Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-10086-2

A multimodal approach for multi-label movie genre classification

Rafael B. Mangolin[1] · Rodolfo M. Pereira[2,3] · Alceu S. Britto Jr.[3] · Carlos N. Silla Jr.[3] · Valéria D. Feltrim[1] · Diego Bertolini[4] · Yandre M. G. Costa[1]

Received: 12 March 2020 / Revised: 14 September 2020 / Accepted: 15 October 2020 /
Published online: 07 November 2020

Abstract

Movie genre classification is a challenging task that has increasingly attracted the attention of researchers. The number of movie consumers interested in taking advantage of automatic movie genre classification is overgrowing, thanks to media streaming service providers' popularization. In this paper, we addressed the multi-label classification of movie genres in a multimodal way. To this end, we created a dataset composed of trailer video clips, subtitles, synopses, and movie posters from 152,622 movie titles of the Movie Database (TMDb). Such a large dataset was carefully curated, organized, and made available as a contribution of this work. We labeled each movie of the dataset according to a set of eighteen genre labels. In the experimental evaluation performed in this paper, we computed different kinds of descriptors, such as Mel Frequency Cepstral Coefficients (MFCCs), Statistical Spectrum Descriptor (SSD), Local Binary Pattern (LBP) from spectrograms, Long-Short Term Memory (LSTM), and Convolutional Neural Networks (CNN). With these descriptors, we trained different monolithic classifiers using BinaryRelevance and ML-kNN techniques. Besides, we also explored the combination of classifiers/features using a late fusion strategy. The fusion of a LSTM trained on synopses and another LSTM trained on the movie subtitles provided our best results in F-Score (0.674) and AUC-PR (0.725) metrics. These results corroborate the existence of complementarity among classifiers trained on different sources of information in this field of application. As far as we know, this is the most comprehensive study developed in terms of diversity of multimedia sources of information to perform movie genre classification.

Keywords Movie genre classification · Multi-label classification · Multimodal classification · Movie trailer

✉ Rafael B. Mangolin
rbmangolin@gmail.com

Extended author information available on the last page of the article.

1 Introduction

Streaming media services have grown steadily over the past decade, mainly due to the consolidation of video on demand as a practical and comfortable way of allowing consumers to access films, series, documentaries, etc. Some giant companies (e.g., Netflix™, Hulu™, Amazon™prime video, YouTube™, and Facebook™watching)[1] are rapidly gaining ground in this market, as they offer exclusive content through agreements with the movie industry and integrate other products and services that serve consumers' interests more comprehensively.

In this study, we addressed the movie genre classification using multimodal classifiers based on audio and images from trailers (i.e., audio and video frames), subtitles, posters, and synopses in a multi-label scenario. First, we obtained representations from each modality using handcrafted and non-handcrafted (i.e., obtained using representation learning) features, totaling 22 descriptors. Then, we trained multi-label classifiers for each representation using the algorithms Binary Relevance and ML-kNN. The predictions provided by the classifiers were combined using different late-fusion strategies (sum rule, product rule, and max rule). We also experimented with compressive sampling, which is a promising dimensionality reduction strategy that was used with some large representations created from textual data.

To our knowledge, this is the first study to combine all these data sources for movie genre classification. Cascante-Bonilla et al. [7] have presented a dataset that includes the same sources of information used here, but they did not experiment with all of them.

We carefully curated and organized a dataset to evaluate our proposal from movie titles taken from The Movie Database (TMDb). Such a large dataset contains 10,594 movies labeled in 18 genres. For each film, we collected the following: video trailer, poster, synopsis, and subtitle. The movies' trailers and subtitles were collected from YouTube™and OpenSubtitles, respectively.

The resulting dataset was used in our experimental protocol and made available to the community,[2] allowing other researchers to evaluate other protocols and properly compare their results with those presented here.

The main contributions of this paper are:

- a new multimodal database for movie genre classification freely available to the community;
- a multimodal approach to perform the multi-label genre classification;
- the evaluation of different representations, including hand-crafted and deep learning techniques;
- the evaluation of late fusion techniques to combine models trained on diverse data sources.

The paper is organized into seven sections. Section 2 summarizes some of the related works regarding movie genre classification and multimodal multimedia classification. In Section 3, we describe our dataset, while in Section 4, we present our multimodal approach. Experimental results are presented in Section 5 and discussed in Section 6. Finally, in Section 7, we conclude and suggest directions for future works.

[1] https://www.tapjoy.com/resources/video-streaming-industry-growth/

[2] https://drive.google.com/file/d/1X9siuhXAsCdQljDjjs9PbqWXQ5Z72FOV/view?usp=sharing

2 Related works

Several researchers have devoted efforts towards the development of new approaches to movie genre classification. They have explored different kinds of data as sources of information, such as movie posters, subtitles, trailers, and synopses. The features captured from these sources feed classifiers induced by machine learning algorithms. In this section, we present some of these works in chronological order.

Brezeale and Cook [5] used bag-of-words (BOW) taken from closed captions and bag-of-visual-features (BOVF) taken from frames of video clips. They extracted BOW features after the removal of stop words [13] and stemming [40]. They computed a histogram for each movie using Discrete Cosine Transform (DCT) on each scene's first frames. The k-means algorithm generates BOVF from the histograms. With BOW and BOVF in hands, they trained SVM classifiers using a multi-label dataset with 81 movies from the MovieLens project labeled on 18 genres. The authors performed a binary classification creating an SVM for each genre. The best results in terms of mean accuracy were 89.71% using BOW and 88.48% using BOVF.

Zhou et al. [57] extracted features using GIST visual descriptor [36], CENTRIST [54], and W-CENTRIST. These descriptors emphasize the textural content of the frames (GIST), the intensity pattern between the pixels (CENTRIST), and the chrominance patterns between the pixels (W-CENTRIST). The used dataset contained 1,239 movie trailers classified into four genres, namely action, comedy, drama, and horror. K-means algorithm generated the BOVF considering a hundred centroids. The best accuracy was 74.7% using the CENTRIST features.

Huang and Wang [24] proposed the movie genre's multimodal classification using features extracted from the audio and video content (i.e., frames) of trailers. They classified the visual features into two categories, temporal (i.e., time-related) and spatial (i.e., frame content-related), totaling 75 features. The categorization of audio features considers audio intensity related, rhythm related, and timbre related. They used a total of 202 audio features, including the Mel-Frequency Cepstral Coefficients (MFCC). The Self-Adaptive Harmony Search (SAHS) algorithm [51] performed a feature selection step on each of the one-against-one SVMs employed in the classification process. The used dataset has 223 video trailers classified into seven genres (action, animation, comedy, documentary, drama, musical, and thriller), and the best-observed accuracy was 91.9%.

Hong and Hwang [23] also proposed a multimodal approach using movie trailer content (i.e., video, audio, and tags). The authors used two concepts taken from the Probabilistic Latent Semantic Analysis (PLSA) [22] to combine features extracted from the text (i.e., social tags), audio, and image. The dataset used in the experimental protocol was composed of 140 movie trailers distributed into four classes (action, biography, comedy, and horror), and social tags obtained via social websites. The proposed approach achieved an accuracy of 70% using the early fusion of audio, video, and text.

Fu et al. [14] combined features extracted from movie posters and synopses. The features extracted from the posters are related to color, texture, shape, and the number of faces depicted in the image. From the synopses, they computed BOW features after preprocessing the texts to stop word removal and stemming [40]. Their dataset contained synopses and posters from 2,400 movies taken from TMDb[3] classified into four genres (action, comedy,

[3] https://www.themoviedb.org/

romance, and horror). The authors considered two SVM classifiers, one for posters, and one for synopses. The best accuracy was 88.5% when both classifiers were fused.

Simões et al. [46] used deep learning to classify movie genres. The authors extracted features from each trailer frame using a CNN. Then, they computed a feature vector with the average values of its frames for each trailer scene and submitted it to the k-means algorithm to generate a BOVF. Audio features were also extracted from the trailers using MFCC. Their experiments use the LMTD-4 dataset, a subset taken from the Labeled Movie Trailer Data (LMTD). It comprises 1,067 trailers classified into four genres (action, comedy, drama, and horror). The best accuracy rate was 73.75% using SVM with BOVF features, audio features, and the CNN's weighted prediction.

Wehrmann and Barros [52] extracted audio and video features from movie trailers to perform multi-label genre classification. The authors used CNN with the Convolution Through-Time (CTT) module, which organizes the features taken from each frame considering their respective positions on the sequence as a whole. The authors proposed five different models: three based on video content (CTT-MMC-A, CTT-MMC-B, and CTT-MMC-C); one based on audio content (CTT-MMC-S); and one that fused the results of the networks trained using audio with the results of the networks using video (CTT-MMC-TN). They conducted experiments on a subset of the LMTD dataset, LMTD-9, which contains trailers from 4,007 movies, labeled into nine genres. The best result was obtained by the CTT-MMC-TN model, achieving an AUC-PR (Area Under Precision-Recall Curve) of 74.2%.

