MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommendation and Retrieval

Yashar Deldjoo
Politecnico di Milano, Italy
deldjooy@acm.org

Mihai Gabriel Constantin
University Politehnica of Bucharest, Romania
mgconstantin@imag.pub.ro

Bogdan Ionescu
University Politehnica of Bucharest, Romania
bionescu@imag.pub.ro

Markus Schedl
Johannes Kepler University, Austria
markus.schedl@jku.at

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

ABSTRACT

In this paper we propose a new dataset, *i.e.,* the MMTF-14K multifaceted dataset. It is primarily designed for the evaluation of video-based recommender systems, but it also supports the exploration of other multimedia tasks such as popularity prediction, genre classification and auto-tagging (aka tag prediction). The data consists of 13,623 Hollywood-type movie trailers, ranked by 138,492 users, generating a total of almost 12.5 million ratings. To address a broader community, metadata, audio and visual descriptors are also pre-computed and provided along with several baseline benchmarking results for uni-modal and multi-modal recommendation systems. This creates a rich collection of data for benchmarking results and which supports future development of this field.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

video recommendation, video trailer benchmarking dataset, content description, social media.

ACM Reference Format:
Yashar Deldjoo, Mihai Gabriel Constantin, Bogdan Ionescu, Markus Schedl, and Paolo Cremonesi. 2018. MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommendation and Retrieval. In *MMSys'18: 9th ACM Multimedia Systems Conference, June 12–15, 2018, Amsterdam, Netherlands.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3204949.3208141

1 INTRODUCTION

In recent years we have witnessed an unprecedented explosion of video content created, shared, and consumed through various web channels, such as YouTube[1] or Vimeo[2]. According to Cisco's latest forecast, more than 75% of the world's mobile data traffic will be video by the end of 2020 and reaches more than 80% for audio-video[3]. Apart from the volume, there is also the diversity of the video content, *e.g.,* user-generated videos, movies, music video clips, and so on. It has become harder and harder for the users to find interesting new content using the traditional search tools. As the result, recommender systems (RS) that automatically predict content that a user may like have emerged and evolved during the last decade [2, 22].

There is an obvious need for researchers and practitioners to have access to stable, large-scale, and multimodal datasets of movies to research personalization, search/retrieval, and recommender systems. Past efforts to establish such datasets include the Netflix[4] and EachMovie datasets, both no longer available. Perhaps the most important, still available, is the MovieLens (ML) dataset [14], which contains timestamped preference information of users for movies, in order to facilitate research on personalized movie search and recommendation. These preferences originate from users of MovieLens[5], which is a movie RS. Several versions of the dataset have been released since its launch in 2005 which foremost differ with regards to the number of users, items and ratings as well as the availability of user-generated tags as items' metadata.

However, one frequently expressed concern about such datasets is related to the lack of real content features, which describe audio and visual properties of the movies. In fact, while in the multimedia retrieval community, content descriptors extracted from the audio-and visual channels have been researched intensely, the RS community interpreted for a long time the term "content" to refer to metadata only. In this vein, datasets like ML [14] and Yahoo! Movies WebScope dataset [1] provide metadata as "content" features, and it is argued that these describe to some extent the content of movies, either by reflecting expert knowledge in case of editorial information, or the wisdom of the crowd in case of user-generated tags or keywords.

As recent research showed [9, 12], many aspects of the audio-visual data of movies are not properly reflected in metadata. In particular, the perception of a movie is influenced by factors not only related to the genre, cast, and plot, but also to the overall film style [5]. These factors affect the viewer's experience. For

[1] https://www.youtube.com
[2] https://www.vimeo.com

[3] http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html
[4] https://www.netflix.com
[5] http://www.movielens.org

MMSys'18, June 12–15, 2018, Amsterdam, Netherlands

ACM ISBN 978-1-4503-5192-8/18/06...$15.00
https://doi.org/10.1145/3204949.3208141

Table 1: Most relevant datasets created for the development of recommender systems (M - metadata, A - audio, V - video, $|\mathcal{I}|$ — number of items, $|\mathcal{U}|$ — number of users, $|\mathcal{P}|$ — number of ratings).

