (BCS) model that mainly exploits color information of the video. This model is effective when performing videos with relatively stable colors. For videos of diverse content with different contexts and colors, this model is somewhat limited. For the textual information, the author uses the word-by-word comparison approach, the limitation of this method is to ignore the semantics of the textual information. In [5], the Vector Space Model (VSM) is used to represent textual features. This model heavily relies on the frequency of words to determine the similarity between texts. Usually, the video description text is in short text and described by different users with different words, so the frequency of occurrence of similar words is rare or even absent. Therefore, the VSM model is also not really effective for representing textual information in this case. In [7], the author proposes a video semantic clustering framework based on tag. The similarities between videos are calculated based on the multiple tag statistical correlations.

Through researching previous related studies, we have chosen a multi-feature combination approach to solve the problem of clustering video search results. We focus on exploiting visual features and analysing the semantics of textual information to improve video clustering quality [9, 10]. In this article, we continue to expand the research results from our previous researches. By analysing the characteristics of video feature types, we propose combining visual features, audio features and textual information. We evaluate the role of each feature in influencing video clustering quality to find the right combination of feature types to improve the quality of video clustering.

The rest of the paper is organized as follows. Section 2 introduces the framework overview for video clustering. Section 3 presents our proposed approach in detail. Section 4 shows experimental results with discussions. Finally, we conclude the paper in section 5.

2. **Framework Overview.**

Figure 2 presents a general framework for clustering web video search results.

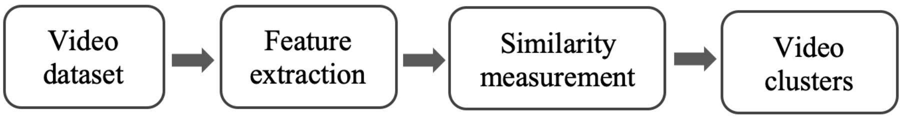


Figure 2. A general framework for clustering web video search results.

The main components of framework as follow:

Videos dataset: Set of videos can be crawled from the video channels online (e.g. YouTube, Google videos).

Feature extraction: The process of extracting features can be extracted from the video such as visual features, audio features and textual information.

Similarity measurement: The similarities between videos are calculated based on their representations and used as input of clustering algorithms. In this work, we focus on extracting the feature and calculating similarity between videos. We do not focus on analysing clustering algorithms because the current clustering algorithms are stable, on the other hand, the quality of video clustering results depends mainly on the similarity between videos based on on their representations.

3. **Proposed Approach.**

3.1. **Proposed model.**

The exploitation of visual features will help group videos that have similar visual representations into a cluster. However, given the variety of video data on the video channels online, videos with similar content (i.e. on the same topic) may contain dissimilar objects and images. Meanwhile, the exploitation of semantic content of the textual information (such as title elements, descriptions, or tags) will help to group videos that have the same semantic content into the same cluster. Visual features and textual information will complement each other to represent video content in a full way to increase the ability to exploit the similarity and the quality of video clustering. However, there is a problem that exploiting textual information is only really effective when it is properly described with the actual content of the video. In fact, the textual information is declared by the user when sharing on online video channels. This textual information may not match the actual content of the video due to various reasons such as subjective user perception, attracting views, etc. In such a context, we believe that the exploitation of combined audio features (e.g. music videos often have sounds such as shouting, applause; racing videos with car engine sounds; etc) will contribute to improving the quality of video clustering.

From the above analysis, we consider combining visual features, audio features, and textual information to solve the problem of clustering video search results. Our framework is shown in Figure 3.