Portolese and Feltrim [42] performed multi-label movie genre classification based on features extracted from synopses. The authors evaluated 19 feature sets based on Term Frequency-Inverse Document Frequency (TF-IDF) features and different models of the word and document embeddings. The dataset used in the experiments was composed of 12,094 synopses written in Brazilian Portuguese collected from TMDb[4] and labeled into 12 genres. They constructed four different classifiers (Multilayer Perceptron (MLP), Decision Tree, Random Forest, and Extra Trees). The best result was 54.8% (f-score) using an MLP classifier trained on TF-IDF features considering 3-grams with dimensionality equal to 1,000. In Portolese et al. [41], the authors extended their dataset to 13,394 Portuguese synopses labeled in 18 genres, and the groups of textual features. They also experimented with an oversampled version of the dataset. The best result for the original dataset was 0.478 (f1-score) using a TF-IDF based classifier. The best result for the oversampled dataset was 0,611 (f1-score) combining several feature groups.

Cascante-Bonilla et al. [7] carried out a large-scale investigation comparing the effectiveness of visual, audio, text, and metadata-based features for the prediction of high-level information about movies, including genre classification. The authors used a multimodal dataset composed of 5,027 films classified with 13 labels and included trailer, text plot, poster, and other metadata. They concluded that the use of features obtained from the text (i.e., fastText) and video (i.e., fastVideo) are better suited for a holistic classification task than the well-known LSTM-based representation. They achieved a mean average precision of 68.6% performing late fusion.

Oramas et al. [37] used deep learning with multimodal data for music genre classification. Despite the difference in application, they also employed deep learning techniques in a multimodal setting. They used deep architectures to learn features from audio tracks, text reviews, and cover art images. Their results showed that combining features from

[4] https://www.themoviedb.org/

different modalities improves performance compared to the modalities in isolation, indicating complementarity.

Table 1 presents a summary of the studies described in this section, focusing on essential characteristics of the works related to the task investigated here (i.e., Author/year, source of information, content descriptor, ML technique, and recognition rate).

3 Dataset

We collected the data from *The Movie Database* (TMDb), a database created in 2008 that contains freely available data regarding movies and TV series from all over the world. Among the most remarkable aspects related to the TMDb, we highlight: i) the different data sources (i.e. synopsis, poster, trailer, and subtitle) are available for each movie title; and ii) that movies are labeled with one or more genres (multi-label).

Table 1 Summary with the main characteristics of the related works

Author/year	Source of information	Content descriptor	ML technique	Recognition rate
Brezeale and Cook [5]*	Subtitles and movies	BOW and BOVF	SVM	89.71%[1]
Zhou et al. [57]*	Frames	GIST, CENTRIST and W-CENTRIST	*k-means*	74.7%[1]
Huang and Wang [24]	Trailers	Structural video descriptors, MPEG-7, RMS, MFCC and LPC	SAHS and SVM	91.9%[1]
Hong and Hwang [23]	Tags and trailers	BOW, BOAF and BOVF	PLSA	70%[1]
Fu et al. [14]	Posters and synopsis	BOW and visual descriptors	SVM	88.5%[1]
Simões et al. [46]	Trailers	BOVF and MFCC	SVM	73.75%[1]
Wehrmann and Barros [52]*	Trailers	CNN	CNN	72.4% [2]
Portolese and Feltrim [42]*	Synopsis	TF-IDF and *word embeddings*	Decision trees and MLP	54.8%[3]
Cascante-Bonilla et al. [7]*	Trailers, movie plots, posters and metadata	CNN and LSTM	CNN and LSTM	68.6%[4]
Portolese et al. [41]*	Synopsis	Several textual features	Binary Relevance with Decision trees	61.1%[3]

[1] Accuracy

[2] AUC-PR

[3] F-Score

[4] mAP

* multi-label classification

Our dataset was inspired by the one described in Portolese et al. [41]. We used their subset of titles as a starting point but focused on retrieving the textual data (i.e. synopsis and subtitle) in English rather than in Portuguese. Besides textual data, we also retrieved the movies' poster and trailer. As we considered only titles with all the data available, our final dataset is composed of 10,594 movie titles containing for each sample: the poster (in image format), the subtitle (in text format), the synopsis (in text format), and the movie trailer clips (which allows the use of both audio and video content, as discussed in Sections 4.2 and 4.1, respectively). We finished all data collection in July 2019.

We followed a two-step protocol to collect the subtitles. First, we retrieved the name, debut year, and ID of the movie in the IMDB[5] website using the TMDb API[6]. Then, we obtained the subtitle from the OpenSubtitles[7] website, either using the IMDB ID, when available, or searching for the movie's title and debut year. With this protocol, we collected the subtitles for 11,066 movies.

We used the TMDb API to collect the movies' posters and synopses. We retrieved posters for 13,047 movies and synopsis for 12,320 movies.

We collected the movies' trailers from YouTube.[8] To identify which video was related to which movie, we used two strategies: if the URL of the movie's trailer on YouTube™was available on TMDb, we used it to download the video; if not, we used the movie's title and debut year to search its trailer using the YouTube API. For each movie searched, we chose the first returned result with a duration length between 30 and 300 seconds. We used a Python library to download the trailers for 13,390 movies. After collecting all the data, we selected only the movie titles with all four data sources (i.e., English subtitles, English synopsis, Poster, and trailer).

As in Portolese et al. [41], the movies in our dataset are labeled with 18 different genre labels, namely: Action, Adventure, Animation, Comedy, Crime, Documentary, Drama, Family, Fantasy, History, Horror, Music, Mystery, Romance, Science Fiction, TV Movie, Thriller, and War. The number of labels assigned to a movie title ranges from one to seven. The most frequent label is Drama, assigned to 5,122 movies, and the least frequent label is TV Movie, which appears in only 119 movies. Figure 1 shows the total amount of movie titles assigned to each genre label. The movie titles in the dataset were launched from 1902 to 2018. The average length of the trailer clips is 2 minutes and 4 seconds, and the average number of words synopsis and subtitles is 51 and 7,276, respectively.

Figure 2 shows the labels' co-occurrence matrix for our dataset. As can be observed, Drama is the genre that more frequently co-occurs with other genres, especially Romance, Thriller, and Comedy. On the other hand, Documentary and TV Movie are the genres with less co-occurrence with others, which can be explained by the fact that they are assigned to only a few movies.

According to Zhang and Zhou [56], other indicators that can be observed in a multi-label dataset are the following:

- Label Cardinality (LCard), which corresponds to the average number of labels per example.

[5] https://www.imdb.com/

[6] https://www.themoviedb.org

[7] https://www.opensubtitles.org

[8] https://www.youtube.com/

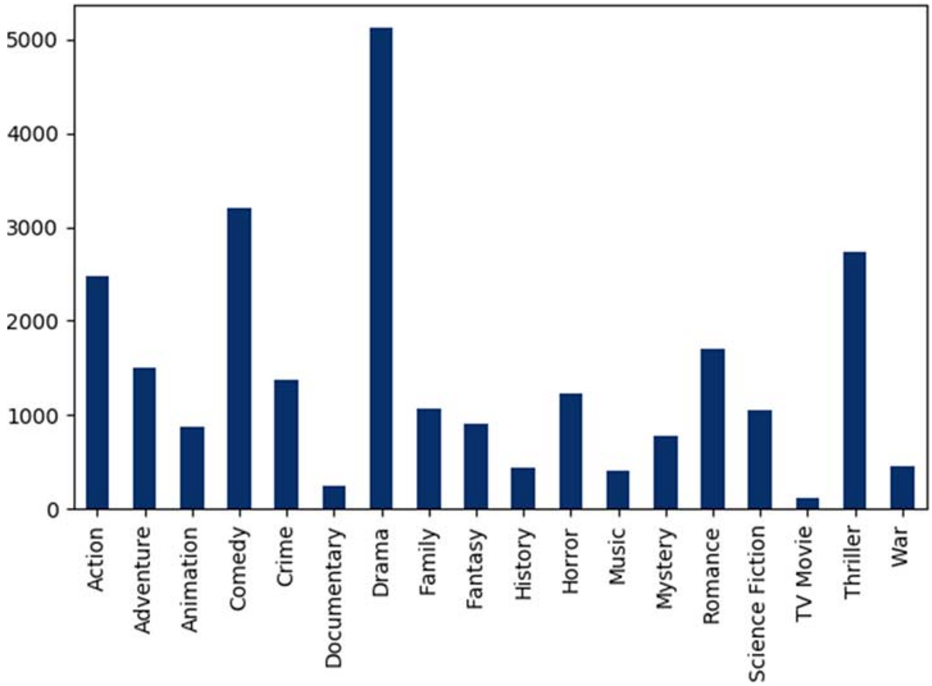

Fig. 1 Number of samples per genre

- Label Density (LDen), which is the normalization of label cardinality by the number of possible labels in the label space.
- Label Diversity (LDiv), which corresponds to the number of distinct label sets in the dataset.
- Proportion of Label Diversity (PLDiv), which is the normalization of Label Diversity by the number of examples, indicating the proportion of distinct label sets in the dataset.