Dataset	Domain	Content Feature	Preference Type	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{P}\rvert$
MovieLens 20M (ML-20M) [14]	movie	M	ratings [1-5]	26.7K	138.5K	20M
Yahoo! Movies WebScope dataset [1]	movie	M	ratings [F-A+]	9K	2K	91K
LDOS-CoMoDa dataset [18]	movie	M + context	ratings [1-5]	1K	1K	2K
Million Song Dataset [4]	music	A, M	listening events	1M (track)	1M	48M
Million Musical Tweets [15]	music	A, M	listening events	134K (track), 25K (artist)	215K	1M
LFM-1b [23]	music	M	listening events	32M (track), 3M (artist)	120K	1.1B
MMTF-14K	**movie**	**M, A, V**	**ratings [1-5]**	**13.6K**	**138.5K**	**12.4M**

example, two movies may be from the same genre and director, but they can be different based on the movie style: "Empire of the Sun" and "Schindler's List" are both dramatic movies directed by Steven Spielberg and describing historical events, however, they are completely different in style; "Schindler's List" is shot like a documentary in black and white, while "Empire of the Sun" is shot using bright colors and making heavy use of special effects. Although these two movies are similar with respect to traditional metadata (*e.g.,* director, genre, year of production), their different styles are likely to affect the viewers' feelings and opinions in a different way [8, 9].

Addressing in particular these limitations, the main contribution of the work in this paper is the design and release of a publicly available multifaceted movie trailer dataset (MMTF-14K). The remainder of the article is structured as follows. Section 2 presents a review of previous datasets created in the recommender systems community and positions our contribution. Section 3 describes the content of MMTF-14k, including the provided audio, visual and metadata features. Section 4 describes the ground truth associated with the data. Finally, some baseline results are discussed in Section 5, while Section 6 concludes the paper.

2 PREVIOUS WORK

In the video domain, perhaps the most important, still available dataset for recommendation tasks is the MovieLens (ML) dataset [14]. It contains timestamped preference information of users for movies. Several versions of the dataset have been released, which foremost differ with respect to the number of ratings, users, and items: ML-100K, ML-1M, ML-10M, and ML-20M. In 2005, ML introduced tagging facilities, and in turn included tag information in the later ML dataset (10M and 20M). Due to the substantial value that such datasets provide in exploring and validating ideas related to personalization and recommendation research, the ML datasets have been widely appreciated by the community, been heavily used and referenced in the research literature ever since (*e.g.,* 7,500+ references to ML in Google Scholar) [14].

Among the few other available video datasets, we can mention the Yahoo! Movies WebScope dataset [1]. It contains a small percentage of the movie community's preferences for various movies, rated on a scale from A+ to F. The dataset also provides a large number of descriptive information, but limited to movies which were released prior to November 2003. The metadata information include information such as cast, crew, synopsis, genre, average ratings, awards. Another example is the LDOS-CoMoDa dataset [18]. It is a movie recommender dataset which contains community ratings given to movies and introduces twelve pieces of contextual information in which the movies were consumed, such as time, day type, season, weather, mood and health-condition. The dataset is designed to facilitate research on context-aware movie recommender systems (CARS). As a general observation, none of these data come with advanced, precomputed, audio and visual descriptors.

Unlike the movie domain, music recommendation [24] has produced much more publicly available resources. The most well-known is the Million Song Dataset (MSD) [4], which was released in 2012 in conjunction with the MSD Challenge[6]. MSD integrates various pieces of information in one million contemporary popular music pieces. Other examples are the the Million Musical Tweets Dataset (MMTD) [15] and the LFM-1b dataset [23]. Although these data have a different focus than our topic (movie recommendation), we consider them important to be mentioned as they are related to the audio information.