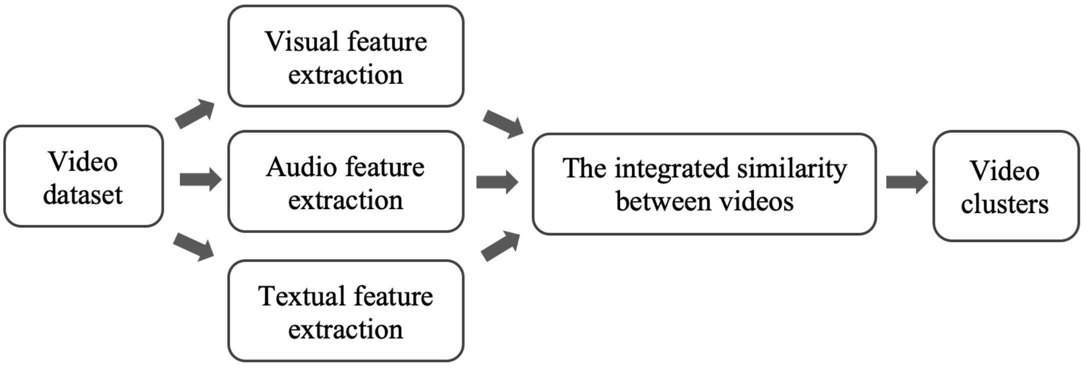

Video dataset
Visual feature extraction
Audio feature extraction
Textual feature extraction
The integrated similarity between videos
Video clusters

Figure 3. The multi-feature integration model solves the problem of clustering web video search results.

3.2. **Representing video and calculating similarity based on visual features.** A video consists of a sequential set of frames. Visual features are extracted directly from each frame and represented as a feature vector. Each video can be represented by a set of feature vectors. Two video clips are compared by matching the frame-by-frame similarity of each video (i.e. each frame in one video must be compared with all the frames in the other video). This method is not effective when the number of frames in the video is large (see Figure 4). Video data on online video channels can be customized and shared by multiple users. That is why the number of frames of the same video can be different. In these cases, if the similarity between videos is calculated based on their similar number of frames, the above method does not completely reflect the similarity between videos.

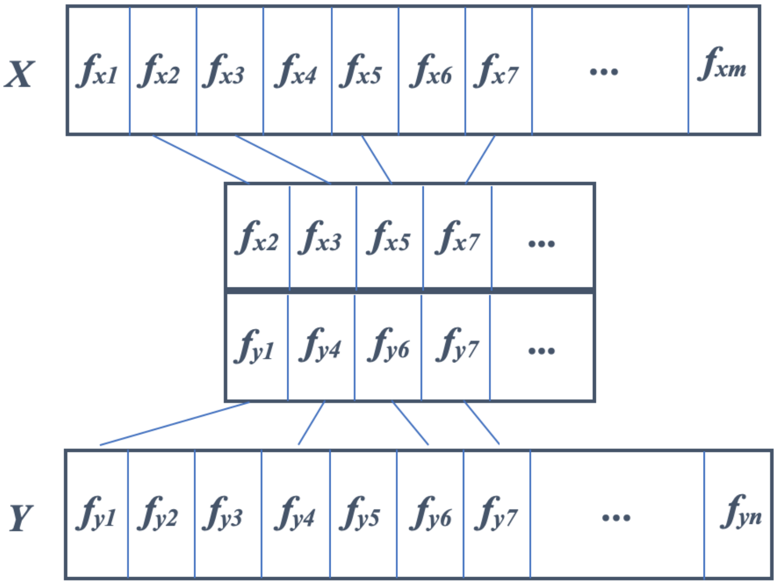

X
f_{x1} f_{x2} f_{x3} f_{x4} f_{x5} f_{x6} f_{x7} ... f_{xm}
f_{x2} f_{x3} f_{x5} f_{x7} ...
f_{y1} f_{y4} f_{y6} f_{y7} ...
Y
f_{y1} f_{y2} f_{y3} f_{y4} f_{y5} f_{y6} f_{y7} ... f_{yn}

Figure 4. The similarity between the two videos is calculated by a frame-by-frame comparison.