Table 2 presents these indicators for our dataset. As we can observe, the average number of labels per movie (LCard) is 2.426, indicating that movies in the dataset are, in average, multi-label. However, considering that there are 18 possible labels in the label space, the density of the dataset (LDen), 0.134, is low. There are 922 distinct label sets (LDiv) in the dataset. As it has 10,594 examples, the proportion of label diversity (PLDiv), 0.087, is also low. From this, we can also infer that, on average, we have about 11.49 examples per distinct label set.

Table 2 Multi-label indicators extracted from the database created for this work

Metric	Value
LCard	2.426
LDen	0.134
LDiv	922
PLDiv	0.087

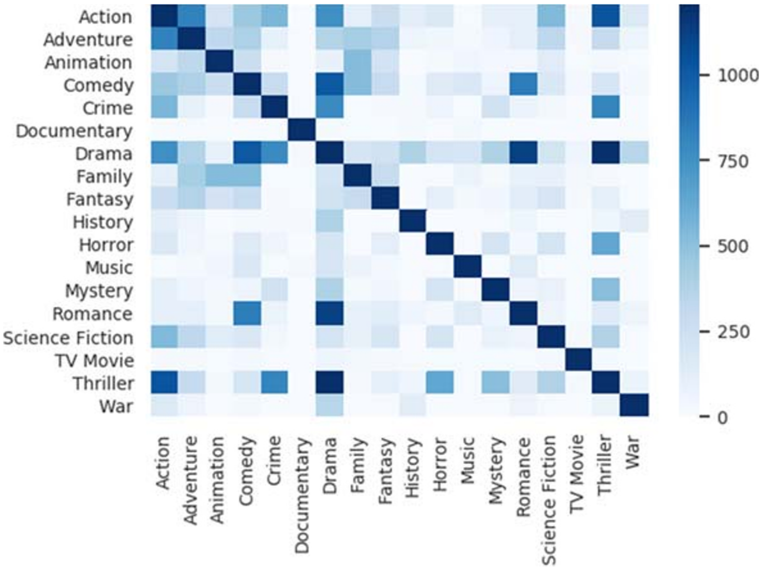

Fig. 2 Labels co-occurrence matrix

4 Proposed method

As aforementioned, our aim was to explore data from different modalities (i.e., audio, video/image, and text) to classify the genre of movie titles. Thus, we chose specific methods that could lead us to obtain the best benefit in terms of the classification performance of each of the data sources.

In Fig. 3, we present a general overview of our proposal, considering: the data source preparation (Phase 1), feature extraction (Phase 2), compression (only in case of huge representations) (Phase 3), resampling (Phase 4), classification (Phase 5), and fusion of the predictions (Phase 6). It is important to note that Phase 4 is faded in Fig. 3 because it is optional, since the original features without resampling can also be used to generate the predictions.

Phase 1 consisted of pre-processing the data for the resource extraction phase. It included cropping and resizing the trailers, cropping and padding the audio spectrograms, and removing time marks from synopses.

Phase 2 consisted of extracting features. The descriptors used to represent each type of data source are the following:

- Audio: Mel-Frequency Cepstral Coefficients (MFCC), Statistical Spectrum Descriptors (SSD) and Local Binary Pattern (LBP);
- Video frames: LBP, Convolutional 3D Neural Network (C3D), Inception-v3, Long-term Recurrent Convolutional Networks (LRCN), Convolution-Through-Time for Multi-label Movie genre Classification (CTT-MMC), and RGB Feature;
- Poster: LBP, Inception-v3 and RGB Feature;

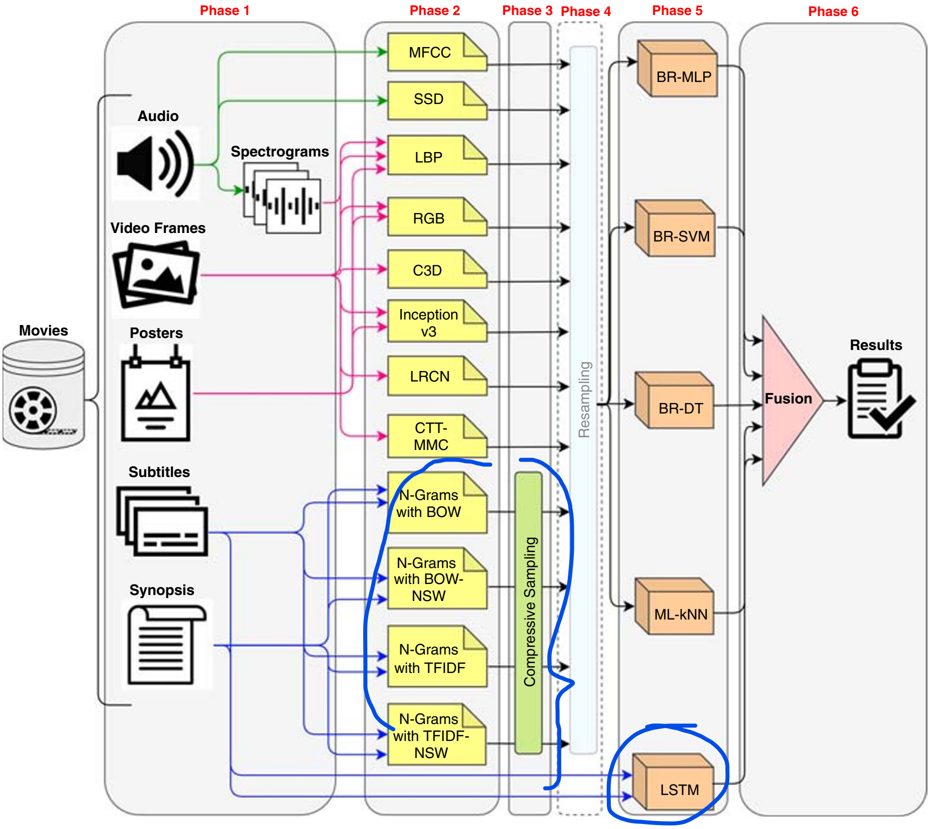

Fig. 3 General overview of features and classifiers used for multimodal multi-label classification. The phase circled with a dashed rounded rectangle is optional. The different colors of the arrows represent the different types of data source used in the feature extraction phase: Green for audio sources, pink for the image sources and blue for the textual sources

- Subtitles: Long Short-Term Memory (LSTM), and N-grams with Term Frequency–Inverse Document Frequency (TF-IDF); and
- Synopsis: LSTM and N-grams with TF-IDF.

As shown in Fig. 3, we used these descriptors with different classifier models that were later combined using a late fusion strategy. Sections 4.1 and 4.2 describe how we explored the image and audio contents, respectively, to build the representations. Section 4.3 describes the poster feature extraction, and Section 4.4 describes how we explored textual content (i.e. subtitle and synopsis). Section 4.5 presents a summary of the representations extracted with their respective identifiers, which are used throughout the text, and also describes the compressive sampling method, that we used to reduce the dimensionality of some representations obtained from the text. Section 4.6 describes the strategy used, as an optional step, to deal with the imbalance of the dataset. Finally, Section 4.7 describes the algorithms we used to infer the classifiers, and how we fused the predictions to get a final decision.

4.1 Feature extraction from trailer image (Phase 2)

To extract features from the selected frames of the movie trailer clips, we followed two different approaches. In the first approach, we used deep learning architectures, more specifically Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM). In the second approach, we experimented handcrafted features: we used different visual features and the $K-means$ clustering algorithm to generate the final set of features to describe the visual trailer content (based on BOVF).

Regarding the first approach, we used three techniques to extract features from the movie frames: C3D [49], CTT-MMC [52], and LRCN [12]. We chose these architectures mainly because they perform end-to-end feature extraction and classification of spatial-temporal data. In this way, these networks process only three-dimensional data (i.e., videos), and are not suitable for processing other data sources, such as the poster or audio. We designed the C3D architecture for video processing and classification. Thus, we equipped it with convolutional and pooling operations in three dimensions to accomplish the spatial and temporal feature extraction. The CTT-MMC architecture is composed of convolutional layers disposed in two dimensions that process the video sequence frame by frame carrying out the spatial feature extraction. We stored the features extracted from the frames in matrices according to the order of the frames in the video. Then, we processed these matrices using another convolutional layer that extracts the temporal features. The LRCN architecture is similar to CTT-MMC, but instead of having a convolutional layer to extract temporal features, it uses an LSTM network [21].