In this paper, we propose and release the MMTF-14K dataset which is designed to be used for building movie RS using latest advances in audio-visual content representations. The dataset is publicly available and consists of 13,623 Hollywood-type movie trailers, ranked by 138,492 users with a total of almost 12.5 million ratings. A summary of the features and a comparison with other datasets is presented in Table 1. To the best of our knowledge, the MMTF-14K dataset is the first large-scale dataset in the recommender systems community that provides all types of content-based descriptors in conjunction with metadata. MMTF-14K is also multifaceted allowing to develop connections with other related domains such as: *popularity prediction* — in the RS community, the common way to calculate popularity is based on how much a movie has received attention/interaction from the user community. Formally, the popularity of item *i* is calculated as the fraction of users who have rated item *i*, over the total number of users [28]. Since MMTF14K provides links to the ML ratings dataset, we can measure popularity based on number of interaction each movie has received; *genre classification* — predict movie genre by using multimedia descriptors. Given the 18 binary genre labels, the problem is in essence a multi-label classification problem; *tag-prediction (auto tagging)* — automatically predict/recommend tags given a media content. This is done by learning the association between textual multimedia features and tag keywords.

[6] https://labrosa.ee.columbia.edu/millionsong/challenge

Table 2: Distribution of genre labels in MMTF-14K.

#	Genre	#pos	#neg	skewness
1	Action	1,766	11,857	6.71
2	Adventure	1,202	12,421	10.33
3	Animation	482	13,141	27.26
4	Children	592	13,031	22.01
5	Comedy	4,139	9,484	2.29
6	Crime	1,473	12,150	8.25
7	Documentary	1209	12414	10.27
8	Drama	6,592	7,031	1.07
9	Fantasy	737	12,886	17.48
10	Film-Noir	151	13,472	89.22
11	Horror	1,453	12,170	8.38
12	Musical	509	13,114	25.76
13	Mystery	754	12,869	17.07
14	Romance	2,003	11,620	5.80
15	Sci-Fi	938	12,685	13.52
16	Thriller	2,233	11,390	5.10
17	War	543	13,080	24.09
18	Western	323	13,300	41.18
Avg.		**1,505.5**	**12,118**	**18.65**

3 DATASET DESCRIPTION

3.1 Provided content descriptors

Apart from the from the movie trailers (which are provided via links), MMTF-14K comes with precomputed state-of-the-art features, addressing three modalities: metadata (textual), audio and visual.

3.1.1 *Metadata descriptors.*

Two types of metadata descriptors are provided with MMTF-14K: (i) genre features as editorial metadata, and (ii) tag features to serve as user-generated metadata. Additionally, we provide the year of production for the movies in MMTF-14K as a side contribution. The metadata originally belong to the ML dataset which provide them in textual form (see Table 3). Nevertheless, in MMTF-14K these data are preprocessed and prepared as ready-to-use numerical feature vectors. The advantages of releasing metadata are multi-fold: first, they can be used in building CB or Hybrid RS as features describing items' content. For example, metadata can be used in conjunction with multimedia features using a variety of fusion or hybridization techniques; ultimately, they serve as baselines for comparing the recommendation quality with other systems.

Genre features: are represented by a 18-dimensional binary vector for each movie trailer, representing each of the 18 annotated movie categories: *Action*, *Adventure*, *Animation*, *Children*, *Comedy*, *Crime*, *Documentary*, *Drama*, *Fantasy*, *Film-Noir*, *Horror*, *Musical*, *Mystery*, *Romance*, *Sci-Fi*, *Thriller*, *War* and *Western*. The distribution of genre labels in MMTF-14K is shown in Table 2. The class imbalance for each genre label is defined by the skewness ratio, given by: $Skewness = \frac{\text{negative examples}}{\text{positive examples}}$. If it is intended to use the genre labels provided by MMTF-14K for genre classification task, this classification task is in essence a *multi-label classification problem*. In such a condition, knowledge of skewness is fundamental since in multi-label classification approaches such as one-vs-rest, the class imbalance can substantially influence the performance of classifiers. In other words, for such datasets adoption of conventional classifiers and/or evaluation metrics (*e.g.,* accuracy) may not provide realistic picture of the overall classification quality [26].