In studies such as video indexing, searching for videos, or identifying duplicate videos, the similarity between videos is estimated based on their global representations. In this paper, we use Convolutional Neural Networks (CNN) architecture for representing videos as compact signatures. This approach has been proven effective in our previous work [10]. We use the CNN feature extractor to extract visual features from the selected frames of video. Each frame will be represented by a 4096-dimensional vector. By pooling the feature vectors of the selected frames of video, we get a 4096-dimensional vector as a global video representation. The process of extracting features and representing video is shown in Figure 5.

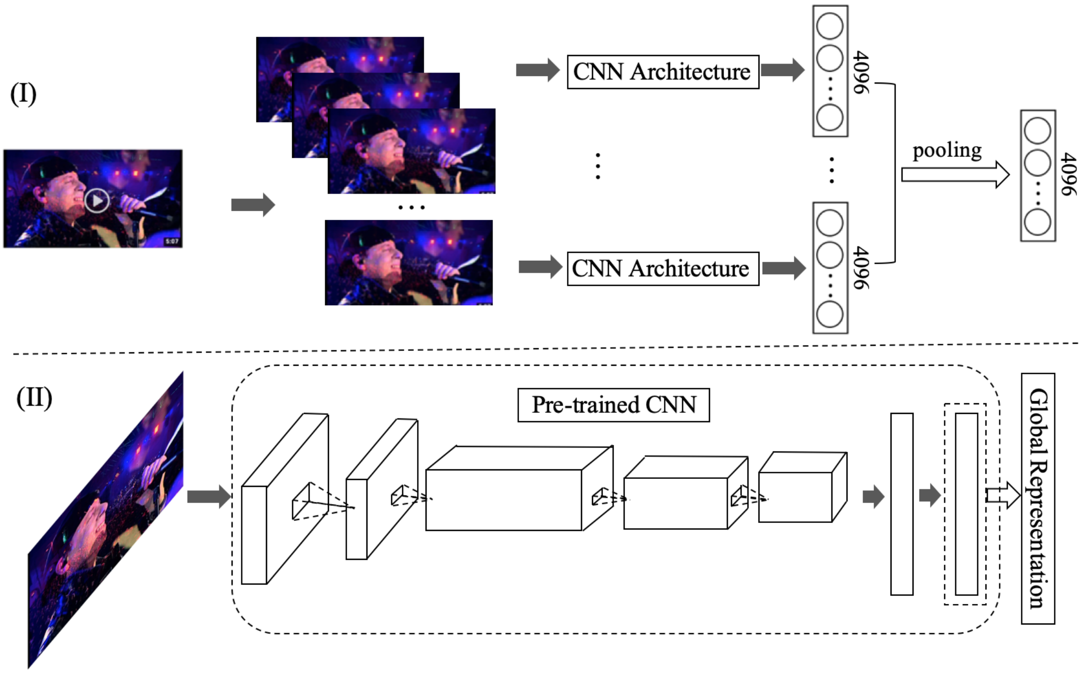


FIGURE 5. The process of extracting visual features and representing video using CNN [10].

The video X is denoted by $V^X = (V_1^X, V_2^X, ..., V_{4096}^X)$. Video Y is also denoted similarly. The similarity between video X and video X is computed as follows:

$$Sim_{visual}(X,Y) = 1 - \frac{\sum_{i=1}^{4096} v_i^X \times v_i^Y}{\sqrt{\sum_{i=1}^{4096} (v_i^X)^2} \times \sqrt{\sum_{i=1}^{4096} (v_i^Y)^2}} \tag{1}$$

3.3. **Representing video and calculating similarity based on audio features.** As analysed above, the audio features play an important role in the representation of video data to improve the ability to exploit similarities between the videos. In this work, we use Mel-frequency cepstral coefficients (MFCC) to represent the audio features extracted from the video [12]. Borrowing the idea from the Bag-of-Words (BoW) model in text data representation, after the audio features are extracted from the videos, these features are grouped into clusters to create a dictionary of audio words. Finally, each video will be represented by a feature vector with a number of dimensions corresponding to the number of words in the dictionary. The similarity between videos is calculated as the distance between the vectors representing them. The process of extracting audio features and representing video using MFCC is illustrated in Figure 6.

