Towards Extreme Multi-Label Classification of Multimedia Content

(Discussion Paper)

Marco Minici[1,2], Francesco Sergio Pisani[2], Massimo Guarascio[2] and Giuseppe Manco[2]

[1]Università degli Studi di Pisa
[2]ICAR-CNR, Rende (CS), Italy

Abstract

Providing rich and accurate metadata for indexing media content represents a major issue for enterprises offering streaming entertainment services. Metadata information are usually exploited to boost the search capabilities for relevant contents and as such it can be used by recommendation algorithms for yielding recommendation lists matching user interests. In this context, we investigate the problem of associating suitable labels (or tag) to multimedia contents, that can accurately describe the topics associated with such contents. This task is usually performed by domain experts in a fully manual fashion that makes the overall process time-consuming and susceptible to errors. In this work we propose a Deep Learning based framework for semi-automatic, multi-label and semi-supervised classification. By integrating different data types (e.g., text, images, etc.) the approach allows for tagging media contents with specific labels. A preliminary experimentation conducted on a real dataset demonstrates the quality of the approach in terms of predictive accuracy.

Keywords

Extreme Multi-Label Classification, Data Integration, Natural Language Processing, Semi-supervised Learning

1. Introduction

Nowadays, entertainment industry represents one of the most profitable and widespread business sector, with a constant growth in terms of number of users. With estimated revenues amounting to about 2 trillion of dollars worldwide, providing effective research services is a crucial task for the companies operating in multimedia content delivery. In particular, the rise of streaming services and on-demand contents fostered the interest for AI-based solutions capable to facilitate the research and identification of contents matching the user interests. Just as an example, Recommender Systems (RS) are technologies widely adopted by big players (e.g., Netflix, Disney+, Amazon, etc.) to suggest items of their catalogues able to arouse users' interest.

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy
marco.minici@icar.cnr.it (M. Minici); giuseppe.manco@icar.cnr.it (F. S. Pisani); massimo.guarascio@icar.cnr.it (M. Guarascio); giuseppe.manco@icar.cnr.it (G. Manco)
https://mminici.github.io (M. Minici)
0000-0002-9641-8916 (M. Minici); 0000-0003-2922-0835 (F. S. Pisani); 0000-0001-7711-9833 (M. Guarascio); 0000-0001-9672-3833 (G. Manco)

Besides them, the technologies that allow for enriching content metadata with informative labels (or tags) act a key role as they can exploited to improve the RS performances and simultaneously enable a more effective research by means of the traditional research engines. Basically, these labels are used to group contents exhibiting common features and provide aggregated views for the users. However, the labelling task is a time-consuming and prone to the error process since it is manually performed by domain experts. Indeed, the lack of a common shared taxonomy can lead to yield repeated labels describing the same concept. Moreover, the assignment of a label to a content is subjective and depends on the skill and perception of the expert.

In this scenario, Artificial Intelligence (AI) techniques represent a valuable tool to automate such a process by limiting the human factor and, as a consequence, reducing the classification error. Anyway, effectively addressing this problem requires the development of specific approaches able to cope with different hard issues, i.e., unbalancing of the classes, lack of labelled data, capability of the models to process different types of data (e.g., text, images, etc.) and providing multi-class predictions on an high number of labels.

In particular, Deep Learning (DL) paradigm [1] is considered a state-of-the-art solution for effectively address these issues. DL-based models can be exploited to extract accurate multi-label classification models by combining raw low-level data, gathered from a wide variety of sources (e.g., wikidata, IMDB, etc.). These models learn in a hierarchical fashion: several layers of non-linear processing units are stacked in a single network and each subsequent layer of the architecture can extract features with a higher level of abstraction compared to the previous one. Therefore, Deep Learning-based approaches allow to extract data abstractions and representations at different levels, they also represent a good choice for analyzing raw data provided in different formats and by different types of source.

In this work we propose to combine different types of data from different publicly available data sources for classifying media contents and enrich them with informative labels. In Figure 1, we sketched the overall learning process. After an *Information Retrieval* phase in which data are gathered and wrapped in a single view, these raw data are provided as input to *Machine Learning* block. Our solution adopts a hierarchical Deep Learning based approach: on top, an ensemble of pre-trained models (*Embedder*) are fine-tuned and used to map the input (text and/or images) in a low-dimensional space. Here, the main idea is that contents with similar labels generate similar vector representations (*embeddings*). Then, a clustering algorithm (*Clusterer*) is used to group similar contents and yield sub-samples of the original dataset. Finally, each sub-sample is exploited to learn a local model focused on a limited set of labels that allows for yielding more accurate predictions for specific cases. Although our approach is totally general and capable to handle different type of data by adding specific models to the Embedder, the current implementation focuses on analyzing text data. An experimental evaluation conducted on a real dataset containing movie plots demonstrates the quality of our approach in providing accurate predictions in this challenging scenario.

