Multimodal movie genre classification

Konstantinos Mavromatakis
Gothenburg University
gusmavko@student.gu.se

Abstract

Multimodal deep learning relates data of multiple modalities so as to generate a desired output. For this project, we use various combinations of such data (specifically, language and visual) to investigate which combination of features is likelier to predict more accurately the genres a movie is associated with. A question we seek to answer is posed by Dannélls and Virk (2020)

> "Do we always need many features to achieve better results? "

1 Introduction

Multimodal processing involving different modalities, such as text, image and speech, has shown great progress thanks to deep learning over the past few years (Deng, 2016). Data of various modalities carry substantial semantic information and naturally, it is expected that combining them can potentially yield better results than using unimodal data. A quite common method in this field is to concatenate multimodal features to get a final representation that will be used as input to a model (Arevalo et al., 2017). There is also the possibility to pass the features of each modality though a fully connected layer so that they have the same shape and then sum them up (Vinyals et al., 2015).

Movie genre classification is a really popular classification task that has been researched to a great extent. Both unimodal and multimodal approaches have been explored the past years. Zhou et al. (2010) divide trailers into shot units and classify each scene into a genre using shot analysis. Chu and Guo (2017) use film posters and by focusing on what appears on the image they train a Convolutional Neural Network (CNN) for film genre classification. Ertugrul and KARAGOZ (2018) claim there is a hidden representation of genre information in the movie plot summaries. They estimate a genre for each sentence of a movie summary separately and, then, use majority voting for the final decision by considering the posterior probabilities of genres assigned to each sentence. Nakano et al. (2019) combine screenplay content and structure to train a Support Vector Machine to predict genre labels. Lastly, Arevalo et al. (2017) perform movie classification by combining poster and plot information and argue that the multimodality of the data seemed to improve the results of their model.

The proposed model builds on the previous approaches by using multimodal data as features, but goes further to investigate other kinds of textual data that have not been explored extensively, such as movie titles and production companies.

In Section 2, we explain the materials and methods used in this movie genre classification task. Section 3 presents the results of the different versions of our model, which are then discussed and analysed. Finally, Section 4 draws the conclusions and ideas for future work.

2 Materials and methods

2.1 Materials

2.1.1 The Movies Database

The Movies Dataset[1] includes approximately 45,000 movie titles – only half of which were used for this project – and various metadata associated with each movie, such as overview, movie id on TheOpenMovieDatabase[2] website (TMDB), genres, production companies, rating, runtime, budget and original languages among others. For the task in question, summaries, genres and production companies functioned as the textual input for the model. Movies that were classified as a non generally accepted genre (e.g. 'TV Movie', 'Carousel

[1] https://www.kaggle.com/rounakbanik/the-movies-dataset/kernels

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

Productions', 'Aniplex') and non English titles were dropped reducing the number of movie genres to 19 – the genre distribution is shown in figure 1.

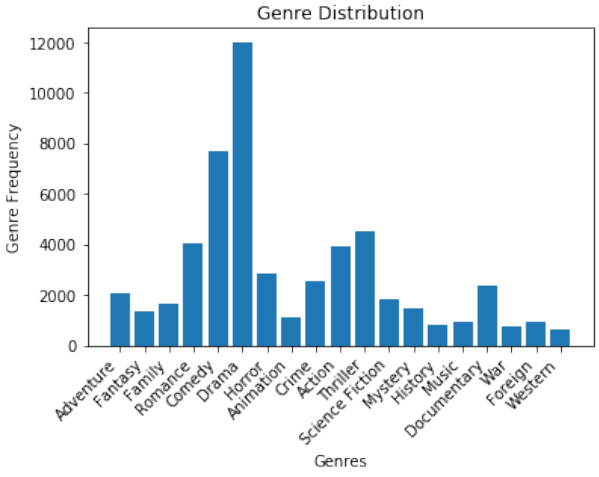

Figure 1: Movie genre distribution in the OpenMovies-Database

2.1.2 Movie Posters

Movie posters were obtained using the TMDB API. Movies that did not have any poster available were automatically discarded from the dataset.

2.2 Methods

2.2.1 Text Preprocessing

A lot of thought was put into text preprocessing. Regarding the cleaning up part, punctuation and digits were removed since they do not contribute substantial semantic information in the context of movie genre classification. After tokenizing with the RegExpTokenizer, the text was lowercased and lemmatized to limit the size of the vocabulary. Stopwords were also removed from the overview text since they also do not seem to provide any semantic meaning. However, because of certain titles only consisting of function words, such as 'Who am I', the stop words were not removed from the titles.

2.2.2 Encoding Language Data

The language data – i.e. the overviews, the titles and the production companies – were transformed to integers and padded to the length of the longest corresponding sequence using Keras. A maximum of 50000 words was kept based on word frequency – less frequent words were treated as unknown tokens and were not included in the vocab. The integer representations were transformed to tensors with the TensorDataset provided by Pytorch. Subsequently, iterators were created that return the movies batch by batch with Pytorch's Dataloader.

2.2.3 Image Preprocessing

The movie posters were resized to 100*100*3 dimensional arrays that were subsequently transformed to tensors and saved locally.