3.4. **Calculating similarity based on textual information.** The text information is often described by users in the form of phrases, sentences, or short text. Textual information about the video (i.e. title, description, tags) represents semantic content. Traditional methods of calculating text similarity (e.g. BoW or VSM) mainly focus on analysing shared words (i.e. similarity between words) in texts. These methods are effective when applied to long texts because long texts with similar content often contain

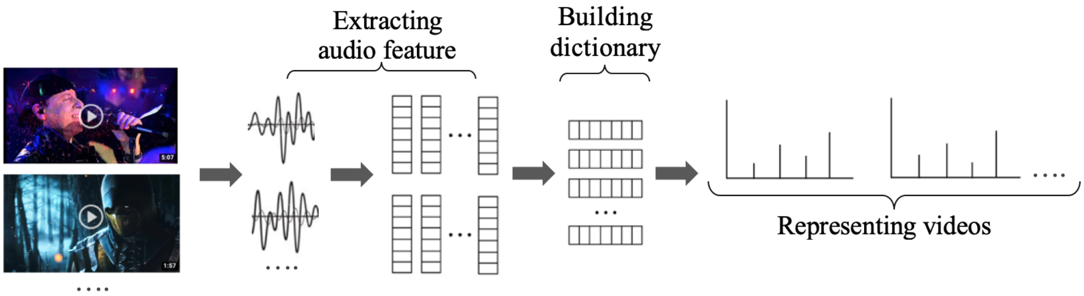


FIGURE 6. The process of extracting audio features and representing video using MFCC.

the same words. However, in short texts, the frequency of the occurrence of similar words is rare. The reason this happened is because the inherent flexibility of natural language allows users to express the same content but with various words.

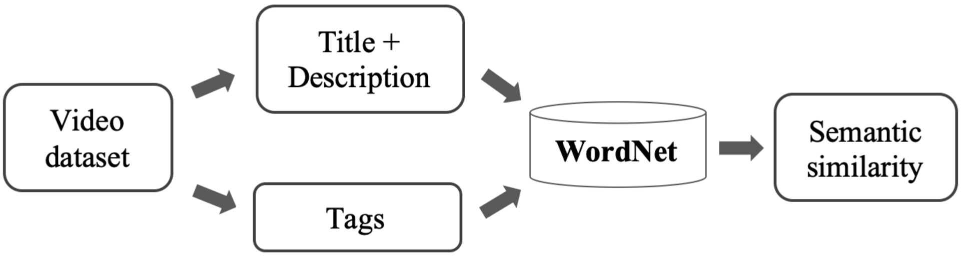


FIGURE 7. The process of calculating the semantic similarity between texts of videos using the WordNet dictionary.

In this work, we inherit the method of using WordNet[1] dictionaries reported in our previous work [9]. The model for calculating the semantic similarity between videos based on text information using the WordNet dictionary is shown in Figure 7.

We combine the title and description in one element and treat them as short texts, we also consider the video's tags as other short texts. The semantic similarity between videos will be estimated based on semantic similarity between short text elements.

3.5. **Calculating similarity based on multi-feature fusion.** Each video is represented with visual, audio, and textual features viewed as a specific object. The similarity between video X and video Y is calculated as follows:

$$Sim(X,Y) = \alpha \times Sim_{vis}(X,Y) + \beta \times Sim_{aud}(X,Y) + (1-\alpha-\beta) \times Sim_{tex}(X,Y) \quad (2)$$

where $Sim(X,Y)$ is the integrated similarity between video X and video Y, $Sim_{vis}(X,Y)$ is the similarity based on visual features, $Sim_{aud}(X,Y)$ is the similarity based on audio features, $Sim_{tex}(X,Y)$ is the similarity based on textual features. For $\alpha, \beta \in (0,1)$ are weights of features. The influence of each feature is expressed by the corresponding weights.

