GIBIS at MediaEval 2017: Predicting Media Interestingness Task

Jurandy Almeida and Ricardo M. Savii
GIBIS Lab, Institute of Science and Technology, Federal University of São Paulo – UNIFESP
12247-014, São José dos Campos, SP – Brazil
{jurandy.almeida,ricardo.manhaes}@unifesp.br

ABSTRACT

This paper describes the GIBIS team experience in the *Predicting Media Interestingness Task* at MediaEval 2017. In this task, the teams were required to develop an approach to predict whether images or videos are interesting or not. Our proposal relies on late fusion with rank aggregation methods for combining ranking models learned with different features and by different learning-to-rank algorithms.

1 INTRODUCTION

In this paper, we explore the use of rank aggregation methods for predicting the interestingness of images and videos. For that, content-based representations for images and videos are obtained by different features, which are used to train different learning-to-rank algorithms, creating rankers capable of predicting the interestingness degree of images and videos. Then, the information provided by different pairs of feature-ranker are combined by rank aggregation methods, yielding more effective predictions [3].

This work is developed in the context of the MediaEval 2017 Predicting Media Interestingness Task, whose goal is to automatically select the most interesting frames or portions of videos according to a common viewer by using features derived from audio-visual content or associated textual information. Details about data, task, and evaluation are described in [7].

2 PROPOSED APPROACH

The start point for our proposal is the work of Almeida [1], where motion features were extracted from videos and then used to train four different ranking models, which were combined with a majority voting strategy [13]. The key idea exploited in the Almeida's work was the use of multiple learning-to-rank algorithms, and their combination was pointed out as promising.

Here, we extend the work of Almeida [1] by exploring rank aggregation methods for combining ranking models learned with different features and by different learning-to-rank algorithms.

2.1 Features

Images. For the image subtask, we used only the pre-computed features provided by the task organizers [7]. Five low-level features were considered: Dense SIFT, Histogram of Gradients (HoG), Local Binary Patterns (LBP), GIST, and Color Histogram. Also, two deep learning features were used and they refer to Convolutional Neural Network (CNN) features extracted from the last layers (i.e., fc7 and prob) of the pre-trained AlexNet model [11].

Videos. For the video subtask, we used nine pre-computed features provided by the task organizers [7]. One of them represents audio information: Mel-Frequency Cepstral Coefficients (MFCC). Seven features are the same used for images and encode visual content: five low-level features (Dense SIFT, HoG, LBP, GIST, and Color Histogram) and two deep learning features (CNN-fc7 and CNN-prob). These eight features are frame-based representations [11]. To obtain a single video representation, we built a Bag-of-Features (BoF) [4] model for each feature. In the BoF framework, visual words [15] are obtained by quantizing a feature space according to a pre-learned dictionary. Thus, a video is represented as a normalized frequency histogram of visual words associated with each feature. In this work, we construct a codebook of 4000 visual words using a random selection. In addition, we considered three video-based representations. One of them is also a pre-computed feature provided by task organizers, denoted C3D [16]. The two others refer to additional visual features we extracted from videos: Histogram of Motion Patterns (HMP) [2] and Bag-of-Attributes (BoA) [8].

2.2 Learning-to-Rank Algorithms

Each of the above features was used as input to train four different learning-to-rank algorithms, which are the same used in [1]. The first three are based on pairwise comparisons: *Ranking SVM* [12], *RankNet* [5], and *RankBoost* [10]. The latter approach considers lists of objects by using *ListNet* [6].

The SVM^{rank} package[1] [12] was used for running Ranking SVM. The RankLib package[2] was used for running RankNet, RankBoost, and ListNet. Ranking SVM was configured with a linear kernel. The others were configured with their default parameter settings.

2.3 Rank Aggregation Models

Let $C = \{o_1, o_2, \ldots, o_n\}$ be a collection of n objects (i.e., images or videos). Let $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ be a set of m feature-ranker pairs. Let $\rho_j(i)$ be the interestingness degree assigned by the feature-ranker pair $r_j \in \mathcal{R}$ to the object $o_i \in C$. Based on the score ρ_j, a ranked list τ_j can be computed. The ranked list τ_j can be defined as a permutation of the collection C, which contains the most interesting objects according to the feature-ranker pair r_j. A permutation τ_j is a bijection from the set C onto the set $[n] = \{1, 2, \ldots, n\}$. For a permutation τ_j, we interpret $\tau_j(i)$ as the position (or rank) of the object o_i in the ranked list τ_j. We can say that, if o_i is ranked before o_k in the ranked list τ_j, that is, $\tau_j(i) < \tau_j(k)$, then $\rho_j(i) \leq \rho_j(k)$ [3].