The rest of this paper is organized as follows: in Section 2 we provide an overview of the main approaches proposed in literature to tackle the automatic content tagging problem. In Section 3, we describe the framework used to address the problem and the deep learning architecture used to learn the multi-label classification model; while in Section 4 we discuss the experimental results. Section 5 concludes the work and introduces some new research lines.

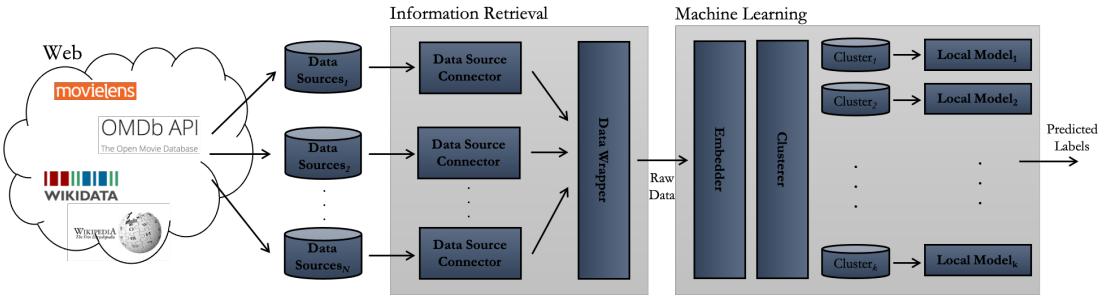


Figure 1: Overview of the Learning Process.

2. Related Work

The problem of classifying movies is not new in the literature and can be considered a general classification task on heterogeneous (video, images, audio, text) data. Wu et al. [2] process user reviews to extract relevant tags for movies. Afterward, they propagate these tags to less popular products according to the movie similarity based on multiple attributes (e.g.: title, summary). Hence, this work draws from the collaborative recommendation paradigm, while our proposal exploits deep metric learning and content-based techniques to solve the tag sparsity problem. Arevalo et al. [3] employ a neural architecture - inspired by recurrent units such as LSTM - named Gated Multimodal Unit (GMU) to effectively combine features coming from the poster image and the plot synopsis. They focus on solving the multi-modal fusion problem rather than the movie tagging itself. Indeed, their dataset contains fewer tags than ours.

The work that most resembles our approach is [4], which makes use of plot synopses to predict tags in the realm of movies. They focus on modeling the plot text as an emotion flow - i.e.: a series of consecutive states of emotion. Their main conclusion is that incorporating the emotion flow increases the tag prediction quality with respect to naive approaches.

Wehrmann and Barros [5] analyze movie trailers for performing multi-label genre classification. They explore the extraction of the audio and image features to establish spatio-temporal relationships between genres and the entire trailer. Similar to our approach, different learners are combined. Standalone models are trained separately for the image and the audio input, then they are fused using a weighted average. Anyhow, as stated by authors, the main limitation of the work relies in the use of only nine common movie genres.

Fish et al. [6] highlight how a single movie genre hold back a large semantic that can be exploited to have a fine-grained description of the movie. The proposed model combines the embeddings produced by four pre-trained multi-modal 'experts' processing the audio and video of the movie. The training process is intended to improve the quality of the embeddings, i.e. the similarity between each movie clip and one of the 20 genres of the tags.

Table 1 summarizes the most significant approaches among those described above. Compared to these approaches, there are some major differences with regards to the problem we aim to tackle: first, the tagging task is relative to a high number of labels. Second, this large number of labels exhibits a long-tail distribution, as illustrated also in Figure 4 and discussed later in the paper. To the best of our knowledge, our solution is the first approach that can handle large

Approach	Dataset	Number of tags	DL architecture	Data Type	XMLC	Multi-Modal	Metric	Result
Kar et al. [4]	MPST	71	LSTM	Text	y	n	Micro F1	0.37
Arevalo et al. [3]	MM-IMDb	26	Multimodal Fusion with Pre-Trained nets	Text, Image	n	y	Micro F1	0.63
Arevalo et al. [3]	MM-IMDb	26	Multimodal Fusion with Pre-Trained nets	Text, Image	n	y	Macro F1	0.54
Wehrmann et al. [5]	LMTD	22	Multimodal Convolutional NN	Audio, Image	n	y	Micro AUC-PR	0.65
Wehrmann et al. [5]	LMTD	22	Multimodal Convolutional NN	Audio, Image	n	y	Macro AUC-PR	0.74
Fish et al. [6]	MMX-Trailer-20	20	Multimodal classifiers	Audio, Image	n	y	F1-weighted	0.60

Table 1
Analysis of current literature on Genre/Tag classification.

amounts of labels (XMLC - eXtreme Multi-Label Classification) and process different types of data.