We get a built-in similarity matrix by calculating the similarity for each pair of videos. The integrated similarity matrix is used as the input of the clustering algorithm.

4. Experiments.

[1] http://wordnet.princeton.edu

4.1. **Dataset and evaluation methods.** We use experimental data set from our previous studies [9, 10]. To enrich the experimental topics, we also crawl more video data from YouTube via TubeKit[2]. The data set included 1752 videos of 20 queries with different keywords. Details of the experimental video datasets are described in Table 1.

Table 1. The experimental video dataset

Query	#Videos	#Categories	#Duration (hours)
Apple	80	4	7.5
Aston	82	4	5.3
Cobra	92	5	5.0
Dragon	82	6	5.6
Jaguar	86	4	5.1
Java	87	4	7.2
Jupiter	82	4	5.1
Leopard	95	5	6.4
Lion	89	4	6.2
Lotus	91	6	5.5
Mustang	83	5	5.6
Ocean	90	7	5.5
Panda	97	5	5.8
Pluto	85	7	8.8
Python	85	4	5.1
Scorpion	90	6	6.7
Tiger	81	4	4.3
Venus	89	7	6.9
Viper	87	5	4.5
Zebra	99	7	6.0

We use Entropy and Purity to evaluate the clustering results.

We suppose there is a set of n videos of k topics that are manually labelled, denoted C_j with $j = 1, ..., k$. The clustering algorithm group n videos into k clusters, denoted P_i with $i = 1, ..., k$. The entropy measure is determined as follows:

$$Entropy = -\sum_i \frac{n_i}{n} \times \sum_j \frac{n_{ij}}{n_i} \times log\frac{n_{ij}}{n_i} \tag{3}$$

where n_i is the number of videos in cluster P_i, n_{ij} is the number of videos in cluster P_i that belong to category C_j and n is the total number of videos.

In the ideal case, the entropy value is zero (i.e. each cluster contains only videos that belong to a single topic). In general, the smaller the entropy value, the better the clustering quality.

In contrast to the entropy measure, the purity measure reflects the purity of the clusters. The larger the purity value, the better clustering results. With the symbols have the same meanings as in Equation (3), the purity measure is determined as follows:

$$Purity = -\sum_i \frac{n_i}{n}(max_j \frac{n_{ij}}{n_i}) \tag{4}$$

[2] www.tubekit.org

4.2. **Experimental settings.** For the purpose of comparing and evaluating the effectiveness of the proposed method, we set up the baseline methods reported in [9, 10]. On the other hand, in order to analyse and evaluate the role of each feature and determine the appropriate combination of features to improve the quality of video clustering results, we set up additional experiments combining different sets of features. We set up the experiments according to the following scenario:

Video clustering based on each individual feature. We set up the experiments as follows: video clustering based on visual features called the V method [10], video clustering based on audio features called the A method, video clustering based on textual features called the T method [9].

Video clustering based on a combination of different feature sets with unweighted linear combinations (i.e. the influence of the features is equally evaluated). We set up the experiments as follows: video clustering based on visual and audio features called the VA method, video clustering based on visual and textual features called the VT method [10], video clustering based on audio and textual features called the AT method, video clustering based on visual, audio, and textual features called the VAT method.

Video clustering based on multi-feature integration with weighted combinations using Equation (2). We set up the experiment as follows: video clustering based on visual, audio, and textual features using a corresponding weight for each feature. This method, called the VAT^* method.

The following are details of the weight selection process and clustering algorithm.

