On the Use of Synopsis-based Features for Film Genre Classification

Giuseppe Portolese[1] , Valéria Delisandra Feltrim[1]

[1]State University of Maringá, Av. Colombo 5790, CEP 87020-900, Maringá - PR, Brazil

`giuportolese@gmail.com,vdfeltrim@uem.br`

***Abstract.** Technological advancements and the interest of companies that operate in digital environments have made the categorization of mediatic products increasingly popular. This is often a multi-label scenario, where an item may be labeled with many categories. Most of the literature approach film genre classification as a mono-label task, usually relying on audio-visual features. In this paper we explore the use of text-based features extracted from film synopses for multi-label film genre classification. We experimented with 19 feature extraction approaches combined with 4 multi-label classifiers. Our experimental results show f1-scores of up to 54.8%, which are significantly higher than other similar studies presented in the literature.*

1. Introduction

The automatic categorization of mediatic products such as films, books and series has gained popularity due to recent technological advancements and to the influence of electronic commerce companies that use such systems.

In this scenario, it is common for a single product to be classified into several different categories simultaneously, characterizing what in the context of machine learning is called multi-label classification.

However, most of studies published in the literature approaches such classification from a single-label point of view, as it is the case for the film genre classification problem. Other common characteristic of the studies in this domain is the classification based on audio-visual features extracted from trailers, even when other sources of information about the movies, such as their synopses, are available.

Given this context, this paper aims at exploring the usage of textual features extracted from film synopses applied to the classification of film genres. We implemented and evaluated common text feature extractions techniques and diferent multi-label classifiers to provide a solid baseline for this task, which currently cannot be found in the literature.

We have used a total of 19 separate feature extraction approaches, 9 of them based on the usage of tf-idf techniques and 10 based on non-sparse word embeddings models. We have evaluated the extracted features on 4 separate multi-label classifiers, resulting in a total of 76 cross-validated experiments. The best result in terms of average f1-score was 54.8% and it was achived by a multi-layer perceptron classifier using tf-idf features.

The remaining of this paper is organized as follows. In Section 2 we present a collection of studies related to this one. In Section 3 we describe the dataset, the used

features, and the classifiers built for this study. Section 4 brings the experimental results, and conclusions and future directions are presented on Section 5.

2. Related Works

This section presents some of the literature about similar problems to the one in this paper. We focused on the techniques employed for classification, the nature of the datasets used, and their main results.

[Rasheed et al. 2005] approached the film genre classification problem by extracting visual features from a dataset composed of movie previews. The classifiers used in their study were based on information such as: camera movement, color variation, duration and contrast. The authors employed the Mean Shift technique to cluster the training set within the 4-dimensional feature space. 101 movie previews were used for testing the model, with 17 of them being considered as outliers. The authors also argued that it may not be intuitive to classify films using single labels, as ideally movies may be classified as several genres at once.

[Lee and Myaeng 2002] present a study that focuses on the classification of textual genres, based on a dataset with information about several web pages in English and Korean. The authors use features on two separate classification steps, first finding the overall category or genre for the document and then classifying the specific subject to which the document is about. The features used to classify each document were extracted using the frequency of terms across all documents, how uniformly distributed a term is across all subjects within the same genre and how discriminating a term is within the different document genres. The dataset used in the study was composed of 15.443 entries and the constructed classifier achieved f1-score of 0.9, improving the results previously obtained by a Naive Bayes approach, which had a f1-score of 0.75.

[Zhou et al. 2010] approach the film genre classification problem by using visual features extracted from a trailer dataset. The proposed method first split each trailer into a set of scenes by analyzing and identifying the boundaries between shots. A central frame from each scene was then used in the extraction of visual features. A K-means algorithm was used to group the trailers into clusters. The used dataset was composed of 1,239 trailers categorized in four genres, namely: action, comedy, drama, and horror. The overall accuracy of the classifier was 70%, with the best result presenting a 74% accuracy.

[Huang and Wang 2012] used a Self-Adaptive Harmony Search (SAHS) meta-heuristic to choose a feature set among 277 audio visual features extracted from movie trailers to classify them into genres. Once a feature set was chosen, it was then used to train a SVM classifier. Experiments were conducted with a dataset composed of 223 trailers from the Apple Movie Trailers website and the best result was 91.9% of accuracy.