Tag features: are based on a term-frequency inverse-document-frequency (tf-idf) Bag-of-Word (BoW) model. A preprocessing stage is added to the process of generating the final movie-level descriptor, involving the following operations: *(a) punctuation removal*, *(b) tokenizing and lower-case conversion*, *(c) word removal* for words with very high or very low frequency, *(d) stop word removal* and finally *(e) Porter stemming* [21]. After these steps, each tag feature is represented by a decimal vector of length 10,228. Note that only 9,646 of the movies were assigned tags by users. This happens in cold-start (CS) situations when a new item is added to the catalog and no metadata is assigned to it [2].

Year of Production: In addition to the above metadata, we release also the year of production for movies. They were automatically obtained by text processing of the ML dataset and extracting the years embedded in the title of movies. For a few movies, this information was not available and was added manually. The average, median and standard deviation of these values are: 1992.2, 1999 and 84.84, respectively.

3.1.2 *Audio descriptors.*

Two sets of audio features are provided, representing both traditional descriptors (Block-level features) as well as current state-of-the-art (I-vectors).

Block-level features (BLFs): extracts features from audio segments of a few seconds, in contrast to frame-level features which operate on much shorter units. BLFs capture temporal aspects of an audio recording to some degree. The block-level feature framework [25] defines six features that capture: *spectral aspects* (spectral pattern, delta spectral pattern, variance delta spectral pattern), *harmonic aspects* (correlation pattern), *rhythmic aspects* (logarithmic fluctuation pattern), and *tonal aspects* (spectral contrast pattern).

I-vector features: this paradigm [7] is the current state-of-the-art representation learning technique in different audio-related domains, such as speech processing, music recommendation, and acoustic scene analysis [11, 27]. An i-vector is a fixed-length and low-dimensional representation containing rich acoustic information, which is usually extracted from short to moderate segments (usually from 10 seconds to 5 minutes) of acoustic signals. The i-vector features are computed using Mel-Frequency Cepstral Coefficients (MFCCs) frame-level features.

3.1.3 *Visual descriptors.*

Similar to the audio descriptors, for the visual modality we provide both a traditional approach (Aesthetic Visual Features) and state-of-the-art descriptors that use deep neural networks (AlexNet Features).

Aesthetic Visual Features: this set of features has been proposed in [13], where they have been used for measuring the aesthetic value of coral reef pictures, while parts of these feature set are inspired from works dealing with artwork and photographic aesthetics [6, 16, 20]. This set is composed of 26 features, split into three main categories: *color based descriptors*, *texture based descriptors* and *object based descriptors*. Three early fusion schemes are presented in our collection for recommendation purposes: individual features, feature fusion according to the three main categories and full fusion with all the features. We also employ four aggregation schemes for obtaining movie level features: average, median, average + variance and median + median absolute deviation.

AlexNet features: the AlexNet [19] deep neural network has been developed for scene and object recognition tasks. In our context, we use the extracted output values of the fc7 layer, as it has been shown to give good performances in a high number of tasks, some of which are related to human centered preference systems such as interestingness [10] and aesthetic ranking [17]. The same four statistical aggregation schemes as for the aesthetic visual features are employed here.

3.2 Dataset basic statistics

The dataset consists of 13,623 movie trailers, with an average duration of approximately 2 minutes and 22 seconds, for a total of almost 23 days of video data. 138,492 users gave 12,471,739 ratings to these trailers, thus each user rates an average of 90 videos and each video has being rated on average 915.5 times as shown in Table 3. As it can be seen, statistical distribution of ratings in MMTF-14K and ML-20M represented by the density of URM $density = \frac{|\mathcal{R}|}{|\mathcal{I}|\times|\mathcal{U}|}$ are similar (density = 1- sparsity).

3.3 Data format

The MMTF-14K dataset can be downloaded from this link[7]. The data is organized in 4 folders, one of them containing general information regarding the dataset (*Data*), while the other 3 (*Metadata descriptors*, *Audio descriptors* and *Visual descriptors*) have the preprocessed features under different aggregation schemes. All the comma-separated values (.csv) files are encoded in simple UTF-8 format, while archive files are in standard ZIP format.