3. Framework

In this section we illustrate our solution and the main components of the proposed DL-based architecture. As highlighted in Section 1, we adopted a hierarchical approach composed of three main components, as shown in Figure 2: *(i)* an *Embedder*, devoted to summarizing the original input into a vector representation (embedding); *(ii)* a cluster module (*Clusterer*) that allows for identifying media with similar contents and extracting focused sub-samples of the original dataset; and *(iii)* the local models that perform the final predictions.

Embedder. As mentioned above, the current implementation of our technique works on text data (i.e., the movie plots), therefore our Embedder takes the form of a widely adopted (pre-trained) neural network i.e., *BERT* (Bidirectional Encoder Representations from Transformers) [7]. BERT is a transformer-based neural architecture able to process natural language and trained through an algorithm including two main steps, respectively named *Word Masking* and *Next sentence prediction*. In the former step, a percentage of the words composing a sentence is masked and the model is trained to predict the missing terms by considering the word context i.e., the terms that precede and follow the masked one. Then, the model is fine-tuned by considering a further task that allows for understanding the relations among the sentences. Basically, given two subsequent sentences, negative examples are created by replacing the second one with a random sentence. As regards the architecture, BERT can be figured out as a stack of transformer encoder layers that include multiple self attention "heads" [7]. In our framework, we use a BERT instance pre-trained on Wikipedia pages and the final embedding is obtained by averaging the output of the last four layers of the model. Notably, our BERT instance is further fine-tuned by adopting a *Deep Metric Learning* [8] based approach: three instances of the same architecture sharing the same weights are trained against triplets $\langle anchor, positive, negative \rangle$. Basically, the term anchor refers the reference input whereas positive and negative represent other examples respectively similar and dissimilar to the anchor. The goal consists of minimizing the distance between the anchor and the positive example while, simultaneously, the distance between the anchor and the negative one is maximized. A customized version of the *triplet loss* for multi-label tasks is exploited in the learning phase. Specifically, we adopted a semi-hard negative mining approach that filters out negative instances which share more tags with the anchor w.r.t. the positive ones. At prediction time, only a model is used to compute the vector

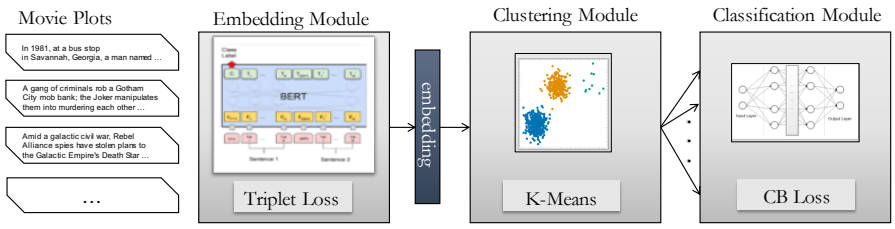


Figure 2: Our proposed 3-step architecture comprises three modules: embedding, clustering, and classification. Each component can be modified to address different goals - e.g.: different data modalities, training strategies, or models.

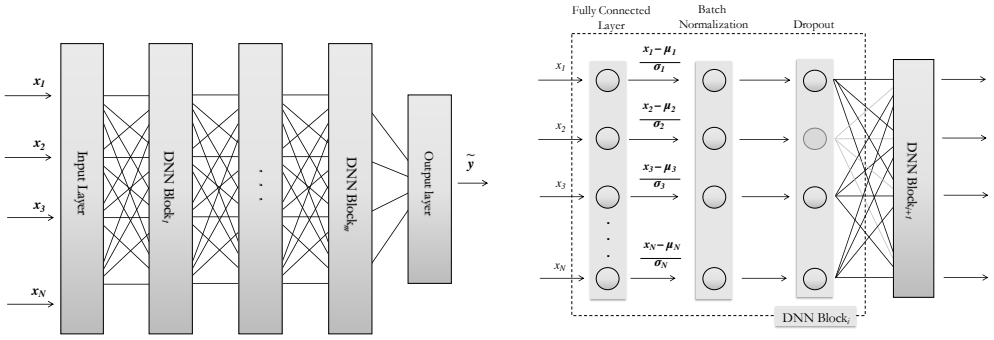
representation of the input data. The main benefit of this approach relies on the possibility of combinatorically increasing the input size and handle the lack of labelled examples.