[Austin et al. 2010] propose the use of features extracted from movies soundtracks to classify their genres. The authors collected a dataset composed of 98 movies soundtracks classified as either romance, drama, horror or action. A total of 1,728 songs were used in the study, with the central 60 seconds of each track being analyzed for feature extraction. The features were them used to train a SVM classifiers, which obtained accuracy results ranging from 60% for the action genre to 43.5% for the horror genre.

[Sugano et al. 2003] tackle the film genre classification problem by using audio

visual features extracted from movie scenes. The authors extracted a total of 7,450 scenes from 4 separate movies and used a randomly selected 347 subset of them to train a LMDT (Linear Machine Decision Tree) classifier. The scenes were manually classified as: action scene, dramatic scene, dialogue scene or generic scene. The classifier used features such as scene duration, nature of audio (silence, dialogue, other) and camera movement, presenting accuracy scores of up to 93%.

[Ivasic-Kos et al. 2015] present the results of a multi-label classifier trained with visual features extracted from movie posters in order to classify their genres. The dataset used in the study was composed of 6,739 images classified in 18 different genres. After a preprocessing stage, the number of genres was reduced to 11. Low-level visual features such as color, value, and saturation histograms were used to train three classifiers using different approaches: Naive Bayes, where the multi-label classification was reduced to several single-label instances, Random k-label sets (RAKEL) and Multi-Label k-Nearest Neighbor (ML-kNN). The best f1-score obtained by the classifiers was 0.38.

Table 1 contains a comparison between the main characteristics of the related works presented in this section, we classify works by the classification approach they've used in their classifiers, the nature of their datasets and the features they've extracted from them and some additional notes that may differentiate the study from others presented in the table.

Table 1. Comparison of the related works

Work	Classification approach	Dataset Nature	Notes
Rasheed et al. (2005)	Mean Shift	Visual	
Lee and Myaeng (2002)	Term Frequency - Inverse Document Frequency	Textual	Text genre classification
Zhou et al. (2010)	K-means	Visual	
Huang and Wang (2012)	Support Vector Machines	Audiovisual	
Austin et al. (2010)	Support Vector Machines	Audio	
Sugano et al. (2003)	Linear Machine Decision Tree	Audiovisual	Individual scene classification
Ivasic-Kos et al. (2015)	ML-kNN, RAKEL, Naive Bayes	Visual	Multi-label classification

3. Methodology

In this section we present the dataset and classifiers used in this study, along with detailed descriptions of how they were constructed and other methodological information.

3.1. TMDB Dataset

For this study, a significantly large dataset of film synopsis and genres was required. The synopses contained in the dataset should be consistent with one another, meaning that they should be extracted from a single source, have the same average word count, and display similar amounts of information about the film they are describing.
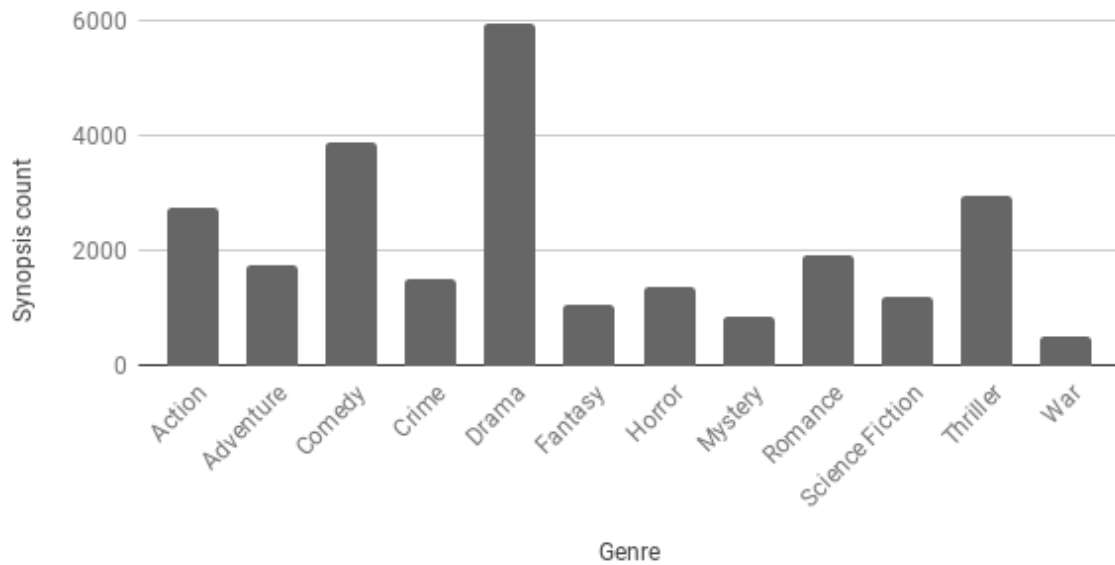
Figure 1. Genre labels frequency in the dataset after pre-processing.