We selected a total of 120 frames 100×100 sized from each movie trailer and processed it using the deep architectures. We used the following criteria to perform the selection of the frames from each movie clip. First, we discarded the first and last 5% of the frames. We did this because these frames are often not discriminating, as their content is usually related to credits at the beginning and end of the video trailer. Then, we defined three equally distant points (i.e., start, middle, and end) and we selected one out of three frames from the previous 60 frames and the next 60 frames from each of those points, totaling 40 frames per point (start, middle and end), and a total of 120 frames of the trailer video as a whole.

In our second approach, we extracted features from the image content of the videos using descriptors successfully applied in other domains [9] and that capture different visual attributes of the images, namely LBP (textural content) [1] and RGB histogram (color content) [15]. In this case, we defined the number of frames used for feature extraction as the lowest number of frames contained in the video trailers, which is 555. Therefore, we selected 555 frames equally distant and linearly distributed from each movie trailer. Other details related to the extraction of handcrafted features from the selected frames are the following:

- Textural content: we converted the frames to grayscale images and we obtained a normalized 59-dimensional feature vector from each of them using uniform $LBP_{8,2}$ [1].
- Color content: we described the color content using a 768-dimensional feature vector created by appending the 256 sized histograms from each of the three channels (Red, Green, and Blue) of the RGB color system.

After the extraction of the handcrafted features, we submitted both datasets (i.e. textural and color feature vectors) to the $K-means$ algorithm. After some experiments varying the number of centroids from 128 to 1024, in a scale of powers of two, we chose to use 512 centroids for LBP and 1,024 centroids for RGB histogram. After assigning the frame vectors to the centroids, we grouped the frames from each movie and a histogram was calculated for each movie group. The size of this histogram was given by the number of centroids previously defined before running the $K - means$ algorithm. In addition, we normalized this histogram by the number of selected frames (i.e. 555). We used the final histogram for each group, i.e. each movie, as the descriptor of that movie in the classification step.

4.2 Feature extraction from audio (Phase 2)

The use of handcrafted features for the classification of audio content, captured based on a variety of descriptors, is widely present in the literature. Mel-Frequency Cepstral Coefficients (MFCC) [31], Statistical Spectrum Descriptors (SSD) [30], and Local Binary Pattern (LBP) [1] (i.e. a powerful descriptor for the textural content of images that, in the case of sounds, can be successfully used to capture the content of spectrograms created from the audio signal [9, 33, 34]) are among the most used descriptors for audio content. More recently, some studies have proposed the use of features learned from audio content by deep architectures [37, 39, 44]. In many of these works, a deep neural network is fed by spectrogram images obtained from the original audio signal.

We explored these two approaches for extracting features from the audio content: the first based on handcrafted features, and the second based on features learned using deep architectures fed by spectrogram images.

For the extraction of handcrafted features, we used SSD [30] and MFCC [31] descriptors as they are among the most successful descriptors for audio classification tasks. Based on the human hearing system, MFCC uses the cosine discrete Fourier transform (DFT) to capture 13 features that are suitable to describe the timbre of the sound. SSD, in turn, captures information regarding audio intensity variation, also aiming at properly representing the timbre of the sound. The SSD feature vector is composed of 168 features. We also used LBP to extract textural features from the spectrograms generated from the audio. The spectrograms were created using the SOX[9] library and they are also available as part of our dataset.

In the second approach, we explored the spectrograms using the Inception-v3, a CNN architecture. Although representation learning techniques had already been proposed by Bengio et al. [4], the use of these techniques was reinforced thanks to the use of transfer learning using IMAGENET [29]. In this sense, Szegedy et al. [48] proposed the Inception-v3. This architecture proved to be more robust than other architectures, presenting low error rates in the ILSVRC-2012 benchmark.[10] It also presented better results than previous architectures, such as GoogleLeNet [47], PReLU [18], and VGG [45].

To allow the training of the Inception-v3, we cropped the spectrogram images. To standardize the spectrograms width, we took the area of the spectrogram that corresponds to the 30 seconds placed in the middle of the audio clip. In the case of audio clips with less

[9] http://sox.sourceforge.net/

[10] http://image-net.org/challenges/LSVRC/2012/

than 30 seconds duration, we used zero padding to fill the images and keep their size in the standard. After the training of the Inception-v3, we used the 2,048 weight values of the penultimate layer of the net as features. Before we extract the features, we applied transfer learning using the weights of an Inception-V3 trained on the IMAGENET Dataset [29].

4.3 Feature extraction from poster (Phase 2)

As with trailer frames and audio, we explored the content of movie posters images using both handcrafted and non-handcrafted descriptors.

In the handcrafted scenario, we employed two strategies widely used to describe textural and chromatic content of images. For texture description, we used LBP, which we applied for the poster image as a whole. We firstly converted the poster images to grayscale before the LBP extraction, which we accomplished using eight neighbors with radius two (i.e. $LBP_{8,2}$), resulting in a 59-dimensional feature vector [1]. We normalized the LBP vectors before performing the classification.

Similar to the protocol we applied to the frames of the trailer, we used color histograms to capture the chromatic content of the posters. In this sense, we summed up the occurrences of each intensity level on each color channel of the RGB (i.e. Red, Green, and Blue) color space to build histograms. Thus, we created a 768-dimensional histogram (i.e. $3 \times 256 = 768$) and we used it as the feature vector to describe the chromatic content of the posters. As with LBP, we also normalized these vectors before performing the classification.

In the non-handcrafted scenario, we used the Inception-v3 for extracting features from the poster images. The choice of this particular architecture was motivated by the reasons previously presented in Section 4.2. In this case, we fed the network with the image of the poster as a whole. Due to the characteristics of the Inception-v3, it outputs a vector of weights with 2,048 dimensions, which we used as the descriptor of the poster. As done for the audio, we applied transfer learning in the weights initialization using the IMAGENET dataset [29].

4.4 Feature extraction from subtitle and synopsis (Phase 2)

To represent textual content, which is given by subtitles and synopses in our dataset, we used two well-known text processing techniques: TF-IDF features extracted from N-grams [10], and a representation learning approach based on Long Short-Term Memory (LSTM) [20].

LSTMs are a special kind of Recurrent Neural Networks (RNN), explicitly designed to avoid the long-term dependency problem. We chose to use the LSTM to process the textual content because it has been used as one of the best RNN variations to solve the word-level language modeling context [16]. Table 3 shows the parameter settings used in the experiments with the LSTM algorithm in this study. We chose Word2vec [32] as the embedding representation because it has presented good results in several Natural Language Processing (NLP) tasks. Moreover, we defined the max features of this layer as 50,000. For every sample (i.e. synopsis or subtitle), we passed the first 300 words to the network, as shown in the parameters "Num. of words" and "Input length". We also defined the number of hidden nodes in the LSTM as 128, which defines the dimensionality of the output.

To briefly describe the rationale that lies behind TF-IDF, we start from the concept of bag-of-word (BOW), which is one of the most used strategies to represent texts in NLP. The BOW representation counts the occurrences of words without considering their position in the text, thus losing contextual information [26]. Some contextual information can be obtained by counting word sequences, in a representation known as N-gram, the most

Table 3 Parameter settings of the LSTM algorithm

Parameter	Value
Embedding	Word2Vec
Preprocessing	Tokenizer
Num. of Words (Tokenizer)	300
Input Length	300
Dropout Rate	0.2
Optimizer	Adam
Loss	Cross Entropy
Max Features (Embedding)	50,000
Activation	Softmax
Dimensionality of the Output	128
Epochs	150

well-known forms being: 1-gram, 2-gram, 3-gram, and 4-gram, which use sequences of one, two, three and four words, respectively. Although the use of N-grams is capable of aggregating more information about context when compared to BOW, it is usually not enough to properly represent the context in some circumstances.

N-grams with high frequency in many documents may not be relevant to the learning process; however, n-grams with high frequency tend to be more relevant than less frequent ones [26]. To emphasize important N-grams, considering both the frequencies in specific documents and all documents in the dataset, it is common to divide each N-gram frequency (Term Frequency - TF) by its inverse frequency in other documents (Inverse Document Frequency - IDF). This procedure is usually called TF-IDF on BOW/N-grams. With the use of TF-IDF, the terms that appear in few documents are highlighted, but without disregarding their frequencies in a specific document.

Before generating the N-grams, we pre-processed the texts to remove special characters and to perform stemming [40], which reduces inflected words to their stem. These techniques were used mainly to reduce the N-grams dimensionality. Since we ranged N from one to four to create the N-grams used to calculate TF-IDF descriptors, we have a total of four feature sets for each data source (subtitles and synopses).