Clusterer. A clustering algorithm is involved in the framework to group similar movies, thus allowing the deployment of local classification models. Each cluster includes movies that share a minimal number of tags with respect to the whole label space. Hence, this phase further alleviates the extreme-classification problem. New instances, to be classified, are assigned to the closest cluster on the basis of a suitable distance metric (in our case, *euclidean distance* is adopted). As shown in Figure 2, we adopted the K-Means algorithm as our clusterer.

Local Models. In our framework, local models take the form of neural networks too. Specifically, we exploited the DNN-based architecture, shown in Figure 3, to provide more accurate predictions also for minority classes. The base building block of our model includes three types of layers: (i) a fully-connected dense layer equipped with Rectified Linear Unit (ReLU) activation function [9], for each node composing the layer, (ii) a batch-normalization layer for improving stability and performances of the current dense layer [10], (iii) and a dropout layer for reducing the overfitting problem [11]. Several instances of this base component can be stacked in a single model, in particular, in our experimentation we tested a solution with three instances. The output layer of the architectures is equipped with a *sigmoid* activation function[12] and a variable number of neurons depending on the number of labels falling on the cluster associated with the local model. Basically, the output layer provides a class probability for each label. To address the Class Imbalance Problem, each local model is trained by using the Class-Balanced (CB) loss proposed in [13]. Here, the main idea consists of weighting loss inversely with the effective number of samples per class.

4. Experimental Results

In order to assess the quality of our approach in labelling movies with related tags, we conducted a preliminary experimentation on a real dataset extracted by fusing data from different data sources. In particular, first we illustrate the dataset and the challenges to address for providing accurate predictions in this scenario; then we describe the adopted evaluation protocol and



(a) Overview of the model. (b) Layers composing a base DNN component

Figure 3: Local Model architecture: overview (a) and details of the internal structure for each base component (b).

Dataset	Number of Movies	Number of Unique Tags	Avg. number of Tags per Movie	Avg. number of words per Movie
MoTags	5595	85	2.82	71

Table 2
Main statistics of the dataset.

metrics; finally, we show an ablation study aiming at highlight the benefits of the proposed approach.

Our approach has been evaluated on a novel media content dataset gathering different types of information on a movie catalogue (e.g., movie plot, trailer, poster, synopsis, tags, etc.). Specifically, we focused our analysis on text data (i.e., movie plots) and tags extracted by multiple open sources such as Wikipedia, Wikimedia, and TMDB. When available, an extended plot is associated with the movie otherwise it is replaced with its synopsis. As shown in Table 2, our dataset contains ∼5k movies, with about 3 tags per movie. The complete list of tags (85) is showed in Figure 4. It is important noting that a restricted number of tags (mainly the genres) occurs more frequently than others, that can be considered as keywords summarizing some aspects of the movie. In particular, we can see that the data distribution exhibits a long tail shape.

The experiments are perfomed on a DGX machine equipped with V100 GPUs. The dataset is split in training and test set respectively with $70\%/30\%$ percentages. In more detail, the dataset is partitioned in a stratified fashion so to reduce the sampling error. *Adam* is used as optimizer while, as mentioned above, we exploits two loss functions to handle the imbalance problem i.e., the triplet loss and the CB-loss. The first one is used during the Embedder learning phase while the last is used for training the local models. As regards the Clusterer, the number of groups k has been empirically determined to 20. As a result of the clustering phase, each local model can focus the learning on a limited number of tags, in particular the average number of tags per cluster is ~ 12.