The weight selection process. In each specific case, the influence of each type of feature is not the same. With weights $\alpha, \beta \in (0, 1)$ in Equation (2), we run an experiment by changing the weight with a step value of 0.1 to determine the appropriate set of weights. Through experiments, we found that with weights $\alpha = 0.4$ (for visual features), $\beta = 0.5$ (for audio features), 1 - α - $\beta = 0.1$ (for textual features) for better results than the rest of the cases.

The clustering algorithm. There are many popular clustering algorithms such as K-Means, K-Medoids, etc. However, we use the K-Medoids algorithm because the distance between objects only needs to be calculated once. This is consistent with the integrated similarity matrix representing the similarity for each pair of videos. To cluster videos, we run the clustering algorithm with the number of input clusters corresponding to the number of topics of each query.

The key to solve the problem of clustering video search results is to estimate similarity between videos based on their representations. The process of feature extraction is processed offline by the video search engine server at the same time as the video is indexed. The video clustering process is done in real time. This is consistent with a real-world video search system because users expect that the video search results should be returned quickly after they type the query.

4.3. **Experimental results.** The results of the experimental evaluation are shown in Figure 8. The clustering results evaluated by the Entropy and Purity measurement. The smaller the Entropy value, the better the clustering effect (i.e. the probability of distributing the same topic videos into different clusters is low). Contrary to the Entropy measurement, the higher the Purity value, the better the clustering effect (i.e. the percentage of distribution of the same topic videos over the same cluster is high).

The experimental results show that both the V method and the A method give better video clustering results than the T method. This shows that the visual and audio features are dominating over the textual information when grouping videos based on individual features. In addition, the video clustering results with the method of combining each pair

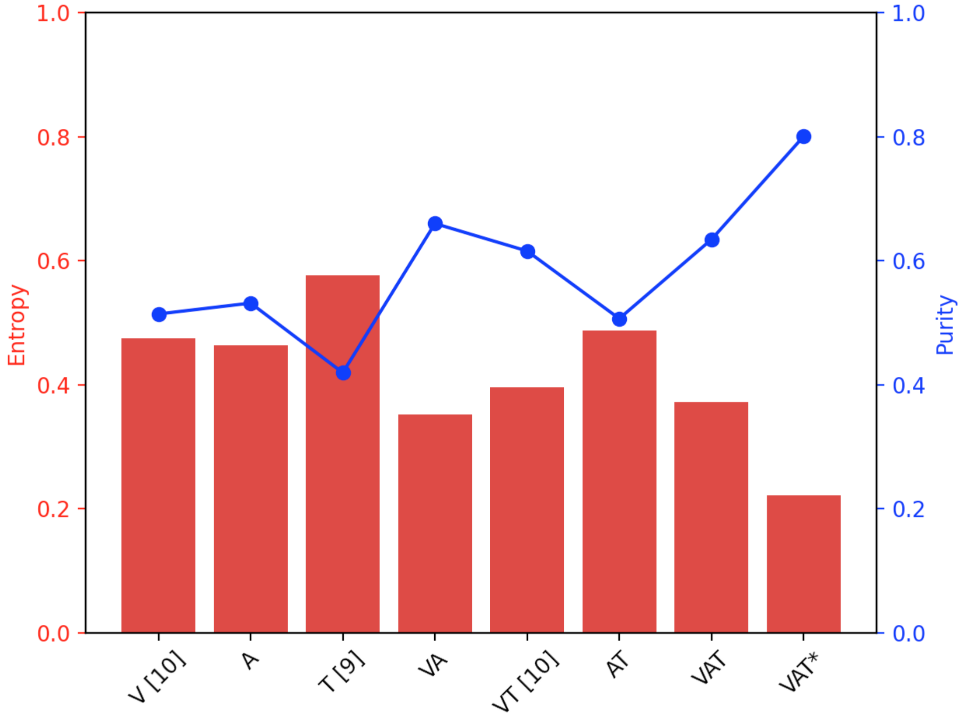


Figure 8. The clustering results evaluated by the Entropy and Purity measurement.

of different features also show that the *VA* method gives better results than both the *VT* method and the *AT*. This shows a tendency for videos with similar content (i.e. on the same topic) to have similar visual and audio features.