4.5 Summary of features and dimensionality reduction (Phases 2 and 3)

We produced a total of 22 different types of feature sets from the different sources of information explored in this work. The name assigned to the feature follows a pattern in which the data source and the feature are separated by the character "-" (e.g. "DATA SOURCE-FEATURE"). The features are organized in five different groups, according to the source of information used to produce them, as follows:

- Features from the frames of the video trailers:
  - TRAILER-C3D: created using a three-dimensional CNN, as proposed in [49];
  - TRAILER-LRCN: created using both CNN and LSTM for feature extraction;
  - TRAILER-CTT-MMC: created using the CTT-MMC architecture, proposed by Wehrmann and Barros [52];
  - TRAILER-LBP: we used the LBP operator, proposed by Ojala et al. [35] to extract texture descriptors of the frames. To combine the results obtained from different frames, whe applied the K-Means algorithm;

  - TRAILER-RGB: created using the histogram of the channels from the RGB color space. We used the K-Means algorithm to integrate the information taken from different frames.

- Features from the audio content of the video trailers:
  - AUDIO-MFCC: composed of coefficients obtained using MFCC, an audio content descriptor widely used to capture the timbral content of the sound [17];
  - AUDIO-SSD: composed of SSD descriptors, which captures rhythmic and timbral information of the sound [30];
  - AUDIO-SPEC-LBP: texture descriptors extracted from the time-frequency visual representation of the sound, as proposed in [9];
  - AUDIO-SPEC-INCv3: features learned by the Inception-v3 with transfer learning applied to the spectrogram image obtained from the sound.

- Features from the movie posters:
  - POSTER-LBP: texture features extracted from poster images using the LBP operator;
  - POSTER-RGB: texture features extracted from poster images using the RGB histogram;
  - POSTER-INCv3: features learned using the Inception-v3 with transfer learning applied to the image of the movie posters.

- Features from the movie subtitles:
  - SUB-TFIDF-1: TF-IDF features extracted using a 1-gram model applied on the subtitles;
  - SUB-TFIDF-2: TF-IDF features extracted using a 2-gram model applied on the subtitles;
  - SUB-TFIDF-3: TF-IDF features extracted using a 3-gram model applied on the subtitles;
  - SUB-TFIDF-4: TF-IDF features extracted using a 4-gram model applied on the subtitles;
  - SUB-LSTM: created using the LSTM on the subtitles;

- Features from the movie synopses:
  - SYN-TFIDF-1: TF-IDF features extracted using a 1-gram model applied on the synopses;
  - SYN-TFIDF-2: TF-IDF features extracted using a 2-gram model applied on the synopses;
  - SYN-TFIDF-3: TF-IDF features extracted using a 3-gram model applied on the synopses;
  - SYN-TFIDF-4: TF-IDF features extracted using a 4-gram model applied on the synopses;
  - SYN-LSTM: created using the LSTM on the synopses.

An important aspect to consider when developing classifying systems concerns the dimensionality of the representations. The representations described in this section presented particular issues that must be addressed before the classification phase. The sizes of the different feature vectors are far from uniform, which means that some descriptors are much bigger than others. Thus, in some cases, the configuration parameters of the classifiers to some feature vectors are not the most suitable for other feature vectors. By carefully analyzing the vectors produced from the different sources of information used here, we noticed that for some representations extracted from the text (such as 4-grams), the feature vectors may have an insurmountable size (about 51 million features). Therefore, they must be reduced to computational feasibility, and also to avoid problems related to the curse of dimensionality.

Proposed by Baraniuk et al. [2], compressive sampling is a method for dimensionality reduction that applies the concept of random projections for machine learning purposes. It was originally proposed to situations in which the signal has a sparse representation, so that it is possible to reconstruct the signal from a few random measurements. The mathematical details behind the method can be found in Johnson and Lindenstrauss [25] and Baraniuk et al. [3]. After evaluating it empirically in our dataset, we observed that, for some feature vectors, the method was highly effective in reducing dimensionality while maintaining the hit rate. Therefore, we applied it to the text representations based on N-grams. Table 4 shows the dimensionality of feature vectors obtained from the text, before and after the use of compressive sampling.

4.6 Resampling (Phase 4)

Many researchers face problems related to unbalanced class distribution, which can make the task of learning an even greater challenge [28]. Classifiers usually focus on minimizing the global error rate and, therefore, when dealing with unbalanced data, tend to benefit the most frequent classes, decreasing the prediction for infrequent classes and, consequently, affecting the overall performance by class.

Resampling techniques, which are the most common and widely used solution for the imbalance issue, can be subcategorized in oversampling and undersampling. While the first balances the dataset by creating new samples for the minority classes, the second aims at removing samples from the majority classes.

To deal with the imbalance of our dataset, we included a resampling phase (represented by the Blue rectangle in Fig. 3) in our classification schema. We used two different resampling algorithms in this phase: the Multi-Label Synthetic Minority Over-sampling Technique (ML-SMOTE) [8] and the Multi-Label Tomek Link (MLTL) [38].

ML-SMOTE was proposed by Charte et al. [8] and is one of the most well-known resampling algorithms in the literature. Its main idea is to create synthetic samples combining the

Table 4 Dimensionality of the features vectors obtained from text before and after applying compressive sampling

Media source	Feature type	Before	After
Subtitle	SUB-TFIDF-1	209,917	128
	SUB-TFIDF-2	6,911,803	128
	SUB-TFIDF-3	28,831,178	128
	SUB-TFIDF-4	51,854,899	128
Synopsis	SYN-TFIDF-1	22,838	128
	SYN-TFIDF-2	231,618	128
	SYN-TFIDF-3	414,287	128
	SYN-TFIDF-4	471,837	128

features of samples from the minority classes with interpolation techniques. In our experiments, we set the resize rate of ML-SMOTE to 25% and a ranking label combination was chosen.

Proposed by Pereira et al. [38], MLTL is one of the most recently published resampling techniques in the literature. This undersampling algorithm detects and removes the so-called Tomek Links from the multi-label dataset. A pair of instances is a Tomek Link if they are the nearest neighbors, but belong to different classes. Besides performing undersampling, MLTL can also be applied in a post-process cleaning step for the ML-SMOTE algorithm. According to Pereira et al. [38], the reason to use it as a post-process cleaning step leans on the fact that, after applying ML-SMOTE, the class groups are usually not well defined, i.e., some instances from the majority class may be invading the minority class space or vice versa. Thus, MLTL may clean the feature space and smooth the edges between the classes.

4.7 Classification algorithms and multimodal integration (Phases 5 and 6)

According to Tsoumakas and Katakis [50], we can group the existing methods for multi-label classification into two main categories: (1) problem transformation methods, and (2) algorithm adaptation methods. Problem transformation methods are those in which the multi-label classification problem is transformed into one or more single-label classification or regression problems, for which there is a huge bibliography of learning algorithms. Algorithm adaptation methods are those that extend specific learning algorithms to directly deal with multi-label data.

In this work, we chose to employ LSTM and Binary Relevance (BR), which are problem transformation methods. In the case of BR, we used Multilayer Perceptron (BR-MLP), Support Vector Machine (BR-SVM), and Decision Tree (BR-DT) as base classifiers. The BR method builds independent binary classifiers for each label (l_i). Each classifier maps the original dataset to a single binary label with values l_i, $\tilde{}l_i$. The classification of a new instance is given by the set of labels l_i produced by the classifiers [6]. In addition to the BR classifiers, we used the LSTM architecture to classify synopses and subtitles.

We also experimented with the algorithm adaptation method ML-kNN. The Multi-Label k-Nearest Neighbors (ML-kNN) [55] is an adaptation of the kNN lazy learning algorithm for multi-label data. In essence, ML-kNN uses the kNN algorithm independently for each label l: It finds the k nearest examples to the test instance and considers those that are labeled at least with l as positive and the rest as negative.

We accomplished the multimodal integration itself by late fusion. Late fusion strategies usually achieve good results in scenarios in which there is complementarity among the outputs of the classifiers involved in the fusion. In these cases, the classifiers do not make the same misclassification, so, when combined, they help each other improving the classification performance [27].

Late fusion strategies are so-called because they combine the output of the classifiers, in opposition to early fusion strategies, which combine the feature vectors. Therefore, the combination is, in most cases, achieved through calculations in the predicted scores for each class involved in the problem.

Among the most used fusion strategies, we highlight the rules described as follows, originally introduced by Kittler et al. [27], and adapted in this work for the multi-label scenario.

- Sum rule (Sum): Corresponds to the sum of the scores provided by each classifier for each class.
- Product rule (Prod): Corresponds to the product between the scores provided by each classifier for each class.
- Max rule (Max): Selects the highest probability score from each classifier.

In addition to the three aforementioned fusion rules, we introduced a threshold, so that classes with a score greater than or equal to 0.3 after the merger (Sum and Max rules) were considered qualified classes and were assigned to the pattern under classification. Due to the reduction in the probabilities values when the product rule is applied, its threshold was set as 0.01. Both thresholds were empirically adjusted.

