A Framework to Automate the generation of movies' trailers using only subtitles

Eslam Amer
Associate professor
Misr International University
Cairo, Egypt
eslam.amer@miuegypt.edu.eg

Ayman Nabil
Assistant professor
Misr International University
Cairo, Egypt
ayman.nabil@miuegypt.edu.eg

ABSTRACT
With the rapidly increasing rate of user-generated videos over the World Wide Web, it becoming a high necessity for users to navigate through them efficiently. Video summarization is considered to be one of the promising and effective approach for efficacious realization of video content by means of identifying and selecting descriptive frames of the video. In this paper, a proposed adaptive framework called Smart-Trailer (S-Trailer) is introduced to automatize the process of creating a movie trailer for any movie through its associated subtitles only. The proposed framework utilizes only English subtitles to be the language of usage. S-Trailer resolves the subtitle file to extract meaningful textual features that used to classify the movie into its corresponding genre(s). Experimentations on real movies showed that the proposed framework returns a considerable classification accuracy rate (0.89) to classify movies into their associated genre(s). The introduced framework generates an automated trailer that contains on average about (43%) accuracy in terms of recalling same scenes issued on the original movie trailer.

CCS Concepts
• **Information systems applications→Data mining→Collaborative filtering**.

Keywords
Movie trailer, natural language processing, classification

1. INTRODUCTION
The diffuse of available high-speed Internet access nowadays is the main cause that videos become the most familiar information medium on the Web. It became easy to create/search for videos that are related to some topics, watch movies through YouTube[1]. The huge amount of videos that are produced or indexed is growing at an accelerated rate. This is coupled also with a rapidly increasing rate in supplying and demanding video contents. The main issue that becomes obvious is that the time required to watch such huge amount of videos is still limited which explicate the human inability to keep up with such enormous amount of video data. Therefore, human needs an assistance to understand the video contents and hence produces a summarization for the whole video in just a few minutes.

Movie trailers can be viewed as an application of video summarization; the objective of a trailer is to encapsulate a whole 2-3 hours length movie into 2-3 minutes only.

The major hindrance affecting video summarization is the way to find distinguishable chunks of sub-videos or scenes that can be taken into consideration as significant or interesting and bypasses other chunks that are neither worthy nor expressively to viewers.

Currently, there are many handy applications that are utilized by humans to edit a video and make selections and merging of video scenes like Apple iMovie [1], Microsoft Windows Movie Maker [2], and some other online applications such as Movavi [3], MakeWebVideo [4], and IBM Watson. Nonetheless, such applications generally necessitate loading the whole video or movie to carry out their tasks. Probably, producers also need to work through many phases that include observing the movie, picking out the best shots that characterize the movie, and finally aggregating such shots in an elegant order.

Manual production of a movie trailer is considered a laborious and time-consuming. As reported by Independent article [5]; the production time of a movie trailer could be accomplished in a matter of three to four months. Production companies that create a movie trailer generally try to find some attractive keywords through which they think that it will be going to be the reason for bringing people to get a ticket and watch their movie.

Some approaches are implemented to ease the process of generating movie trailers. One of the approaches is the one introduced by Amy Paval, et.al in [6] to segmenting a video into different partitions and enriching them with short text summaries and thumbnails for each particular partition. Viewers become able to read and navigate to their favorite partitions by browsing the summary. However, the approach isn't effective in the case of movies, as there is no technique that a trailer creator provides a word and get back the corresponding scenes associated with the word yet.

Another work introduced by Zhe Xu, et.al in [7] to create an automatic movie trailer by using featured frames and shots from the movie. The work proposed by Zhe Xu uses some movie trailers as training examples to acquire some features. The

ICSIE '18, May 2–4, 2018, Cairo, Egypt

ACM ISBN 978-1-4503-6469-0/18/05…$15.00
DOI:https://doi.org/10.1145/3220267.3220293

[1] www.youtube.com

obtained features are then employed as patterns to fetch similar shots as a result when a new movie is given.

The drawback of current approaches is that it still requires totally user participation to generate trailers. Such involvements result in a considerable delay in time and great efforts produced by movie editors when working on a movie in order to generate a trailer.