3.3.1 *Data.*

The *Data* folder contains two files that offer information regarding the dataset: *movie_description.csv* and *rating.txt*. The first file gives details about the trailers in this dataset, with the first column indicating the movie id, the second indicating the full title of the movie trailer while the last one representing the preferred trailer link (YouTube identifier)[8] where the trailers can be accessed. The second file gives general information regarding the way the movie ratings were obtained, along with a download link for the corresponding movie ratings files (*i.e.,* ML-20M)[9]. For recommendation tasks, the ratings represent the ground truth data that can be used with this dataset for movie recommendation purposes.

3.3.2 *Metadata descriptors.*

The *Metadata descriptors* folder contains three subfolders: *Genre features*, *Tag features* and *Year of Production*. All the metadata features are obtained from Movie Lens. The *Genre features* folder contains the *GenreFeatures.csv* file with every row representing a movie. The first column of this csv file represents the id of the movie trailer corresponding the the ML movie ids, while the rest of the columns represent the binary values of the genre feature vector. The *Tag features* folder has a similar structure, containing a *TagFeatures.csv* file, where movie trailers are represented as different rows. Again, the first column is the id of the movie trailer and the tag feature vector is contained in the rest of the columns.

3.3.3 *Audio descriptors.*

The *Audio descriptors* folder contains two sub-folders: *Block level features* and *I-Vector features.* While the Block-level features include different fusion schemes, the I-Vector features include different parameters for the Gaussian mixture model (GMM) and total variability dimension (tvDim). The *Block level features* folder has two sub-folders: *All* and *Component6*, and while the former contains precomputed similarities using all 6 subcomponents, the latter contains each of the 6 components in separate csv files. Note that for *All*, for ease of use, we preferred to provide the precomputed similarities. For space requirements, similarities are provided in two files, one containing pair-wise similarities and the other one the corresponding movieIds for each column of the similarity matrix.

The *Component6* folder contains 6 .csv files, each representing a component of the BLF vector (*e.g., BlockFeatures - Component6 - Spectral.csv*, *BlockFeatures - Component6 - SpectralContrast.csv*) with a similar structure as the previous file. The *I-Vector features* folder includes the 180 files, corresponding to the all combinations of the parameters used (e.g.: *IVectorFeatures - GMM_tvDim_fold - 16_10_1.csv*, *IVectorFeatures - GMM_tvDim_fold - 512_400_5.csv*) where the first number of the title of the files represents the number of mixture models of the GMM (16, 32, 64, 128, 256 and 512), the second the tvDim (10, 20, 40, 100, 200 and 400) and the last one the fold number (1, 2, 3, 4 and 5). Extraction of i-vectors requires building an acoustic space from the audio signals of the item in the train phase which is used to learn/extract i-vector features from each item in a subsequent stage. Since this task is dataset-dependent, in the folder Data we provide an additional folder *rating-splitted-5foldCV* which contains information regarding user and items used in the cross-validation (needed both for i-vector extraction and replicating the recommendation experiment).

3.3.4 *Visual descriptors.*

The *Visual descriptors* folder contains two subfolders: *Aesthetic features* and *AlexNet features*, each of them including different aggregation schemes for the two types of visual features. The *Aesthetic features* folder includes 4 subfolders, corresponding to the 4 aggregation schemes: *Avg* containing the average aggregation scheme, *AvgVar* with the average and variance aggregation scheme, *Med* containing the median scheme and finally *MedMad* with the median and median absolute deviation aggregation. Each of these folders contain 30 .csv files, representing the different early fusion schemes applied to these features: individual components (*i.e., AestheticFeatures - MED - Feat26Convexity*, *AestheticFeatures - AVG - Feat26Edge*), early fusion based on the 3 main types (*i.e., AestheticFeatures - MEDMAD - Type3Color.csv*, *AestheticFeatures - AVG - Type3Texture.csv*) and finally a vector containing all the component concatenated (*i.e., AestheticFeatures - MED - All.csv*). The *AlexNet features* folder has a similar structure, containing the 4 subfolders, each of them corresponding to a different aggregation scheme: *Avg, AvgVar, Med* and *MedMad*. The structure of these archives is simpler than the case for the AVF features, considering that no early fusion scheme is needed or applicable to the fc7 layer output. Therefore, only one file will be

[7] MMTF-14K dataset is available for download at: https://mmprj.github.io/mtrm_dataset/index or https://zenodo.org/record/1225406