We have been unable to find a dataset with those characteristics in the literature; therefore, a new dataset was extracted and processed for this study. The source chosen for extracting the synopses was the The Movie Database (TMDb) website [1].

TMDb was launched in 2008 and currently hosts over 360 thousand movie entries in 39 languages. All movies are classified with at least one of 18 distinct genres, namely: Action, Adventure, Animation, Comedy, Crime, Documentary, Drama, Family, Fantasy, History, Horror, Music, Mystery, Romance, Science Fiction, TV Movie, Thriller, and War.

The synopses were extracted in February 2018 using available TMDb APIs. Only synopses in Brazilian Portuguese were selected, totalizing $13,449$ synopses. During the pre-processing of the dataset, we discarded synopses labeled as Animation, Documentary, Family, History, Music, and TV Movie, since we considered them as not representative of film genres in the scope of this study. After pre-processing, the dataset contains $12,094$ synopses, all labeled with at least one of the 12 genres. Figure 3.1 shows the frequency of each genre in the dataset, noting that one synopsis may have several genre labels.

Additionally, we present Table 2 that contains the occurrence percentages of all genre pair combinations in the dataset. The percentage values for each of the genres are also presented, highlighted in the main diagonal. As expected, the most common genre combinations often include the genres with the highest frequency in the dataset. The most common combinations are: Drama/Thriller with a 10.66% frequency in the dataset, Drama/Romance with 10.29%, Drama/Comedy with 9.36% and Action/Thriller with 9.36%.

---

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

Table 2. Genre co-occurrence percentages

	Mystery	Romance	Horror	Crime	Drama	Fantasy	Adventure	Action	Science Fiction	Comedy	War	Thriller
Mystery	**7.03**	0.51	1.79	2.08	3.41	0.51	0.49	0.98	0.75	0.58	0.07	4.56
Romance	0.51	**16.05**	0.27	0.66	10.29	1.22	1.08	1.08	0.57	7.97	0.57	1.31
Horror	1.79	0.27	**11.33**	0.52	1.89	1.07	0.45	1.60	1.91	1.28	0.04	5.69
Crime	2.08	0.66	0.52	**12.38**	7.00	0.15	0.83	4.87	0.30	2.57	0.03	7.14
Drama	3.41	10.29	1.89	7.00	**49.40**	2.24	3.45	6.85	1.99	9.36	3.14	10.66
Fantasy	0.51	1.22	1.07	0.15	2.24	**8.82**	3.54	2.65	1.86	2.75	0.04	1.05
Adventure	0.49	1.08	0.45	0.83	3.45	3.54	**14.47**	7.68	3.10	3.88	0.55	2.61
Action	0.98	1.08	1.60	4.87	6.85	2.65	7.68	**22.85**	5.01	4.21	1.45	9.12
Science Fiction	0.75	0.57	1.91	0.30	1.99	1.86	3.10	5.01	**9.92**	1.60	0.07	3.38
Comedy	0.58	7.97	1.28	2.57	9.36	2.75	3.88	4.21	1.60	**32.08**	0.27	1.70
War	0.07	0.57	0.04	0.03	3.14	0.04	0.55	1.45	0.07	0.27	**4.11**	0.61
Thriller	4.56	1.31	5.69	7.14	10.66	1.05	2.61	9.12	3.38	1.70	0.61	**24.57**

3.2. Features

Two separate approaches of textual feature extraction were implemented in this study: one based on tf-idf (term frequency-inverse document frequency), and one based on embeddings.