In this paper, a framework is introduced that automate the process of producing movie trailers using only the textual features indexed in its subtitle. The framework utilize Natural Language Processing, and Machine Learning to analyze the included text in the subtitle file. The framework initially classifies the movie to its related genre(s), and generate a featured of significant keywords associated to the different genre(s).

The subtitle file associated with the movie is converted to a graph of sub-scenes where each node in the graph is a sub-scene, and each sub-scene is connected to other sub-scene if they have similar contents. We utilized PageRank algorithm proposed by [8] to retrieve effective sub-scene to capture their corresponding time frames.

This paper is organized as follows. Section 2 describes the related works. The proposed approach is presented in Section 3. Initial results are presented in section 4, and finally, section 5 presents the conclusion and future work.

2. RELATED WORKS

In this section, a concise overview of some approaches is introduced. As there are rarely related works so far that specifically generate an automatic movie trailer, most of the current approaches fall into the more general task of video summarization. However, the field of automatic trailer generation can be considered as an untouched field of research.

Text mining approaches that generate automatic trailer is introduced by Konstantinos Bougiatiotis, et.al [9] and R. Ren, et.al [10], they demonstrate in their works the ability to extract the topic representation from movies based on subtitle mining through inspecting the presence of a similarity correlation between the content of movie and low-level textual features from particular subtitles. The approaches presented in [9-10] generate a topical model browser for movies which allow users to scrutinize the various aspects of similarities between movies. However, the approaches presented in [9-10] doesn't take into account the movie genre(s), it can be seen as a recommendation system for movies based on the similarity of topics.

Amy Pavel, et.al [6] introduced a work that create an affordable digest that enable video browsing and skimming through segmenting videos into separate sections and providing short summaries of text to each segment. Users can navigate to a certain subject of the video by reading the summary section and pick out the corresponding video that is relevant to the section in e textual summary.

Although the work presented in [6] provides a decent infrastructure to handle the problem of searching inside a video, however, the work mainly used to partition videos according to titles, chapters, or sections. If any title is missed for any topic inside the video, the system becomes unable to summarize it correctly.

Another work introduced by J. Nessel, et.al [11] that endorse movies to users based on extracting words from the user examples. It then compares user preferences and examples with textual contents of movies. The developed system works recursively in the context of decidable languages and computable functions. However, the system lacks to get any extra preferences or opinions from users as it just relies only on anonymous keywords.

Multimedia-mining is another approach used to generate movie trailers. In the work presented by Go Irie, et.al [12], trailer generation method called Vid2Trailer (V2T) automatically generate impressive trailers from original movies through identifying audiovisual components, as well as featured key symbols such as the title logo and the theme music. V2T showed more appropriateness as compared to conventional tools. The major drawback of the V2T system is the huge processing effort which is considered too much due to speech filtering after considering the top words of the whole movie. The processing could be reduced if the generation of top-impacted keywords happened after filtering subtitles from trivial words rather than processing the whole subtitle file.

Howard Zhou, et.al [13] suggested a trailer system based on scene categorization. The system introduced by Howard utilizes intermediate structured temporally level features to improve the classification performance over the use of low-level visual features alone. Despite the slight enhancement in terms of classification performance, the system relies on a bag of visual words which indeed require a huge storage to save the bag of visual words that are related to each movie genre.

Alan F. Smeaton, et.al [14] introduced an approach that selects specific shots from action movies that facilitate the process of creating a trailer. The approach makes use of visual scenes to produce a structure of shots through identifying a shot boundary techniques for a movie. The approach analyzes also the audio track of a movie to know how to distinguish the presence of categories like speech, music, silence, speech with background music and other audio. Due to the mixture of genres in nowadays movies, it becomes necessary for any trailer generator to reflect such mixture. Alan's approach looks promising, however the approach designed specifically to suit action movies. Therefore, the system won't be able to produce a pleasing trailer if the movie has many genres.

The approaches presented in this section state considerable efforts to realize textual content, audio-video contents, or both together. However, it showed a deficiency in grasping user preferences and opinions. Today's Movies, as well as its associated trailers, comes in a variety of forms due to the diversity of cultural environments, therefore systems that generate movies' trailers should be adaptive to suit the diversity of culture environments as well as different user perspectives.