Given the different scores ρ_j and their respective ranked lists τ_j computed by distinct pairs $r_j \in \mathcal{R}$, a rank aggregation method aims to compute a fused score $F(i)$ to each object o_i [3]. In this work, we used three different methods based on score and rank information:

(1) *Borda Method* [17]: $F(i) = \sum_{j=0}^{m} \tau_j(i)$,

MediaEval'17, 13-15 September 2017, Dublin, Ireland

[1] https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html (As of August 2017)
[2] https://sourceforge.net/p/lemur/wiki/RankLib/ (As of August 2017)

(2) *Multiplicative Approach* [14]: $F(i) = \prod_{j=1}^{m}(1 + \rho_j(i))$,

(3) *Weighted Sum Model* [9]: $F(i) = \sum_{j=1}^{m}(\tau_j(i) \times \rho_j(i))$.

3 EXPERIMENTS & RESULTS

Five different runs were submitted for each subtask configured as shown in Table 1[3]. For both subtasks, the first run is the best feature-ranker pair in isolation and the others refer to the fusion of the top performing feature-ranker pairs with rank aggregation methods. All the evaluated approaches were calibrated through a 3-fold cross validation on the development data.

Table 1: Configuration of the submitted runs.

Subtask	Run	Fusion	Feature-Ranker Pairs
Image	1	-	CNN-fc7 & RankBoost
	2	Weighted Sum	CNN-fc7 & RankBoost, CNN-fc7 & RankNet
	3	Multiplicative	CNN-fc7 & RankBoost, CNN-fc7 & RankNet, CNN-fc7 & RankSVM, CNN-prob & RankSVM
	4	Borda	
	5	Weighted Sum	
Video	1	-	HMP & RankSVM
	2	Multiplicative	HMP & RankSVM, MFCC & RankSVM
	3	Multiplicative	HMP & rankSVM, HMP & RankBoost, C3D & RankNet, HoG & RankSVM, Dense SIFT & RankBoost
	4	Borda	
	5	Weighted Sum	

The development data is composed of 7,396 videos from 78 movie trailers. For the image subtask, the middle keyframe of each video was extracted, forming a dataset with 7,396 images. Each of the features (Section 2.1) was used as input to train each of the learning-to-rank algorithms (Section 2.2). In this way, we obtained 28 feature-ranker pairs (i.e., 7 features × 4 rankers) for the image subtask and 44 feature-ranker pairs (i.e., 11 features × 4 rankers) for the video subtask. Next, each of the feature-ranker pairs was used to predict the interestingness degree of test images and videos. Finally, the prediction scores of the top performing feature-ranker pairs in isolation were combined using rank aggregation methods (Section 2.3), producing fused prediction scores.

To assess the effectiveness of each approach, we computed the Mean Average Precision (MAP). For that, we transformed prediction scores into binary decisions using the strategy proposed in [1]. First, the prediction scores associated with images and videos of a same movie trailer were normalized using a z-score normalization. Then, an empirical threshold of 0.7 was applied to the normalized prediction scores, producing binary decisions.

Table 2 presents MAP scores obtained for each run on the development data. For both subtasks, the fusion of the top performing feature-ranker pairs (runs 2 to 5) performed better than the best feature-ranker pair in isolation (run 1). The only exception was the run 2 of the video subtask, which was a required run for the task where the use of audio features (i.e., MFCC) was mandatory. All the machine-learned rankers using MFCC achieved poor results. By analyzing the confidence intervals, it can be noticed that the results achieved by the rank aggregation methods seem promising.

[3]The run 1 of the image subtask and the run 2 of the video subtask were the required runs for the task, while the other runs were optional.

Table 2: MAP results obtained on the development data.

Subtask	Run	Avg. MAP	Confidence Interval (95%)	
			min.	*max.*
Image	1	27.78	22.78	32.77
	2	28.95	23.17	34.72
	3	29.36	25.18	33.53
	4	28.98	24.53	33.43
	5	29.74	25.28	34.19
Video	1	22.41	21.48	23.34
	2	21.85	20.65	23.05
	3	23.43	22.77	24.09
	4	23.19	21.68	24.70
	5	23.07	21.88	24.27

Table 3 presents the official results reported for 2,435 videos and images from 30 movie trailers of the test data. MAP is a good indication of the effectiveness considering all the results (i.e., images or videos) of the same movie trailer. MAP@10, in turn, focuses on the effectiveness considering only the 10 results classified as the most interesting ones. On one hand, for the image subtask, the best results were achieved by a feature-ranker pair in isolation (run 1). On the other hand, for the video subtask, the use of rank aggregation methods (runs 2 to 5) improved the overall performance. One of the reasons is the strategy used for selecting the feature-ranker pairs to be combined by the rank aggregation methods. For that, we sorted all the pairs in an increasing order of MAP. We believe the ordering obtained on the development and test data may not be consistent.