[8] The full link in order to access the trailers is created by: https://www.youtube.com/watch?v= + YouTube identifier

[9] http://www.movielens.org

Table 3: Characteristics of the user-rating matrix associated with MMTF-14K and ML-20M: $|\mathcal{U}|$ — number of users, $|\mathcal{I}|$ — number of items, $|\mathcal{R}|$ — number of ratings. (Note that *w.r.t.* Table 2 $|\mathcal{R}| = |\mathcal{P}|$)

dataset	$\|\mathcal{U}\|$	$\|\mathcal{I}\|$	$\|\mathcal{R}\|$	$\frac{\|\mathcal{R}\|}{\|\mathcal{U}\|}$	$\frac{\|\mathcal{R}\|}{\|\mathcal{I}\|}$	$\frac{\|\mathcal{R}\|}{\|\mathcal{I}\|\times\|\mathcal{U}\|}$ (density)
MMTF-14K	138,492	13,623	12,471,739	90.05	915.5	0.0066
ML-20M	138,493	26,744	20,000,263	144.4	747.8	0.0054

present in these folders, depending on the aggregation scheme (*i.e., AlexNetFeatures - MED - fc7*).

4 GROUND TRUTH

Since the proposed MMTF-14K dataset is mainly meant for movie recommendations task but can also address a broader audience in machine learning and multimedia domain, the ground truth here is the actual rating scores provided by the user to the movies.

The ground truth associated to the data was extracted from the MovieLens 20M dataset [14], also called ML v4. This released version of the dataset consists of ratings sampled throughout a large portion of the history of the ML initiative, more precisely from January 1995 to March 2015. The rating system is a "half star" system, moving away from a "whole star" only system in 2003, as a result of user demand in some surveys, thus granting users the permission to choose from 10 preference scores (0.5 to 5). Users also participated in the creation of the original tag features assigned to each movie, a function that was added to ML in December 2005. In what concerns the provided ratings, there are fewer ratings in the "half star" categories than in the "whole star" ones, most likely due to the later introduction of the 10 score system. Secondly, the distribution of ratings counts per user (*i.e.,* number of ratings given by each user to movies in the catalog) and item (*i.e.,* number of ratings given to each item by the users) is shown in Figure 1. As it can be seen, the pick of rating counts per user lies in the region 11-30 rating, with an average number of ratings equal to 90.05 and a maximum and minimum equal to 4,873 and 2 ratings respectively (std: 139.14). This is while for per-item rating score, majority of items have less than 10 rating scores. For example, 3,265 movies have between 1-3 ratings. The average number of ratings per item is 915.5 while the maximum and minimum equal to 63,366 and 1 rating(s) (std: 3,434.9). As it can be noted there is a large standard deviation/sharp discontinuity between two types of items, the items in the left side of the orange curve (which attract a fair number of rating) and second the popular items (the items in the right side of orange curve which attract a large number of ratings per user). Knowledge of popularity is important because some recommendation algorithms such as collaborative filtering (CF) [2, 22] base their recommendation on the ratings obtained by community of users (instead of content descriptions as in CB) thereby promoting the chance of these popular items being more recommended. Thus, in some tasks a portion of popular items are removed from the recommendation process in order to obtain a realistic picture of the overall recommendation quality of the system[10].

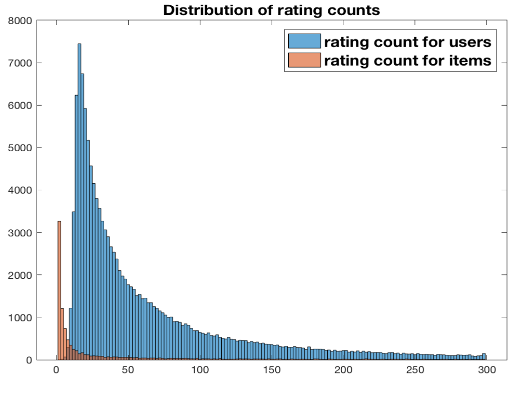

Figure 1: Distribution of rating counts for users and items (x-axis: number of ratings, y-axis: number of users or items). Note that for simplicity we show the plot for rating count equal to 300.