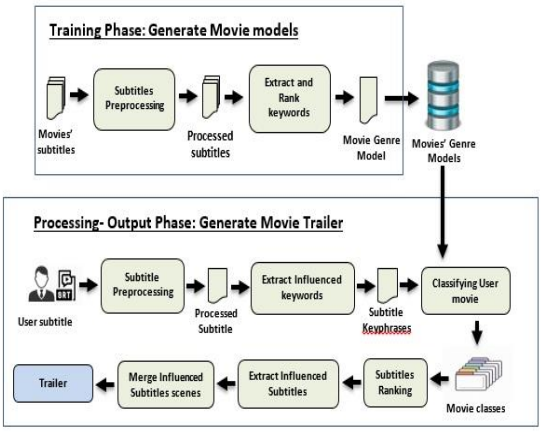

Figure 1. Smart-Trailer architectural Model

3. PROPOSED SYSTEM (S-TRAILER)

In this section, the proposed framework Smart trailer or (S-trailer) is introduced. As shown in Figure (1), S-Trailer contains two main phases namely, training, and processing-output phase. In the following subsections each phase will be described in details

3.1 Training Phase

The training phrase is essential to acquire a bag of words or generally a corpus that outlines the most common words or sentences that characterize each genre. The process is done by collecting English subtitles for the top rated movies in each genre according to IMDB *Top rated movies by genre* [15]. In the training phase, the top-20 movies for each genre were used to extract the bag of words for each movie genre or category.

A movie subtitle file contains three parts:

1 – A number that indicates scene index.

2- A time interval, that point out when the subtitle will appear, and when it disappears.

3- The script of that scene.

Figure (2) outline an example of sample subtitle from Titanic movie.

633
00:51:29,650 --> 00:51:31,686
Why can't I be like you, Jack?
634
00:51:31,770 --> 00:51:35,045
Just head out for the horizon
Whenever I feel like it.

Figure 2. Sample subtitle sequences

Where (633,634) indicating the order of subtitle in the movie sequence, 00:51:29,650 --> 00:51:31,686 showing the time duration that tells when the subtitle will appear on the screen, and when it disappears, the text "*Why can't I be like you, Jack?*" represents the script itself.

Initially, all subtitles' files are preprocessed to exclude unneeded text. This includes removing trivial characters and words which are considered insignificant to be represented as a featured text of movie genre. As reported by experimental observations in [16-19], words that are annotated as nouns or adjectives considered meaningful and worthy. Therefore the framework extracts all nouns and adjectives that occurred individually or exist in any pattern like an adjective + noun and prune anything else. The preprocessing step relied on part-of-speech tagger in NLTK library to tag words.

The objective from the training phase is to build a model for each movie genre. This is accomplished through constructing hierarchical n-grams of unique words and/or the co-occurrences of words with other words in processed documents and their frequencies. The framework rely on methodologies presented in [18, 20] to build the generation model for each genre.

Each word or keyphrase in the resulting model is given rank using keyphrase ranking algorithm presented in [20] with some minor modification.

$$Rank(i) = log\left(p(i)\frac{TF_i + TI_i}{L}\right) \qquad (1)$$

Where p_i is the position of entry *i*. The position is computed as (*L*–*Ls*) where L represents the total lines in the document, *Ls* is the first sentence where entry *i* occurs. TF_i , TI_i indicate the term frequency, influence weight for entry *i* respectively.

The final outcome of the training phase is set of genres' models, each model contains a list of weighted keyphrases. Each model can be viewed as a sign that represents specific movie genre. Eventually, the generated models are stored in a genre dictionary.

3.2 Processing-Output Phase

The processing phase is considered the essence phase of the framework model. In this phase, the user supplies the model with the movie subtitle which he needs to generate a trailer for. The supplied subtitle will undergo in the same process similar to the training phase to produce the featured words from user subtitle.