An important aspect regarding the multimodal integration concerns the criteria adopted to select the classifiers to be used in the fusion. We tried two approaches: TOP-N, and BEST-ON-DATA. The former consists of selecting the N classifiers with the best overall performance based on the F-Score metric. The latter consists of using the best classifier for each kind of data source, also considering the F-Score. In this case, we have always five classifiers to be combined. In total, we generated 40 different classifiers using fusion (36 using TOP-N, and four using BEST-ON-DATA).

5 Experimental protocol and results

Considering the number of sources of information, representations, and classification algorithms used in this work, it would not be appropriate to describe in this paper all the experimental results. This scenario could be even more complex if we consider the wide range of evaluation metrics for classification problems available in the literature. Therefore, we decided to organize and describe in this section the main results, emphasizing those situations where the best results were achieved. We also included an explanation regarding the evaluation metrics chosen. In addition, we made available a complete description containing all our results for those who are interested in it.[11] We also included the experiments with the resampling strategy presented in Section 4.6 in the results file, but they are not discussed in Section 5.2 because this strategy did not improve the results.

5.1 Evaluation metrics

There are several metrics proposed in the literature to evaluate multi-label classifiers. One of the main differences among them refers to the rigorousness in considering the classification result as a hit or a misclassification. In one extreme, we can point out “Subset Accuracy” as the most severe metric. Based on this metric, we have a hit only when the classifier correctly identifies the exact subset of labels assigned to the sample under classification. Other metrics, like Accuracy, Precision, and Recall [19], have a more relaxed criterion and thereby tends to provide higher hit rates. Provost et al. [43] claim that the use of a single metric, accuracy for example, may lead to mistakes in the interpretation of results. Therefore, they recommend the use of AUC-ROC as a good metric for general purposes. On the other hand, Davis and Goadrich [11] claim that the AUC-ROC can lead to an optimistic view of the results when the dataset is unbalanced. Alternatively, the authors suggest the use of the

[11] https://drive.google.com/file/d/1uWbLhjAvHovFdqRIgeTJUkN5T2KovnoE/view?usp=sharing

AUC-PR metric, which is supposed to present similar results to AUC-ROC, but with a more realistic view reflected in the rates, better showing the space for improvement.

With this discussion in mind, we decided to use two metrics to evaluate the classifiers in our experiments: i) F-Score, because it is a harmonic mean between the values of recall and precision; and ii) AUC-PR, considering the argument of Davis and Goadrich aforementioned.

All experiments were conducted using 5-fold cross-validation. Therefore, for each combination of features and classifier, we performed the training five times using four folds, and we tested the obtained model on the remaining fold. The results presented correspond to the average of these five tests.

5.2 Results

Table 5 presents the best results, based on F-Score and AUC-PR, for each kind of representation of each of the different sources of information. In cases where the classifier that achieved the best F-Score rate does not match the one that provided the best AUC-PR rate, we presented both results. As we can see, the best results were obtained using the representation SYN-LSTM both considering F-Score or AUC-PR metrics (shown in bold).

Table 6 shows the best results considering the fusion of the TOP-N classifiers, with N ranging from 2 to 10. The TOP-N classifiers were selected from the 10 best classifiers (in terms of F-Score rate) among all possible classifiers. These 10 best classifiers are described in Table 7. Similarly to Table 5, in Table 6 we have duplicated the rows of TOP-N where the best results with AUC-PR and F-Score were obtained using different fusion rules. Again, the best results are in bold.

The best results we obtained using the unimodal classifiers created from each data source, in isolation, are described in Table 8. Conversely, we describe in Table 9 the results obtained by fusing the classifiers with best F-Scores from each different source of information. We called these latter “best-on-data”.

Figure 4 shows the recall per label considering the best classifiers with and without fusion, based on both F-Score and AUC-PR metrics. As can be observed, the values from the best fusion improves the results for all labels when comparing with the best single classifier.

5.3 An ablation study

Aiming to achieve a better understanding regarding the impact of each data source on the results, we conducted an ablation study. In this study, we evaluated in each round of experiments the impact of excluding each different data source. It is worthy mentioning that we used the classifier with best performance for each data source, except that one created from the data source ignored in that round. By this way, it was possible to identify data sources that eventually do not necessarily contribute to improve the results. The results obtained are shown in Table 10.

The best result was obtained by excluding the classifiers created using trailers as the data source, with 0.674 of F-score and 0.725 of AUC-PR. Surprisingly, it was the best of all results obtained in this work. Both results were obtained using the TOP-2 fusion strategy with the Prod_Proba rule, combining the classifiers SYN-LSTM and SUB-LSTM.

Table 5 Best results for all the representations evaluated considering the F-Score and the AUC-PR

Data source-Feature	Classifier	F-Score	AUC-PR
Trailer frames			
TRAILER-C3D	BR_MLP	0.471	0.473
TRAILER-LRCN	BR_MLP	0.292	0.395
TRAILER-LRCN	MLkNN	0.297	0.306
TRAILER-CTT-MMC	BR_MLP	0.317	0.362
TRAILER-LBP	BR_MLP	0.327	0.322
TRAILER-RGB	BR_MLP	0.318	0.285
TRAILER-RGB	MLkNN	0.283	0.299
Trailer audio			
AUDIO-MFCC	BR_MLP	0.264	0.401
AUDIO-MFCC	MLkNN	0.312	0.315
AUDIO-SSD	BR_MLP	0.326	0.434
AUDIO-SPEC-LBP	BR_MLP	0.254	0.404
AUDIO-SPEC-LBP	BR_DT	0.312	0.189
AUDIO-SPEC-INCv3	BR_MLP	0.334	0.311
Poster			
POSTER-LBP	BR_MLP	0.223	0.385
POSTER-LBP	BR_DT	0.275	0.172
POSTER-RGB	BR_MLP	0.267	0.304
POSTER-RGB	BR_DT	0.276	0.172
POSTER-INCv3	BR_MLP	0.409	0.391
Subtitle			
SUB-TFIDF-1	BR_MLP	0.366	0.329
SUB-LSTM	Deep Learning	0.436	0.613
Synopsis			
SYN-TFIDF-1	BR_MLP	0.288	0.266
SYN-TFIDF-1	MLkNN	0.225	0.285
SYN-LSTM	Deep Learning	**0.488**	**0.678**

6 Discussion

To analyze the results from different perspectives, we decided to guide our discussion in the search for answers to the following questions: (1) Which data source/representation provided the best and worst individual results? (2) Has the fusion contributed to improve the results? (3) Which classifier performed better as a whole? (4) Which classifier algorithm, and with which representations, contributed to the best fusions in terms of performance? (5) Which movie genres are easier/harder to identify? (6) What are the impacts of each data source in the classification results?

One of the first questions that come to mind about our results is related to the data source/representation which provides the best and worst individual results (question 1). Looking at Table 5, we observe that, in terms of both F-Score and AUC-PR, a deep learning

Table 6 Best TOP-N fusion results with N ranging from 2 to 10 considering the F-Score and the AUC-PR metrics. “Proba” refers to the probabilities, and “Pred” refers to the predictions produced by the classifiers

Strategy	Fusion Method	F-Score	AUC-PR
TOP-2	Proba_Prod	**0.628**	0.664
TOP-3	Proba_Prod	0.326	0.636
TOP-3	Pred_Sum	0.577	0.483
TOP-4	Proba_Sum	0.527	0.566
TOP-4	Proba_Prod	0.229	0.636
TOP-5	Proba_Prod	0.099	**0.673**
TOP-5	Pred_Sum	0.563	0.539
TOP-6	Proba_Sum	0.544	0.619
TOP-6	Pred_Sum	0.590	0.555
TOP-7	Proba_Sum	0.549	0.602
TOP-8	Proba_Sum	0.555	0.607
TOP-8	Pred_Sum	0.569	0.564
TOP-9	Proba_Sum	0.532	0.606
TOP-9	Pred_Sum	0.578	0.567
TOP-10	Proba_Sum	0.531	0.603
TOP-10	Proba_Sum	0.540	0.568

Table 7 Ten best unimodal classifiers considering the F-Score metric

#	Data source-Feature	Classifier	F-Score
1	SYN-LSTM	Deep Learning	0.488
2	TRAILER-C3D	BR_MLP	0.471
3	TRAILER-C3D	MLkNN	0.457
4	TRAILER-C3D	BR_SVM	0.457
5	SUB-LSTM	Deep Learning	0.436
6	POSTER-INCv3	BR_MLP	0.409
7	TRAILER-C3D	BR_DT	0.402
8	SUB-TFIDF-1	BR_MLP	0.366
9	POSTER-INCv3	MLkNN	0.355
10	AUDIO-SPEC-INCv3	BR_MLP	0.334

Table 8 Best classifiers for each kind of data source considering the F-Score metric

Data source	Feature	Classifier	F-Score
Trailer frames	TRAILER-C3D	BR_MLP	0.471
Trailer audio	AUDIO-SPEC-INCv3	BR_MLP	0.334
Poster	POSTER-INCv3	BR_MLP	0.409
Subtitle	SUB-LSTM	Deep Learning	0.436
Synopsis	SYN-LSTM	Deep Learning	0.488

Table 9 BEST-ON-DATA results

Fusion Method	F-Score	AUC-PR
Proba_Sum	0.558	**0.622**
Proba_Max	0.495	0.465
Proba_Prod	0.263	0.547
Pred_Sum	**0.574**	0.534

method (LSTM) achieved the best rate. The synopses obtained the best result, and, considering the AUC-PR metric, the subtitles obtained the second-best one. This evinces the strength of the features created by deep learning on textual data in the application domain investigated here. It is important to mention that, considering the F-Score metric, this representation also obtained the second-best rate. Considering all the results obtained from each source of information described in Table 5, the trailer audio achieved the worst performance in terms of F-Score, with a top result of 0.334. Considering the AUC-PR, the worst result regarding the different types of data sources remains on the use of poster, with a top result of 0.391 (POSTER-INCv3).