Table 3: Official results reported for the test data.

Subtask	Run	MAP	MAP@10
Image	1	27.10	11.29
	2	26.45	10.29
	3	25.02	09.24
	4	25.25	09.16
	5	25.31	09.39
Video	1	16.67	03.96
	2	18.07	05.30
	3	18.77	06.14
	4	18.36	06.24
	5	18.30	06.28

4 CONCLUSIONS

Our approach has explored rank aggregation methods for combining feature-ranker pairs. Obtained results demonstrate that the proposed approach is promising. Future work includes the investigation of a smarter strategy for selecting the pairs to be combined.

ACKNOWLEDGMENTS

We thank the São Paulo Research Foundation - FAPESP (grant 2016/06441-7) and the Brazilian National Council for Scientific and Technological Development - CNPq (grant 423228/2016-1) for funding. This work has also benefited from the support of the Association for the Advancement of Affective Computing (AAAC) and the ACM Special Interest Group on Information Retrieval (SIGIR).

REFERENCES

[1] J. Almeida. 2016. UNIFESP at MediaEval 2016: Predicting Media Interestingness Task. In *Proc. of the MediaEval 2016 Workshop*. http://ceur-ws.org/Vol-1739/MediaEval_2016_paper_28.pdf

[2] J. Almeida, N. J. Leite, and R. S. Torres. 2011. Comparison of Video Sequences with Histograms of Motion Patterns. In *IEEE Intl. Conf. Image Processing (ICIP'11)*. 3673–3676.

[3] J. Almeida, L. P. Valem, and D. C. G. Pedronette. 2017. A Rank Aggregation Framework for Video Interestingness Prediction. In *Intl. Conf. Image Analysis and Processing (ICIAP'17)*. 1–11.

[4] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. 2010. Learning Mid-Level Features for Recognition. In *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR'10)*. 2559–2566.

[5] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. 2005. Learning to rank using gradient descent. In *Intl. Conf. Machine Learning (ICML'05)*. 89–96.

[6] Z. Cao, T. Qin, T-Y. Liu, M-F. Tsai, and H. Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Intl. Conf. Machine Learning (ICML'07)*. 129–136.

[7] C-H. Demarty, M. Sjöberg, B. Ionescu, T-T. Do, M. Gygli, and N. Q. K. Duong. 2017. MediaEval 2017 Predicting Media Interestingness Task. In *Proc. of the MediaEval 2017 Workshop*. Dublin, Ireland.

[8] L. A. Duarte, O. A. B. Penatti, and J. Almeida. 2016. Bag of Attributes for Video Event Retrieval. *CoRR* abs/1607.05208 (2016). http://arxiv.org/abs/1607.05208

[9] P. C. Fishburn. 1967. *Additive Utilities with Incomplete Product Set: Applications to Priorities and Assignments*. Operations Research Society of America (ORSA).

[10] Y. Freund, R. D. Iyer, R. E. Schapire, and Y Singer. 2003. An Efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research* 4 (2003), 933–969.

[11] Y-G. Jiang, Q. Dai, T. Mei, Y. Rui, and S-F. Chang. 2015. Super Fast Event Recognition in Internet Videos. *IEEE Transactions on Multimedia* 17, 8 (2015), 1174–1186.

[12] T. Joachims. 2006. Training linear SVMs in linear time. In *ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining (ACM SIGKDD'06)*. 217–226.

[13] L. Lam and C. Y. Suen. 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Trans. Systems, Man, and Cybernetics, Part A* 27, 5 (1997), 553–568.

[14] D. C. G. Pedronette and R. S. Torres. 2013. Image Re-Ranking and Rank Aggregation based on Similarity of Ranked Lists. *Pattern Recognition* 46, 8 (2013), 2350–2360.

[15] J. Sivic and A. Zisserman. 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *IEEE Intl. Conf. Computer Vision (ICCV'03)*. 1470–1477.

[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE Intl. Conf. Computer Vision (ICCV'15)*. 4489–4497.

[17] H. P. Young. 1974. An axiomatization of Borda's rule. *Journal of Economic Theory* 9, 1 (1974), 43–52.