The featured words list generated from user subtitle is compared against genres lists stored at movie genres' models in order to classify user subtitle into its related movie genre(s). The classification is done through Naïve Bayes classifier. In general, a movie could be related to several genres in different percentages depending on type and number of scenes that are related to each genre. For example, Titanic[2] movie contains two genres (Drama, and Romance). Therefore, the classification results represent percentages of ordered genres that are closely related to user subtitle (i.e., 70% Action, 30% Drama). The percentages returned by the classifier is used as guidance to the proposed framework to allocate the trailer with types of scenes.

When it comes to rank the influenced scenes inside the movie, it becomes necessary to initially build a graph of movie scenes. The established graph is represented as NxN adjacency matrix, where N is the number of sequences in the subtitle file. The matrix values of rows and columns are computed using the following equation:

$$A(i,j) = \begin{Bmatrix} 0 if(i=j) \\ CosineSimilarity(i,j) if(i \neq j) \end{Bmatrix} \qquad (2)$$

Where $A(i,j)$ denotes the relationship between item *i* and item *j* in the adjacency matrix, the cosine similarity is used to measure the similarity value between two sequences in the subtitle file. The value of $A(i,j)$ will reflect the degree of similarity between two sequences in the subtitle file [18]. The generated matrix will represent the relations in terms of similarity between scenes or sequences inside the subtitle file where vertices represent the scenes and edges showed how scenes are related to each other.

The proposed framework utilizes PageRank algorithm proposed by [8] to rank the graph represented in the resultant adjacency matrix. PageRank is used to weight influential sequences which is the most popular sequences in the subtitle file.

The result of the ranking module is a set of ordered weighted sequences and fetch their associated time frame. The proposed framework selects the *top-K* sequences to be presented in the trailer scenes. Determination for the value of *K* is depending on the required time duration for the trailer.

The associated time frames come with *top-K* sequences are passed to video editing library in python, the library contains methods that fetch the corresponding scenes from the original movie. All fetched scenes are aggregated, merged, and then finally passed to the user as an output trailer.

[2] http://www.imdb.com/title/tt0120338/

4. EXPERIMENTS AND RESULTS

In this section, experimental tests and evaluations are presented in order to prove the validity of the proposed framework. Evaluations of the framework tends to be focused on evaluation of movie genre(s) lists, and the accuracy of the generated trailer.

The movie genres models resulted from the training phase are tested against Kaggle movie dataset which is available to download from [21]. Kaggle movie dataset contains about 5000 movies that are related to different genres. Each movie in the dataset comes with its associated IMDB genre(s). For the purpose of testing the effectiveness of the generated movie genre(s) model, a group of random 500 movie are selected from Kaggle movie dataset. For each movie in the selected dataset, its English subtitle is downloaded from Subscene[3] Homepage.

Experimental results on the first outcome generated by the proposed framework; which is the movie genres model (Table 1) showed a strong correlation in classification between the movies' genres originally indexed in IMDB and the generated movie genres model.

Table 1: Evaluating S-trailer genre classification accuracy

Genre	Classification Accuracy
Action	89%
Comedy	75%
Drama	72%
Romance	60%
Fantasy	55%

The order of genre appearance is taken into consideration in the evaluation process of classification accuracy. For example, if the original appearance of the genre in a movie is Action, Crime, and Drama; the classifier has to return identical order of genres or result of classification will be regarded as misclassification.

As there is no standard golden corpus that could be used to classify movies into its related category or genre(s), the movie genre model resulted from the framework can be viewed as a seed corpus for movies' classification.

The second experimental evaluation is the evaluation of the produced trailer. Evolutions to generated trailers will be based on *precision*, *recall*, and *F-measure* metrics:

Where PRECISION is defined as the ratio of the number of relevant scenes retrieved to the total number of scenes indexed in original trailer [22], it is calculated as:

$$Precision = \frac{Numberofrelevantscenesretrieved}{totalnumberoforiginaltrailerscenes} \quad (3)$$

RECALL is defined as the ratio of the number of correct or relevant scenes retrieved to the total number of relevant scenes indexed in the produced trailer [22], and it is calculated as:

$$Recall = \frac{Numberofrelevantscenesretrieved}{totalnumberofgeneratedtrailerscenes} \quad (4)$$

And, F-MEASURE is considered as the weighted harmonic mean of precision and recall [22], and it is calculated as:

$$F - measure = \frac{2xPrecesionxRecall}{Precesion+Recall} \quad (5)$$

[3] https://subscene.com/

For the purpose of evaluation, random 10 movies were selected for each genre which counts to a total of 50 movies used to evaluate the performance of the proposed model. Table (2) show the average performance of S-trailer in retrieving similar scenes presents in the original trailer for top 10, 30, and 50 scenes respectively.