Regarding the fusion strategies (question 2), we can highlight the fusion performed using the textual data sources (synopsis and subtitles). Considering all results presented in this paper, the fusion of the best individual classifiers using synopsis (SYN-LSTM) and subtitles (SUB-LSTM) achieved the best F-Score (0.674) and AUC-PR (0.725) rates, as shown in Section 5.3. This scenario confirms one of the main hypotheses investigated in this work, concerning the complementarity among different data sources.

We have also performed a statistical test in order to properly compare the best results obtained with and without fusion for each metric, aiming to check if there is a significant statistical difference between those results, and to indicate that the fusion strategies improve

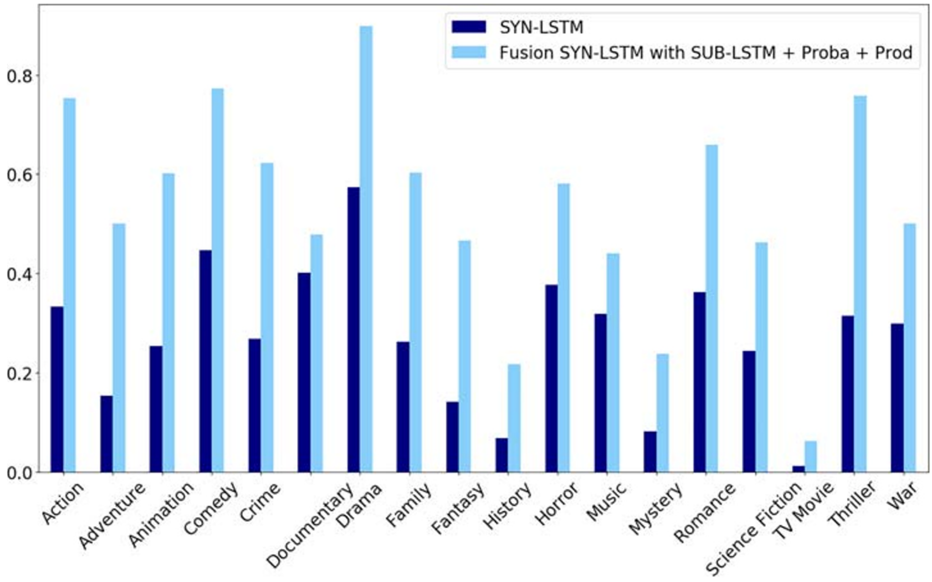

Fig. 4 Hit per genre considering the best classifiers with and without fusion

Table 10 Best results after removing each data source

Removed data source	F-Score	AUC-PR
Audio	0.628	0.673
Poster	0.628	0.673
Subtitles	0.628	0.664
Synopsis	0.553	0.611
Trailer	**0.674**	**0.725**

the results (question 2). For the results without fusion, we used the SYN-LSTM classifier, both for F-Score and AUC-PR. In turn, we have used the classifiers SYN-LSTM and SUB-LSTM with the "Prod_Proba" fusion rule to evaluate the fusion results considering both metrics. As Hypothesis 1, we used the Wilcoxon [53] two-side test to verify whether or not the hypothesis of the results with and without fusion are equal. We also presented a Hypothesis 2 using the Wilcoxon less hypothesis test to check if the mean of the difference between the results with and without fusion is negative (to confirm the Hypothesis 2 the results without fusion cannot be higher than the results with fusion). As described in Table 11, we can discard the Hypothesis 1 with a statistical confidence level of 5%, and consider true the Hypothesis 2 with a statistical confidence level of 5% considering both metrics. These tests indicate that the fusion of classifiers obtained using different kinds of data source improves the rates achieved when analyzing the F-Score and AUC-PR metric (Question 2).

Regarding the classifier algorithms evaluated here, by looking at Table 5, we can observe that the BR_MLP classifier is the one with the biggest number of occurrences, fourteen in total. This algorithm also appears at least once for each different kind of data source. Those affirmations indicate that the BR_MLP classifier performed better in comparison with the other classifiers when looking in a wide scenario (question 3). Despite this impressive number of occurrences, we need to point out that the LSTM deep learning strategy has figured as the best one in terms of hit rate, both considering F-Score and AUC-PR, especially when applied on the synopses. It also appeared as one of the most important classifiers in the fusion stage (question 4).

Figure 4 lets us evaluate the results individually obtained for the 18 labels of the database (question 5). By analyzing those results, we can observe that four classes present hit rates above 70%: Drama, Comedy, Action, and Thriller. In all cases, the results were achieved using late fusion. Considering this same strategy, the worst results appear for the genre TV Movie. This evidences the impact of the dataset imbalance, since the genres with most samples (i.e. Drama, Comedy, Animation, Action, and Thriller) were easier to classify, and the genre with fewer samples (i.e. TV movie) was the hardest one to predict. It is worth remembering that we tried a resampling strategy to mitigate imbalance impacts, but even in these cases we did not get better results.

Finally, the last question is raised concerning the impacts of each data source in the classification results. The answer to this question is mainly grounded in the ablation study

Table 11 Statistical tests comparing the best results obtained with and without fusion

Metric	Wilcoxon	Wilcoxon "Less" Hypothesis
F-Score	0.043	0.022
AUC-PR	0.043	0.022

provided in Section 5.3. Analyzing Table 10, the first important point to observe is two opposite impacts on the classification results when removing the Synopsis and Trailer data sources. Whilst the ablation study has show that the removal of the Synopsis from the fusion schema significantly decreased the classification results, showing its complementarity with the other data sources, the removal of the Trailer source from the fusion has actually improved the results, which shows that the classification probabilities extracted from the Trailer sources might be mixing up the combinations. Furthermore, in the ablation study, the removal of the Audio, Poster or Subtitles sources, did not show neither positive or negative impacts in the classification results.

In summary, the data source “synopsis” with the LSTM classifier achieved the best individually performance (Question 1). On the other hand, the poster and the trailer audio presented the worst results. To compare the improvements achieved when using late fusion strategies, we applied the Wilcoxon statistical test. For the F-Score and AUC-PR metric, we checked with a confidence interval of 5% that the result obtained with the fusion strategy was higher than the conventional approach. The test indicates that the use of late fusion improved the classification performance (Question 2). The classifier that most occurred on the best results for the features is the BR_MLP, showing to be a consistent and reliable classifier (Question 3). The LSTM classifier figured as the one that most contributed to the fusion, presenting the two best results for the AUC-PR metric (Question 4). Regarding the performance for each genre individually, genres with more samples (Drama and Action) were the easiest to classify, while the genre with the fewest samples (TV movie) was the most difficult (Question 5). Finally, concerning the impacts of the different data sources in the classification results, an ablation study has shown a positive impact of the Synopsis source and a negative impact of the Trailer source (Question 6).

7 Concluding remarks and future works

In this work, we addressed the multimodal movie genre classification as a multi-label classification problem. As far as we know, this is the most comprehensive study done in this scenario. To perform the classification, we investigated the use of five different sources of information, namely trailer audio, trailer frames, synopses, subtitles, and posters. We designed and curated a dataset to support the development of the experimental protocol and made it available to the research community.

We created classifiers using different representations based on the different sources of data, and they were evaluated both individually, and combined with each other by late fusion. The results were reported using F-Score and AUC-PR metrics, and in both cases, the best individual result was obtained using an LSTM representation created from the synopses. Regarding the classifiers combined with fusion, the combination of SYN-LSTM (LSTM created using synopses) with SUB-LSTM (LSTM created using subtitles) achieved the best F-Score (0.674) and AUC-PR (0.725) rates. These results corroborate the existence of complementarity among classifiers created using different sources of information in the task addressed here. These results also confirm the success of deep learning strategies to perform multimedia classification, already investigated by the authors on music classification.