2.2.4 Model

The model, which was implemented in Pytorch, uses both language and image features to make predictions. The textual sequences – the overview, the movie title and the production companies – are represented with pretrained word2vec embeddings (Mikolov et al., 2013) of 300 dimensions and then concatenated along the same dimension. Then, the concatenated output is passed through two stacked LSTM layers. Given the fact that the longest textual sequence is 879 tokens long, an LSTM was chosen to act as a memory storage. LSTMs can regulate the flow of information by learning which data should be stored and which data should be forgotten. Thus, it can pass relevant information down the long chain of sequences to make predictions, which renders it an excellent choice for the task in question.

Regarding the image features, we loosely based our model's architecture on the VGG16 architecture – we decided not to use all 6 convolutional blocks from the original paper (Simonyan and Zisserman, 2015) or the pretrained visual features to get acquainted with training our own. Our Convolutional Neural Network (CNN) consists of 4 consecutive 2D convolutional and max-pooling layers. All hidden layers are equipped with the rectification (ReLU) non-linearity allowing complex relationships in the data to be learned. The output of the last max-pooling layer is flattened and passed as input to a fully connected layer.

The features that were used as input to our model are the following:

- Movie overview
- Movie poster
- Movie overview and poster
- Movie title and poster
- Production companies of movie and poster
- Movie overview, title and poster

	Accuracy concat	AUC-ROC concat	Accuracy sum	AUC-ROC sum
overview and poster	0.8645	0.6914	0.8634	0.6845
companies and poster	0.8438	0.6036	0.8315	0.6367
title and poster	0.8363	0.6310	0.8422	0.6490
title, overview and poster	0.8310	0.6395	0.8464	0.6395
overview, title, companies and poster	0.8296	0.6227	0.8366	0.6361

Table 1: Accuracy and Area Under the ROC Curve (AUC) score on the test data using different features and different combinations of features

- Movie overview, title, production companies and poster

When combining the representations of different modalities, we evaluated two different methods: (1) similarly to Suk and Shen (2013), we concatenate the visual and the textual representations across the same dimension; (2) we also evaluate the summation of the language and visual features after passing them through fully connected layers to have them in the same space (Vinyals et al., 2015). Concatenation keeps all information where they are, while summation aggregates the data, reducing their size and inevitably leading to the loss of some information. The architecture of the model can be seen at figure 2.

Finally, it should be noted that since this is a task of multi label classification and getting an exact match between the ground truth and our model's prediction is quite challenging, we decided to not evaluate our model with subset accuracy. Conversely, we consider a prediction as accurate if at least one label is predicted correctly. Another evaluation metric we employ is Area Under the Receiver Operating Characteristics (AUC-ROC) to show to what extent a model is capable of distinguishing between classes. AUC-ROC is an excellent metric for multi label classification tasks, especially when there is a class imbalance in the data (Herrera et al., 2016). An AUC-ROC score closer to 1 means that a model has a good measure of separability while a score closer to 0.5 denotes that there is no particular capability of distinction of classes (a score near 0 indicates that a model has the worst measure of separability).

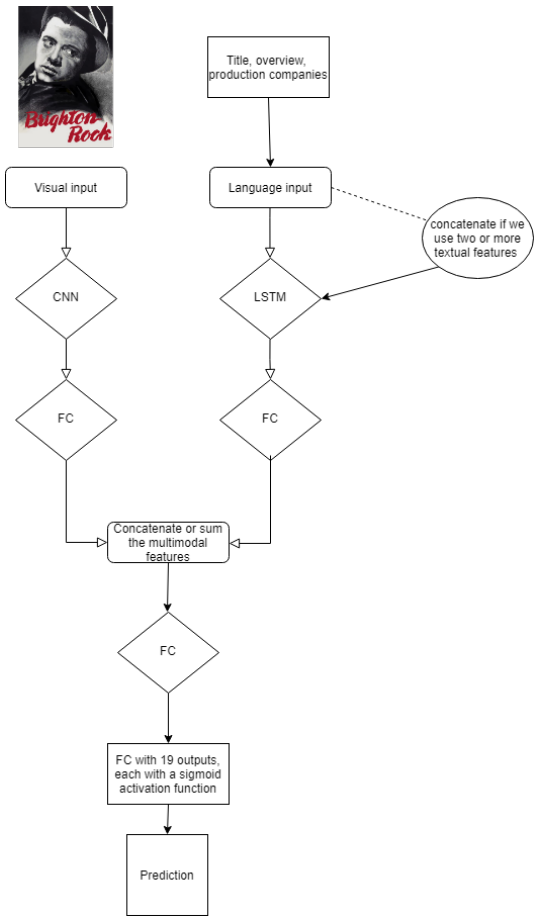

Figure 2: Architecture

3 Results and Discussion

We evaluated our model with unimodal features to use as a benchmark. The accuracy and the AUC-ROC score were 0.8025 and 0.5978 for the visual features and 0.8239 and 0.6004 for the language features (specifically, the overview text), which shows that text seems to carry more substantial semantic information than images.