In Table 3 we report F1 score averaged according to two different strategies, respectively named *macro* and *micro* [14]. In the former, the F1 is computed for each class and then it is

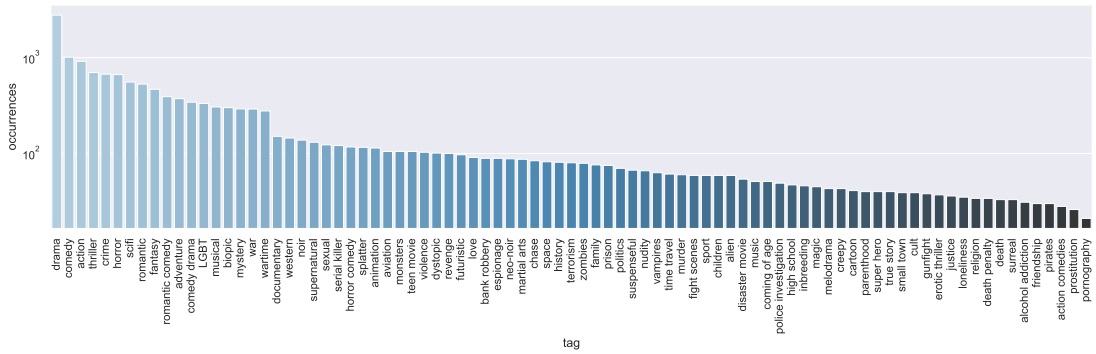


Figure 4: Label distribution. There are 85 tags comprising genres (e.g.: Drama, Comedy) and more specific keywords (e.g.: parenthood, small town). It shows a long-tail shape because of few tags (movie genres) occurring most of the times.

Method	Micro-F1	Macro-F1
BERT	0.176	0.025
BERT*	0.498	0.163
our_approach	0.507	0.253

Table 3
Experimental results on movie tags prediction. BERT* denotes the version trained using Triplet Loss.

averaged, whereas in the latter, the cumulative sum of the counts of various true/false positive/negative is computed, and then the overall measure is calculated. While macro-averaging weights all classes equally, micro-averaging favors bigger classes.

The results shown in table 3 highlight the poor performance of the base model, i.e. the model trained on all tags. It is unable to handle the high number of tags (i.e. classes) and provides inaccurate predictions. The comparison of the values of Micro-F1 and Macro-F1 highlight the influences of the majority classes such as Drama or Comedy on the overall performances. The low value of Macro-F1 show that the model is unable to detect the under-represented tags which are the majority in the dataset.

The adoption of the triplet loss architecture allows for improving the performances of the base model, although the low value of the Macro-F1 indicates poor performances on the minority classes. Finally, the full approach, named in table as *our_approach*, allows for improving also the Macro-F1 value.

5. Conclusions and future work

Enriching metadata with informative labels is a crucial task for the enterprises operating in the media content delivery field. However, automating this process requires to cope with different challenging issues. In this work we proposed a hierarchical DL-Based approach for extreme multi-label classification aiming at providing accurate predictions for movie tagging task. An experimentation conducted on a real dataset demonstrates the quality of the approach.

As a pointer of further research, we aim at boosting the overall performance of the proposed

approach by integrating information coming from unlabeled data in a semi-supervised or self-supervised way. Also, active learning schemes can be fruitfully exploited by implementing ad-hoc oracle labeling strategies. Finally, we are interested to extend the experimentation for a fully multi-modal scenario by including heterogeneous data e.g., movie posters and trailers.

Acknowledgments

This work was supported by PON I&C 2014-2020 FESR MISE, Catch 4.0.

References

[1] Y. Le Cun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[2] C. Wu, C. Wang, Y. Zhou, D. Wu, M. Chen, J. H. Wang, J. Qin, Exploiting user reviews for automatic movie tagging, Multimedia Tools and Applications 79 (2020) 11399–11419.
[3] J. Arevalo, T. Solorio, M. Montes-y Gómez, F. A. González, Gated multimodal units for information fusion, arXiv preprint arXiv:1702.01992 (2017).
[4] S. Kar, S. Maharjan, T. Solorio, Folksonomication: Predicting tags for movies from plot synopses using emotion flow encoded neural network, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 2879–2891.
[5] J. Wehrmann, R. C. Barros, Movie genre classification: A multi-label approach based on convolutions through time, Applied Soft Computing 61 (2017) 973–982.
[6] E. Fish, J. Weinbren, A. Gilbert, Rethinking movie genre classification with fine-grained semantic clustering, arXiv preprint arXiv:2012.02639 (2020).
[7] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
[8] M. KAYA, H. S. BILGE, Deep metric learning: A survey, Symmetry 11 (2019). doi:10.3390/sym11091066.
[9] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th Int. Conf. on Machine Learning, ICML'10, 2010, pp. 807–814.
[10] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proc. of the 32Nd Int. Conf. on Machine Learning - Volume 37, ICML'15, 2015, pp. 448–456.
[11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.
[12] M. Guarascio, G. Manco, E. Ritacco, Deep learning, Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics 1-3 (2018) 634–647.
[13] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9268–9277.
[14] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (2009) 427–437.