5 BASELINE AND REFERENCE RESULTS

To provide a baseline for recommender system experiments, we randomly chose a subset of 3,000 users, with the condition that each user has a minimum of 50 movie ratings in their profile, and performed a 5-fold cross validation experiment by creating 5 non-overlapping segments. *Mean Reciprocal Rank* (MRR), *Mean average precision* (MAP) and *Recall* (R) are then calculated for different cutoff values (@4 and @10) and in two different scenarios: warm-start (WS) and cold-start (CS). While the WS scenario takes into account all the tag features, the CS scenario keeps all the tag features for training the system, while on the test set only a random selection of 3% of the tag features are kept. The cold-start scenario is supposed to simulate real-world conditions, by acknowledging the fact that some movies, especially the newer or less popular ones, have a small set of user input data, therefore have fewer tags attached to them. The corresponding code for calculating the MRR, MAP and R values is available with the dataset.

The results for each default extracted descriptor are presented in Table 4, along with the results for the best performing late fusion combinations of these features. The CBF is based on standard k-nearest neighboring approach [12]. The fusion scheme is based on the Bodra count method [3] which fuses ranking results of different recommenders into a unified ranking of videos. As it can be seen the SoA i-vec (audio) and Deep AlexNet (visual) have supervisor performance compared with traditional BLF (audio) and AVF (visual) descriptors. It can be also noted that while both SoA audio and visual descriptors have a higher quality compared with

[10] Note that in RS community, popularity is measured by the number of ratings assigned to items (*e.g.,* movies)

Table 4: Baseline performance for movie recommendation. Best results are in bold. CS: cold-start, WS: warm-start

feature name	modality	MRR@4	MAP@4	R@4	MRR@10	MAP@10	R@10
tag (CS)	M	0.0195	0.0051	0.0042	0.0274	0.0037	0.0111
tag (WS)	M	0.0213	0.0057	0.0046	0.0294	0.0041	**0.0120**
genre	M	0.0162	0.0044	0.0039	0.0245	0.0034	0.0112
i-vec	A	**0.0233**	**0.0060**	**0.0052**	**0.0311**	**0.0042**	**0.0120**
BLF	A	0.0170	0.0045	0.0038	0.0242	0.0032	0.0097
AlexNet	V	0.0219	0.0057	0.0043	0.0296	0.0038	0.0111
AVF	V	0.0187	0.0049	0.0039	0.0263	0.0034	0.0102
i-vec + AlexNet	A + V	0.0232	0.0061	0.0051	0.0318	0.0043	0.0122
AlexNet + tag	V + M	0.0239	0.0062	0.0053	0.0325	0.0044	0.0130
i-vec + tag	A + M	**0.0266**	**0.0072**	**0.0059**	**0.0359**	**0.0049**	**0.0139**

the genre recommender, the i-vec has also a superior performance compared to semantic-rich tag (in both CS WS) with regards to all evaluation metrics and all cut-off values.

6 CONCLUSIONS

In this work we release the MMTF-14K dataset, a dataset consisting of 13,623 Hollywood-like movie trailers which are rated by more than 138,492 users. The primary scope of this dataset is to support the development of movie recommender systems, and to the best of our knowledge, this is the first large-scale dataset in the recommender systems community that provides all types of content-based descriptors in conjunction with metadata. However, these data go beyond the recommending scenario thanks to its rich content. It can also be used for tasks such as popularity prediction, tag prediction and genre classification. Apart from the data, we are also releasing some baseline results to allow further benchmarking. The data is publicly available.

REFERENCES

[1] [n. d.]. Yahoo!: Webscope movie data set (Version 1.0). http://research.yahoo.com/. ([n. d.]). Accessed: 2018-03-01.

[2] Charu C Aggarwal. 2016. An Introduction to Recommender Systems. In Recommender Systems. Springer, 1–28.

[3] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. In Proceedings of the fourth ACM conf. on Recommender systems. ACM, 119–126.

[4] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In ISMIR, Vol. 2. 10.