Given the richness and variety of video data on the online video channels, videos belong to the same topic, but can have different visual and audio features. In such cases, we hope that exploiting the textual information will help improve the video clustering quality. In general, the visual features, audio features, and textual information will complement each other, increasing the ability to exploit similarities between the videos, thereby improving the quality of video clustering results. However, the problem is how to combine to take advantage of each feature. To take this problem, we execute the *VAT* and VAT^* methods as described in the experimental settings above.

With the *VAT* method, the weight of the features considered equivalent. Experimental results show that this method also gives better results than the method using each individual feature. This again proves the effectiveness of the multi-feature combination method. However, given the variety of content of video data on the web, in each specific case, each feature type plays a different role in the process of video representation, therefore, combining multiple feature types with weight balance does not always give the best clustering results. Assuming one of the features does not represent well for the video content, combined with a balance of weights limits the dominance of the other. For example, in cases where the textual information described by the user does not match the actual content of the video, combining additional textual information with a balance of weight limits the dominance of visual and audio features. The results in Figure 8 show that the *VA* method gives better clustering results than the *VAT* method when the weights of the features are balanced.

With the VAT^* method, each feature is weighted to representing a different dominance. The results in Figure 8 show that this method gives the best video clustering results. In summary, the dominance of the features is not the same in each particular case. Our experimental results show that combining visual, audio and textual information with the right weights will significantly improve video clustering quality.

The visual result of video clustering is shown in Figure 9. For example, the users search for videos with the keyword "Scorpion", users want to search for videos related to the scorpion (i.e. animal) but most of the returned video results are related to the singer, game, helmet and others. Organizing video search results by cluster can help users locate the desired video quickly.

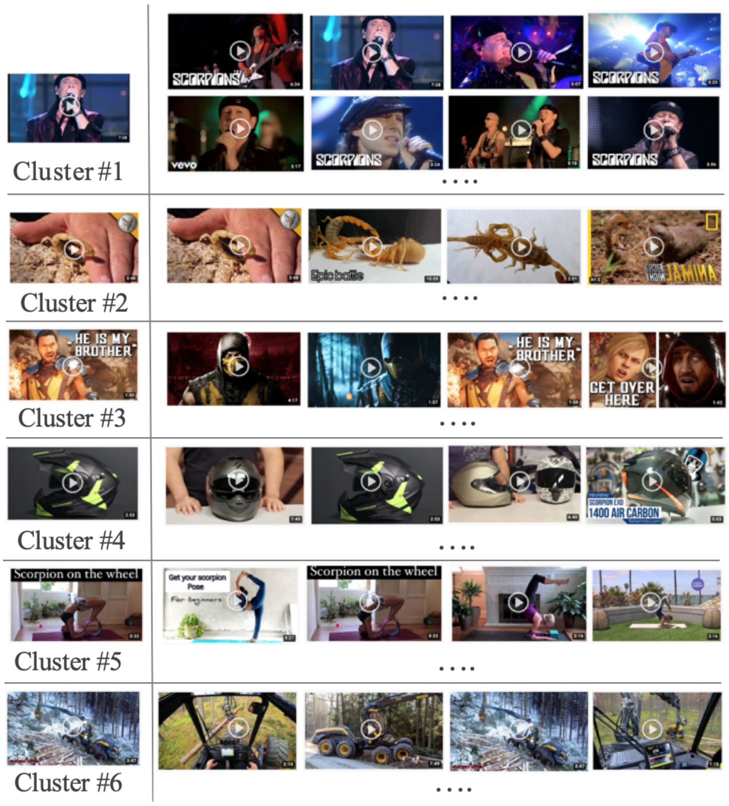


FIGURE 9. The video search results with the query "Scorpion" are visualized by cluster.