The tf-idf approach extracts features from the synopses by analyzing the training set and calculating the frequency of each term (or set of terms) divided by how often the same term appears in other synopses. Once tf-idf scores are calculated for all terms in the training subset, we use a chi-squared distribution to select the most significant features. We have built nine separate feature extraction methods in this approach by combining tf-idf calculations and chi-squared thresholds: we have used tf-idf with unigrams (single terms), bigrams (pairs of terms, including all unigrams), and trigrams (triples of terms, including all unigrams and bigrams). For each of the three n-gram sets we have used the chi-squared distribution to select 50, 100, and 1000 best terms from each set.

In the embeddings-based approach, we have used pre-trained word embeddings (non-sparse vectors representing each term) to extract a vector from each individual term. Then we used the tf-idf score of each term as its weight to combine the embeddings from each singular term into a feature vector for the synopsis as a whole. We have built seven feature extracting methods using word embeddings. We tested four word embeddings models, namely: Word2Vec [Mikolov et al. 2013], FastText [Bojanowski et al. 2016], Wang2vec [Ling et al. 2015], and Glove [Pennington et al. 2014], all of them trained for Portuguese and made available by [Hartmann et al. 2017]. For Word2vec, FastText, and Wang2vec, two separate versions have been used: CBOW and Skip-Gram. Due to time and resource constraints, all embeddings used in the study had 50 dimensional vectors.

In addition to the pre-trained word embeddings models described above, three additional models were created using the Doc2Vec model described by [Le and Mikolov 2014]. In such cases, a new model has been trained and tested for each cross-validation execution. The models were created with vector sizes of 50, 100 and 1000 dimensions. In addition, all models used a window of 8 words, that is the range in which words for each sentence are considered. The models were set to ignore all words with a frequency of 5 or below in the dataset and their learning algorithm was set as 'distributed bag of words' (PV-DBOW).

All feature extraction methods described in this section included a preprocessing phase in which we removed stopwords, stemmed and tokenized the synopses.

3.3. Classifiers

The classifiers were built using scikit-learn[2] implementations. We evaluated two separate strategies for multi-label classification that are provided by the framework: one based on decision trees and one based on multi-layer perceptrons (MLP). The decision tree based classifiers were built using three different algorithms, namely: DecisionTreeClassifier, ExtraTreesClassifier, and RandomForestClassifier.

The DecisionTreeClassifier is an optimized version of the CART algorithm and implements a standard decision tree approach to classification. For this classifier, we used the following parameters: the quality of each split was measured though Gini impurity, the split strategy prioritized the best split, no maximum depth was set for the tree, all features were considered while searching for the best split, and all classes had the same weight.

The RandomForestClassifier (randomized trees) is an ensemble method that works by splitting the initial dataset in random sub-samples, which are then used to build random trees. The sub-samples are drawn with replacement and all of them have the same size of the training set. Differently of traditional decision trees, random trees inducers use a random subset of features to search for the best split each time a new split is required. After creating a forest of random trees, the resulting classification is given as the averaged prediction of the individual trees. For this classifier, we set the number of trees in the forest to 10, and the remainder of its parameters was set to the same values used for the DecisionTreeClassifier.

The ExtraTreesClassifier (extremely randomized trees) works similarly to the RandomForestClassifier, differing from the latter in the way that splits are computed. Besides using a random subset of features to search for the best split, the thresholds for a candidate feature are also drawn at random and the best randomly-generated threshold is used. After creating a forest of extremely random trees, the resulting classification is given as the averaged prediction of the individual trees. We set this classifier parameters to the same values used for the RandomForestClassifier.

All decision tree based classifiers were induced using the same random seed.

Besides the decision trees, we also built a MLP classifier. Our MLPClassifier has a single hidden layer containing 100 neurons with rectified linear unit (relu) as their activation function. The weights were estimated using a stochastic gradient-based optimizer. The learning rate was set to a 0.001 constant, and the maximum amount of iterations (epochs) was set to 200.