[5] David Bordwell, Kristin Thompson, and Jeff Smith. 1997. Film art: An introduction. Vol. 7. McGraw-Hill New York.

[6] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. 2006. Studying aesthetics in photographic images using a computational approach. In European Conference on Computer Vision. Springer, 288–301.

[7] Najim Dehak, Patrick J Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2011. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19, 4 (2011), 788–798.

[8] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, and Pietro Piazzolla. 2016. Recommending movies based on mise-en-scene design. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM, 1540–1547.

[9] Yashar Deldjoo, Mehdi Elahi, Paolo Cremonesi, Franca Garzotto, Pietro Piazzolla, and Massimo Quadrana. 2016. Content-based video recommendation system based on stylistic visual features. Journal on Data Semantics 5, 2 (2016), 99–113.

[10] Claire-Hélène Demarty, Mats Viktor Sjöberg, Bogdan Ionescu, Thanh-Toan Do, Hanli Wang, Ngoc QK Duong, Frédéric Lefebvre, et al. 2016. Mediaeval 2016 predicting media interestingness task. In MediaEval 2016 Multimedia Benchmark Workshop Working Notes Proceedings of the MediaEval 2016 Workshop.

[11] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer. 2016. CP-JKU Submissions for DCASE-2016: a Hybrid Approach Using Binaural I-Vectors and Deep CNNs. Technical Report. DCASE2016 Challenge.

[12] Mehdi Elahi, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Leonardo Cella, Stefano Cereda, and Paolo Cremonesi. 2017. Exploring the Semantic Gap for Movie Recommendations. In Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, 326–330.

[13] Andreas F Haas, Marine Guibert, Anja Foerschner, Sandi Calhoun, Emma George, Mark Hatay, Elizabeth Dinsdale, Stuart A Sandin, Jennifer E Smith, Mark JA Vermeij, et al. 2015. Can we measure beauty? Computational evaluation of coral reef aesthetics. PeerJ 3 (2015), e1390.

[14] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4 (2016), 19.

[15] David Hauger, Markus Schedl, Andrej Košir, and Marko Tkalcic. 2013. The million musical tweets dataset: What can we learn from microblogs. In Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013).

[16] Yan Ke, Xiaoou Tang, and Feng Jing. 2006. The design of high-level features for photo quality assessment. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 1. IEEE, 419–426.

[17] Shu Kong, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. 2016. Photo aesthetics ranking network with attributes and content adaptation. In European Conference on Computer Vision. Springer, 662–679.

[18] Andrej Košir, Ante Odic, Matevz Kunaver, Marko Tkalcic, and Jurij F Tasic. 2011. Database for contextual personalization. (2011).

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. 1097–1105.

[20] Congcong Li and Tsuhan Chen. 2009. Aesthetic visual quality assessment of paintings. IEEE Journal of selected topics in Signal Processing 3, 2 (2009), 236–252.

[21] Martin F Porter. 1980. An algorithm for suffix stripping. Program 14, 3 (1980), 130–137.

[22] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In Recommender systems handbook. Springer, 1–34.

[23] Markus Schedl. 2016. The LFM-1b Dataset for Music Retrieval and Recommendation. In Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR). New York, USA.

[24] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2018. Current Challenges and Visions in Music Recommender Systems Research. International Journal of Multimedia Information Retrieval (2018), 1–22.

[25] Klaus Seyerlehner, Gerhard Widmer, Markus Schedl, and Peter Knees. 2010. Automatic Music Tag Classification based on Block-Level Features. In Proceedings of the 7th Sound and Music Computing conf. (SMC 2010). Barcelona, Spain.

[26] Lei Tang, Suju Rajan, and Vijay K Narayanan. 2009. Large scale multi-label classification via metalabeler. In Proceedings of the 18th international conference on World wide web. ACM, 211–220.

[27] Andreu Vall, Hamid Eghbal-zadeh, Matthias Dorfer, Markus Schedl, and Gerhard Widmer. 2017. Music Playlist Continuation by Learning from Hand-Curated Examples and Song Features: Alleviating the Cold-Start Problem for Rare and Out-of-Set Songs. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems. ACM, 46–54.

[28] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. 2014. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, 73–82.