The visual result, shown in Figure 9, includes 6 video clusters related to query "Scorpion". Cluster #1 consists of music videos performed by the band Scorpions. Cluster #2 includes videos related to animals (scorpions). Cluster #3 consists of video games. Cluster #4 contains videos introducing Scorpion brand helmets. Cluster #5 includes Yoga videos (Scorpion Pose). Cluster #6 contains videos related to a specialized vehicle for harvesting pine (Ponsse Scorpion). Through video clustering results, users can easily locate the videos they are interested in instead of having to go through a flat list of search results as before.

5. **Conclusions.** In this paper, we propose a solution to organize video search results by cluster to improve user experience on the online search engine. We evaluate the influence of each feature to find the right match between feature types to improve the clustering quality of web video search results.

The experimental results show that using a combination of visual, audio, and textual information with appropriate weights has significantly improved the quality of video clustering.

REFERENCES

[1] A. Hindle, J. Shao, D. Lin, J. Lu and R. Zhang, Clustering Web Video Search Results Based on Integration of Multiple Features, in *WWW*, pp. 53–73, 2011.

[2] D. Cai, X. He, Z. Li, W. Y. Ma, J. R. Wen, Hierarchical clustering of www image search results using visual, textual and link information, in *ACM Multimedia*, pp. 952–959, 2004.

[3] F. Jing, C. Wang, Y. Yao, K. Deng, L. Zhang, W. Y. Ma, Igroup: web image search results clustering, in *ACM Multimedia*, pp. 377–384, 2006.

[4] F. Li, J. Wang, R. Lan, Z. Liu, and X. Luo, Hyperspectral image classification using multi-feature fusion, *Optics & Laser Technology*, vol.110, pp. 176–183, 2019.

[5] H. Huang, Y. Lu, F. Zhang, S. Sun, A Multi-modal Clustering Method for Web Videos, in *Trustworthy Computing and Services*, pp. 163–169, 2013.

[6] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, Deep adaptive image clustering, in *Proceedings of the IEEE international conference on computer vision*, pp. 5879–5887, 2017.

[7] J. Wang, X. Zhu, and S. Gong, Video semantic clustering with sparse and incomplete tags, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol.30, no. 1, 2016.

[8] L. Wang, J. Zhang, P. Liu, K. K. R. Choo, and F. Huang, Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification, *Soft Computing*, vol.21, no. 1, pp. 213–221, 2017.

[9] P. Q. Nguyen, A. T. Nguyen-Thi, T. D. Ngo, and T. A. H. Nguyen, Using textual semantic similarity to improve clustering quality of web video search results, in *The Seventh International Conference on Knowledge and Systems Engineering (KSE)*, pp. 156–161, 2015.

[10] P. Q. Nguyen, T. Do, A. T. Nguyen-Thi, T. D. Ngo, D. D. Le, and T. A. H. Nguyen, Clustering web video search results with convolutional neural networks, in *The 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pp. 135–140, 2016.

[11] S. Liu, M. Zhu, Q. Zheng, Mining similarities for clustering web video clips, in *CSSE*, vol.4, pp. 759–762, 2008.

[12] U. Srinivasan, S. Pfeiffer, S. Nepal, M. Lee, L. Gu, and S. Barrass, A Survey of Mpeg-1 Audio, Video and Semantic Analysis Techniques, in *Multimedia Tools and Applications*, vol.27, no.1, pp. 105–141, 2005.

[13] Y. Deldjoo, M. Elahi, M. Quadrana, and P. Cremonesi, Using visual features based on MPEG-7 and deep learning for movie recommendation, *International journal of multimedia information retrieval*, vol.7, no.4, pp. 207–219, 2018.

[14] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, Content-based video recommendation system based on stylistic visual features, *Journal on Data Semantics*, vol.5, no.2, pp. 99–113, 2016.