4. Experimental Results

Experimental results were obtained by combining the described classifiers and the features extraction methods described in Section 3.2 to the TMDb dataset. All combinations were evaluated using 5-fold cross-validation and the final classification metrics were obtained by averaging the results from each fold.

A total of 76 experiments were conducted, combining each of the 19 feature extraction approaches with the 4 classification algorithms. In each experiment, precision,

---

[2]http://scikit-learn.org/stable/

recall, and f1-score values were calculated for each of the 12 classes in the dataset. A weighted average, using the support values of each class in each specific cross-validation execution, was then applied in order to calculate precision, recall and f1-score values for each individual fold.

Table 3 presents the main results of the conducted experiments along with standard deviation values obtained from all cross-validation executions. The best results for each of the four classifiers are highlighted in each column.

Table 3. Average f1-score results for each experiment

Used features			DecisionTree		ExtraTrees		RandomForest		MLP	
Source	Type	Dim.	%	stdev	%	stdev	%	stdev	%	stdev
TF-IDF	Unigrams	50	36.0 ± 0.71		35.2 ± 0.84		34.2 ± 0.84		37.6 ± 1.14	
TF-IDF	Unigrams	100	38.2 ± 0.84		36.6 ± 0.55		35.6 ± 0.55		44.2 ± 1.30	
TF-IDF	Unigrams	1000	**39.8 ± 0.45**		32.6 ± 0.89		31.2 ± 0.45		54.4 ± 0.55	
TF-IDF	Bigrams	50	36.6 ± 0.55		35.6 ± 1.14		34.8 ± 0.84		40.0 ± 0.71	
TF-IDF	Bigrams	100	37.8 ± 1.10		**37.0 ± 1.00**		35.6 ± 1.14		45.0 ± 0.71	
TF-IDF	Bigrams	1000	39.4 ± 0.55		34.4 ± 0.55		32.0 ± 0.71		54.6 ± 0.89	
TF-IDF	Trigrams	50	36.4 ± 0.89		35.2 ± 0.84		34.6 ± 0.55		41.6 ± 0.55	
TF-IDF	Trigrams	100	37.6 ± 0.89		36.4 ± 0.55		**36.0 ± 0.71**		45.6 ± 0.89	
TF-IDF	Trigrams	1000	39.0 ± 0.71		34.6 ± 0.55		32.0 ± 0.71		**54.8 ± 0.84**	
Word2Vec embeddings	CBOW	50	28.2 ± 0.84		14.2 ± 0.45		14.4 ± 0.55		13.0 ± 1.00	
Word2Vec embeddings	SKIP-GRAM	50	28.2 ± 0.45		14.2 ± 0.84		14.6 ± 0.55		13.4 ± 1.14	
FastText embeddings	CBOW	50	28.0 ± 0.71		14.4 ± 0.55		14.6 ± 0.55		15.0 ± 1.58	
FastText embeddings	SKIP-GRAM	50	28.4 ± 0.55		14.8 ± 0.45		15.4 ± 0.55		12.4 ± 1.52	
Wang2Vec embeddings	CBOW	50	28.0 ± 0.00		13.8 ± 0.84		14.6 ± 0.55		12.2 ± 1.64	
Wang2Vec embeddings	SKIP-GRAM	50	28.4 ± 0.55		14.6 ± 0.55		14.8 ± 1.10		12.0 ± 1.87	
Glove embeddings		50	28.0 ± 1.00		14.6 ± 0.55		15.2 ± 0.45		13.6 ± 1.14	
Doc2Vec embeddings		50	27.6 ± 0.89		16.2 ± 0.45		15.4 ± 0.55		14.2 ± 0.84	
Doc2Vec embeddings		100	27.6 ± 0.55		16.2 ± 0.45		15.6 ± 0.89		12.4 ± 1.67	
Doc2Vec embeddings		1000	28.0 ± 0.71		16.2 ± 0.84		15.4 ± 0.55		12.8 ± 1.48	

By analyzing the results of Table 3 we can notice a clear trend in which all the methods that make use of tf-idf in their feature extraction phase have outperformed the ones that have applied the embeddings based approaches. It is also noticeable that, while the tf-idf based approaches have better classification results when using the multilayer perceptron classifier, in the case of experiments conducted using embeddings, the DecisionTree classifier has outperformed the others.