Table 2: Evaluating S-trailer Performance according to retrieved scenes

Genre	Top 10 Scenes			Top 30 Scenes			Top 50 Scenes		
	PR	RC	FM	PR	RC	FM	PR	RC	FM
Action	0.062	0.174	0.091	0.172	0.217	0.192	0.372	0.427	0.398
Comedy	0.045	0.152	0.069	0.092	0.185	0.123	0.215	0.355	0.268
Drama	0.047	0.133	0.069	0.100	0.203	0.134	0.310	0.412	0.354
Romance	0.052	0.137	0.075	0.101	0.206	0.136	0.321	0.389	0.352
Fantasy	0.044	0.121	0.065	0.089	0.211	0.125	0.299	0.345	0.320

Where PR, RC, and FM stands for precision, recall, and F-measure respectively.

A key observation point is that the generated trailer accuracy is increased in terms of precision and recall when the number of test scenes increased which indicate the reliability of the proposed framework in fetching valuable scenes.

However, it is observed that some movies' trailers contains lots of silence scenes. It tends that producer's main focus is to catch user's anxiety or fear especially in horror movies or some action movies. For example, the trailer for movie like American Sniper 2, it was observed that the majority of scenes that were used in the official trailer contains no speech (silence) or a silence scene with dialog in background that is not related to the scene. In such cases, the performance of framework generated trailer become very weak.

Experimental results showing the Smart-Trailer framework provide a promising results. It achieves a recall accuracy ratio of 43% in Top-50 scenes retrieved by the smart trailer. A major drawback of the proposed framework is that, it cannot fetch silence senses, which are scenes where there is no speech in it. It is noted that producers like that type of scenes for the purpose of attraction or surprising especially in horror and romance movies. However, such drawback will going to be overcome in the future work.

5. CONCLUSION AND FUTURE WORK

In this paper, a framework called a Smart-Trailer is proposed to automate the process of trailer generation relying on natural language processing and machine learning. Smart-Trailer framework that originally introduced in [23] revealed how it can be used successfully in the field of marketing through the phases described before in order to generate an attractive trailer to the audience. The framework establishes efficiently a golden corpus for each movie category through which it can be used to classify any movie into its related genre(s). The main contribution of Smart-trailer is its capability to generate a trailer without any human involvement. Smart-Trailer returns an average of 43% in terms of accuracy in recalling scenes exist in the original trailer scenes.

The future work includes enhancements to the framework by extracting latent information indexed in silence scenes that could be reflected in an increase of the accuracy rate. The framework will also add a recommendation module that grasps user

behaviour, and suggest or recommend user with special scenes that likely matches user preferences.

6. REFERENCES

[1] iMovie - Apple.

https://www.apple.com/lae/imovie/

[2] Windows Movie Maker

https://www.windowsmovie-maker.org/

[3] Movie Trailer Maker—How to Make a Movie Trailer

Movavi.

https://www.movavi.com/support/how-to/how-tomake-a- movie-trailer.html.

[4] Create Your Own Movie Trailer With Our Online Video Maker."https://www.makewebvideo.com/en/make/movie-trailervideo.

[5] The Independent. "We spoke to the people who make film trailers " 17 Jan. 2017, http://www.independent.co.uk/artsentertainment/films/features/film-trailers-editors-interviewcreate-teasers-tv-spots-a7531076.html.

[6] A. Pavel, C. Reed, B. Hartmann, and B. Hartmann, Video dig ests: a browsable, skimmable format for informational lecture videos, in UISTUser Interface Software and Technology, SIGGRAPH ACM Special Interest Group on Computer Graphics and Interactive Techniques. NY, USA: ACM New York, October 2014, pp. 573-582.