In the future, we intend to investigate the use of optimization methods (e.g. Particle Swarm Optimization (PSO), Genetic Algorithm (GA), among others) to conduct the search for the best strategy of fusion of classifiers. We also plan to evaluate the use of dynamic classifier selection techniques and the use of segment selection methods aiming at automatically identifying the most promising parts of the movie trailer to perform the genre classification. Another future work is to expand the dataset to allow works in other applications, such as recommender systems.

Acknowledgments We thank the Brazilian Research Support Agencies: CNPq - National Council for Scientific and Technological Development (grants #156956/2018-7 and #164837/2018-3), CAPES - Coordination for the Improvement of Higher Education Personnel and FA - Araucária Foundation for their financial support for their financial support. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan XP GPU used in this research.

References

1. Ahonen T, Hadid A, Pietikäinen M (2004) Face recognition with local binary patterns. In: Proceedings of the European conference on computer vision. Springer, pp 469–481
2. Baraniuk RG, Cevher V, Wakin MB (2010) Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. Proc IEEE 98(6):959–971
3. Baraniuk R, Davenport M, DeVore R, Wakin M (2008) A simple proof of the restricted isometry property for random matrices. Constr Approx 28(3):253–263
4. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
5. Brezeale D, Cook DJ (2006) Using closed captions and visual features to classify movies by genre. In: Proceedings of the International workshop on multimedia data mining
6. Brinker K, Fürnkranz J, Hüllermeier E (2006) A unified model for multilabel classification and ranking. In: Proceedings of the European conference on artificial intelligence. IOS Press, pp 489–493
7. Cascante-Bonilla P, Sitaraman K, Luo M, Ordonez V (2019) Moviescope: Large-scale analysis of movies using multiple modalities. arXiv:1908.03180
8. Charte F, Rivera A, del Jesus M, Herrera F (2015) MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation. Knowl.-Based Syst 89:385–397
9. Costa YandreMG, Oliveira LS, Koerich AL, Gouyon F, Martins JG (2012) Music genre classification using lbp textural features. Signal Process 92(11):2723–2737
10. Damashek M (1995) Gauging similarity with n-grams: Language-independent categorization of text. Science 267(5199):843–848
11. Davis J, Goadrich M (2006) The relationship between precision-recall and roc curves. In: Proceedings of the international conference on machine learning. ACM, pp 233–240
12. Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2625–2634
13. Frakes WB, Baeza-Yates R (1992) Information Retrieval: Data structures & algorithms, vol 331. Prentice Hall, Englewood Cliffs
14. Fu Z, Li B, Li J, Wei S (2015) Fast film genres classification combining poster and synopsis. In: International conference on intelligent science and big data engineering. Springer, pp 72–81
15. Gonzalez RC, Woods RE, et al. (2002) Digital image processing. Prentice Hall, Upper Saddle River
16. Greff Klaus, Srivastava RupeshK, Koutník J, Steunebrink BR, Schmidhuber J (2017) Lstm: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10):2222–2232
17. Hasan MR, Jamil M, Rabbani MG, Rahman MS (2004) Speaker identification using mel frequency cepstral coefficients. Variations, 1(4)
18. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of The IEEE international conference on computer vision
19. Herrera F, Charte F, Rivera AJ, Del Jesus MJ (2016) Multilabel classification. Springer, Berlin
20. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
21. Hochreiter S, Schmidhuber J (1997) Lstm can solve hard long time lag problems. In: Advances in neural information processing systems, pp 473–479
22. Hofmann T (2001) Unsupervised learning by probabilistic latent semantic analysis. Mach Learn 42(1-2):177–196
23. Hong H-Z, Hwang J-IG (2015) Multimodal PLSA for movie genre classification. In: Proceedings of the International workshop on multiple classifier systems. Springer, pp 159–167

24. Huang Y-F, Wang S-H (2012) Movie genre classification using svm with audio and video features. In: Proceedings of the International conference on active media technology. Springer, pp 1–10
25. Johnson WB, Lindenstrauss J (1984) Extensions of lipschitz mappings into a hilbert space. Contemporary Mathematics (26)189–206
26. Jurafsky D, Martin JH (2008) Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition
27. Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20(3):226–239
28. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Progress Artif Intell 5(4):221–232
29. Krizhevsky A, Sutskever I, Hinton GE, Bottou L (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Weinberger KQ (eds) Advances in Neural Information Processing Systems, vol 25, pp 1097–1105
30. Lidy T, Silla Jr CN, Cornelis O, Gouyon F, Rauber A, Kaestner CAA, Koerich AL (2010) On the suitability of state-of-the-art music information retrieval methods for analyzing, categorizing and accessing non-western and ethnic music collections. Signal Process 90(4):1032–1048
31. Logan B, et al. (2000) Mel frequency cepstral coefficients for music modeling. In: Proceedings of the international society for music information retrieval conference, vol 270, pp 1–11
32. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:1301.3781
33. Nanni L, Costa YMG, Lucio DR, Silla CN, Brahnam S (2016) Combining visual and acoustic features for bird species classification. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, pp 396–401
34. Nanni L, Costa YMG, Brahnam S (2014) Set of texture descriptors for music genre classification
35. Ojala T, Pietikäinen M, Mäenpää T (2001) A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In: Proceedings of international conference on advances in pattern recognition. Springer, pp 399–408
36. Oliva Aude, Torralba A (2001) Modeling the shape of the scene: A holistic representation of the spatial envelope. Int J Comput Vis 42(3):145–175
37. Oramas S, Nieto O, Barbieri F, Serra X (2017) Multi-label music genre classification from audio, text, and images using deep features. arXiv:1707.04916
38. Pereira RM, Costa YMG, Silla Jr. CN (2020) MLTL: A multi-label approach for the tomek link undersampling algorithm. Neurocomputing 383:95–105
39. Pons J, Serra X (2019) Randomly weighted cnns for (music) audio classification. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing. IEEE, pp 336–340
40. Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137
41. Portolese G, Domingues MA, Feltrim VD (2019) Exploring textual features for multi-label classification of portuguese film synopses. In: EPIA Conference on artificial intelligence. Springer, pp 669–681
42. Portolese G, Feltrim VD (2018) On the use of synopsis-based features for film genre classification. In: Anais do XV Encontro Nacional de Inteligência Artificial e Computacional. SBC, pp 892–902
43. Provost F, Fawcett T, Kohavi R (1998) The case against accuracy estimation while comparing induction algorithms. In: Proceedings of the international conference on machine learning
44. Salamon J, Bello JP (2017) Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters 24(3):279–283
45. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
46. Simões GS, Wehrmann J, Barros RC, Ruiz DD (2016) Movie genre classification with convolutional neural networks. In: Proceedings of the international joint conference on neural networks. IEEE, pp 259–266
47. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In proceedings of the IEEE conference on computer vision and pattern recognition
48. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition
49. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497
50. Tsoumakas G, Katakis I (2007) Multi-label classification: An overview. International Journal of Data Warehousing and Mining 3(3):1–13

51. Wang C-M, Huang Y-F (2010) Self-adaptive harmony search algorithm for optimization. Expert Syst Appl 37(4):2826–2837
52. Wehrmann J, Barros RC (2017) Movie genre classification: A multi-label approach based on convolutions through time. Appl Soft Comput 61:973–982
53. Wilcoxon F (1992) Individual comparisons by ranking methods. In: Breakthroughs in statistics. Springer, pp 196–202
54. Wu J, Rehg JM (2008) Where am i: Place instance and category recognition using spatial pact. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 1–8
55. Zhang M-L, Zhou Z-H (2007) ML-KNN A lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
56. Zhang M-L, Zhou Z-H (2014) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837
57. Zhou H, Hermans T, Karandikar AV, Rehg JM (2010) Movie genre classification via scene categorization. In: Proceedings of the ACM International conference on multimedia. ACM, pp 747–750

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Rafael B. Mangolin[1] · Rodolfo M. Pereira[2,3] · Alceu S. Britto Jr.[3] · Carlos N. Silla Jr.[3] · Valéria D. Feltrim[1] · Diego Bertolini[4] · Yandre M. G. Costa[1]

Rodolfo M. Pereira
rodolfomp123@gmail.com

Alceu S. Britto Jr.
alceu@ppgia.pucpr.br

Carlos N. Silla Jr.
carlos.sillajr@gmail.com

Valéria D. Feltrim
valeria.feltrim@gmail.com

Diego Bertolini
diegobertolini@utfpr.edu.br

Yandre M. G. Costa
yandre@din.uem.br

[1] Department of Informatics, State University of Maringá, Av. Colombo, 5790,
Maringá, Paraná, Brazil

[2] Federal Institute of Paraná, Pinhais, Paraná, Brazil

[3] Pontifical Catholic University of Paraná, Curitiba, Paraná, Brazil

[4] Federal Technological University of Paraná, Campo Mourão, Paraná, Brazil