In both cases, the way that the features are extracted and their nature clearly affected how they may be used by their classifier. We theorize that the features extracted using tf-idf can absorb more meaningful domain information due to being extracted directly from the training sets, and that extraction leads to more complex patterns that are not easily codified by decision trees. On the other hand, the pre-trained word embeddings may have a more consistent representation of the terms contained in them, leading to decision trees being able to infer classification rules more easily, but don't have enough domain-specific information due to being pre-trained on other data. Regarding the experiments that use Doc2Vec embeddings created from the training data, they present the same trend of being better suited for use with decision tree classifiers and at the same time do not show significant improvements compared to the others embedding based approaches. That fact may be attributed to Doc2Vec embeddings being trained in a much smaller amount of data than the pre-trained word embeddings, although one may also theorize that embeddings, as they are being applied to classification in this domain and trained on

a relatively small dataset, might simply not be able to outperform other common feature extraction approaches such as tf-idf.

Table 4 presents the detailed classification results for the experiment with the highest average f1-score while using the MLP classifier. The table shows the precision, recall, f1-score and support values obtained by the simple averaging of the 5 folds used in the cross-validation for each genre in the test set.

Table 4. Averaged classification results for the overall best experiment

Genre	precision (%)	recall (%)	f1-score (%)	support
Action	57.0	64.0	60.2	553
Adventure	48.8	45.6	47.0	350
Comedy	61.6	42.8	50.4	776
Crime	52.0	48.2	50.0	299
Drama	69.0	64.8	66.6	1195
Fantasy	46.8	38.8	42.2	213
Horror	57.8	57.6	57.6	274
Mystery	36.2	24.6	29.0	170
Romance	51.6	46.0	48.6	388
Science Fiction	54.2	58.4	56.0	240
Thriller	54.4	54.6	54.4	594
War	51.0	60.0	55.0	99

By calculating the weighted average between the values across all genres, the results obtained from the best experiment show a precision of 57.61%, recall of 53.36% and f1-score of 54.80%.

Figure 4 shows the relationships between the results presented above and the genre frequencies presented in Table 2. The best results were obtained for Drama, which is the genre with the biggest support in the dataset. For the remaining genres, however, such relation cannot be easily established. As some of the genres with the lowest support values still present f1 values comparable with other higher frequency genres. It is also worth noting that the results for genres such as "Science Fiction" and "War" present themselves in the within the higher f1 values, even as these genres rank between some of the lowest support values in the dataset. These facts lead us to believe that, while the genre's frequency in the dataset may alter the classification results, the extracted features may contain significant genre information that allows their identification despite their frequency in the dataset.