[7] Z. Xu and Y. Zhang, Automatic generated recommendation for movie trailers, in Broadband Multimedia Systems and Broadcasting (BMSB), 5-7 June 2013, London, UK. IEEE, October 2013. [Online

[8] Ying Ding, et.al. PageRank for Ranking Authors in Co-citation Networks. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY, 60(11):2229–2243, 2009

[9] K. Bougiatiotis and T. Giannakopoulos, Content representation and similarity of movies based on topic extraction from subtitles, in SETN 16 Proceedings of the 9th Hellenic Conference on Artificial Intelligence, May 18 - 20, 2016 ,

[10] R. Ren, H. Misra, and J. Jose. Semantic based adaptive movie summarization. In S. Boll, Q. Tian,L. Zhang, Z. Zhang, and Y.-P. Chen, editors, Advances in Multimedia Modeling, volume 5916 of Lecture Notes in Computer Science, pages 389-399.Springer Berlin Heidelberg, 2010.

[11] J. Nessel and B. Cimpa, The movieoracle - content based movie recommendations, in Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on, 22-27 Aug.2011, Lyon, France. IEEE, October 2011.

[12] G. Irie, T. Satou, A. Kojima, T. Yamasaki, and K. Aizawa, Automatic trailer generation, in MM 10 Proceedings of the 18th ACM international conference on Multimedia, Firenze, Italy, SIGMULTIMEDIA ACM Special Interest Group on Multimedia. ACM New York, NY, October 2010, pp. 839-842

[13] H. Zhou, T. Hermans, A. V. Karandikar, and J. M. Rehg, Movie genre classification via scene categorization, in MM 10 Proceedings of the 18th ACM international conference on Multimedia, October 25 - 29, 2010 , Firenze, Italy, SIGMULTIMEDIA ACM Special Interest Group on Multimedia. New York,USA: ACM New York, NY, October 2010, pp. 747-750

[14] A. F. Smeaton, B. Lehane, N. E. OConnor, C. Brady, and G. Craig, Automatically selecting shots for action movie trailers, in MIR 06 Proceedings of the 8th ACM international workshop on Multimedia information retrieval, Santa Barbara, California, USA, SIGGRAPH ACM Special Interest Group on Computer Graphics and Interactive Techniques. New York, USA: ACM New York, NY, October 2006, pp. 231-238

[15] www.imdb.com/

[16] Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Collins, M., Steedman, M. (eds.) Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 216 –223 (2003).

[17] Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of the 23rd National Conference on Artificial intelligence, AAAI 2008, vol. 2, pp. 855 −860. AAAI Press (2008)

[18] Eslam Amer. Enhancing Efficiency of Web Search Engines through Ontology Learning from unstructured information sources, Proceeding of 16th IEEE International conference of Information Integration and Reuse (IRI2015), PP.542- 549, 13-15 August 2015. San Francisco, USA.

[19] Youssif, Aliaa AA, Atef Z. Ghalwash, and Eslam A. Amer. "HSWS: Enhancing efficiency of web search engine via semantic web." In Proceedings of the International Conference on Management of Emergent Digital EcoSystems, pp. 212-219. ACM, 2011.

[20] Aliaa A.A. Youssif, Atef Z. Ghalwash, and Eslam Amer. KPE: An Automatic Keyphrase Extraction Algorithm, Proceeding of IEEE International Conference on Information Systems and Computational Intelligence (ICISCI 2011), pp. 103 -107, 2011.

[21] "Kaggle." https://www.kaggle.com/

[22] Eslam Amer, and Khaled Foad. "Akea: an Arabic keyphrase extraction algorithm." In *International Conference on Advanced Intelligent Systems and Informatics*, pp. 137-146. Springer, Cham, 2016

[23] Mohammed Hesham, Bishoy Hany, Nour Foad, and Eslam Amer. " Smart Trailer: Automatic generation of movie trailer using only subtitles." The First International Workshop on Deep and Representation Learning, *IWDRL 2018*. PP.26-30. IEEE, 2018