Considering the averaged f1-score of this experiment, "Mystery" presented itself as an outlier. Its f1-score was 29.0%, while for all the other genres the f1-score were in the 42.2% to 66.6% interval. This can be attributed to the genre's low frequency in the dataset of 7%, it is important to note however, that genres such as "War" and "Fantasy" that have frequencies of 4.1% and 8.8% respectively have both displayed higher f1 values than "Mystery", which could imply that other genre specific synopsis characteristics might affect the quality of the classification.
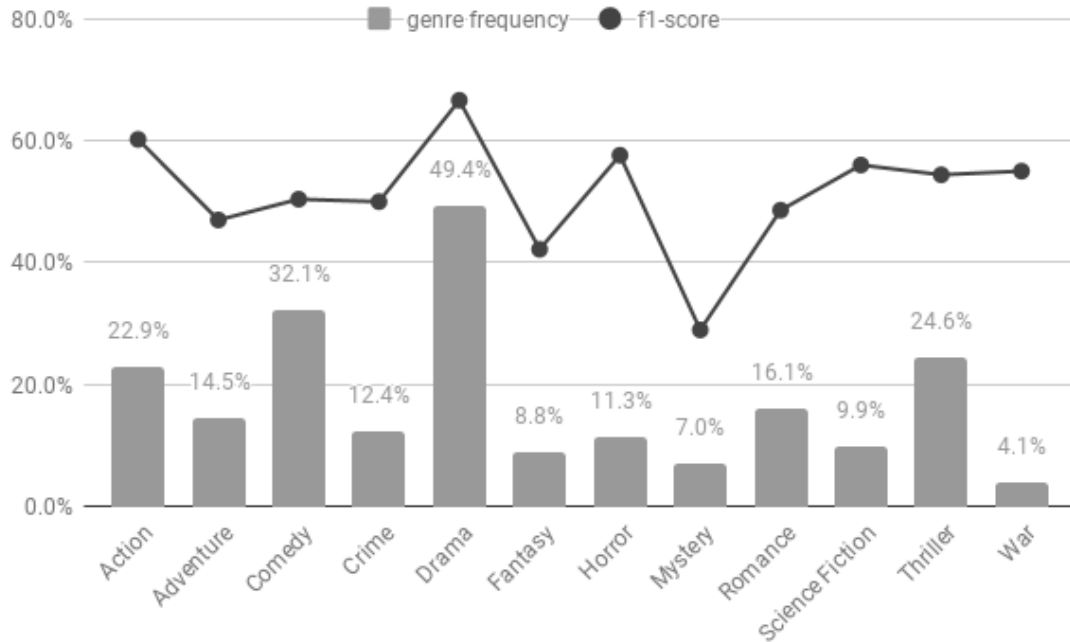
Figure 2. Relationship between frequency and f1-score for each genre in the experiment with the best overall result

5. Conclusion and Future Work

In this study we have presented baseline results for the classification of film genres using textual features and multi-label classifiers. We have executed 76 experiments, combining 19 textual feature extraction approaches with 4 separate classifiers.

We obtained the best results by using features extracted by tf-idf approaches and using a Multilayer Perceptron for the classification. The results of the best experiment present the following averaged scores between all 12 genres present in the dataset: precision of 57.61%, recall of 53.36% and f1-score of 54.80%. We have noticed that none of the embedding based feature extraction techniques was able to outperform the tf-idf approaches.

While we cannot directly compare our results to those presented by related works due to the novelty of the usage of textual features in this domain, we note that we achieved scores that are substantially higher than the ones in [Ivasic-Kos et al. 2015], where a similar multi-label movie classification approach presented f1-scores of up to 0.38 by using visual features extracted from movie posters.

This study presents a positive outlook on the film genre classification problem, since the 54.8% f1-score obtained in the best experiment could be further improved in future works where more specific features and classifiers are used. The authors of this paper hope that it may provide a substantial comparison baseline that can be used for the development and improvement of synopsis based film genre classification methods.

As future works, we intend to continue studying the domain and the dataset by focusing on extracting features more specialized to the film genre classification problem,

which can then be compared to the baseline ones presented in this study.

6. Acknowledgements

References

Austin, A., Moore, E., Gupta, U., and Chordia, P. (2010). Characterization of movie genre based on music score. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 421–424. IEEE.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., and Aluísio, S. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In *Proceedings of Symposium in Information and Human Language Technology*, pages 122–131. Sociedade Brasileira de Computação.

Huang, Y.-F. and Wang, S.-H. (2012). Movie genre classification using svm with audio and video features. In *International Conference on Active Media Technology*, pages 1–10. Springer.

Ivasic-Kos, M., Pobar, M., and Ipsic, I. (2015). Automatic movie posters classification into genres. In *ICT Innovations 2014*, pages 319–328. Springer.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lee, Y.-B. and Myaeng, S. H. (2002). Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 145–150. ACM.

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Rasheed, Z., Sheikh, Y., and Shah, M. (2005). On the use of computable features for film classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):52–64.

Sugano, M., Isaksson, R., Nakajima, Y., and Yanagihara, H. (2003). Shot genre classification using compressed audio-visual features. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II–17. IEEE.

Zhou, H., Hermans, T., Karandikar, A. V., and Rehg, J. M. (2010). Movie genre classification via scene categorization. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 747–750. ACM.