With these scores as a benchmark, we then carried out experiments with combinations of various language and visual features in order to determine if particular language features carry more substantial information than others that can contribute to a more accurate classification.

As mentioned in Section 2, the language and visual features were combined by (1) concatenation and (2) summation. The former method yielded

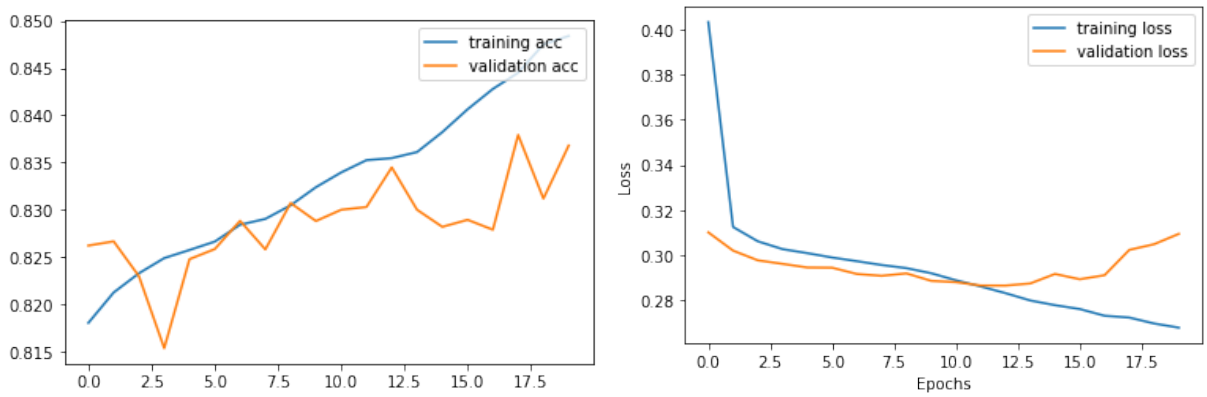

Figure 3: Accuracy (left) and loss (right) plots for the model that used the summation of the titles and movie posters as input.

better results when fewer features were used. Combining image with just one of the three textual features (e.g. overview, production companies or title) through concatenation seems to have a better performance than using many textual features (e.g. title, overview and company along with the poster). A possible explanation for this is that longer sequences, which are a result of concatenating the multiple padded textual inputs, are harder for an LSTM to keep in its memory. Similar are the results for the summing method, which can be seen in table 1. Most of the models that used only two features (e.g. overview or title and image) performed better than the models using many features (e.g. title, overview and company along with image). However, the summation of the production company representation and the image representation gave the lowest accuracy (0.8315) and the second lowest AUC-ROC score (0.6367) out of the models which used summed input. There seems to be a pattern across the previously mentioned performances – a combination of many different features does not guarantee better results. Concatenating the overview text and the image representations resulted in a higher accuracy (0.8645) and AUC-ROC score (0.6845) in comparison to their summation (0.8634 and 0.6845 respectively). This is the best performance out of the models we trained, which perhaps indicates that the film overview is the most informative out of all the textual features. The concatenation of production movies and images gave a quite promising accuracy score of 0.8438 and a respectable AUC-ROC score of 0.6036, which is perhaps related with the fact that particular production companies tend to be associated with producing movies of certain genres (Vertigo Films[3], for example, are well known for producing horror films and thrillers). The summation of companies and posters, on the other hand, yields an accuracy of 0.8315 and an AUC-ROC score of 0.6367. Movie titles seem to be equally informative – summing and concatenating them with the movie posters yields an accuracy of 0.8422 and 0.8363 respectively. Interestingly, when using a greater number of features, summing instead of concatenating seems to improve the model's performance. Specifically, a combination of movie overview, title, production companies and movie poster yields an accuracy of 0.8464 and 0.8310 respectively – accordingly, the AUC-ROC score is higher for the model using the summed features than for the model using the concatenated features. These can be seen in table 1.

An example of the accuracy score and loss of one of our models - which received the film titles and the image posters as input – can be seen in figure 3. The loss for the training data diverges with the loss for the validation data after approximately 12 epochs of training proving that the model is learning to generalize on the validation set. Expectedly, the accuracy on the training set is quite high while the accuracy on the validation set tends to be lower.

It is also worth noting that all models mostly predicted one or two labels per movie on the test set, which reflects the training data.

To evaluate the results of our models, we decided to investigate the confusion matrices of each genre across our different models. Certain genres' metrics scores seem to carry quite interesting information (the confusion matrices for the drama genre

[3] https://vertigofilms.com/

Movies	Correct labels	Overview	Title	All features	Production company
Brighton rock	crime, thriller, drama	crime, thriller, drama	thriller, drama	-	-
Coming soon	horror, thriller	horror, thriller	-	comedy	-
Beethoven's 2	family, comedy	-	comedy	family, comedy	family, comedy, action, adventure, music

Table 2: Ground truth labels and predictions across the different models

are available at Appendix A).

Drama is the film genre with the lowest accuracy across all models. The low accuracy has to do with the large number of false positives and false negatives in the prediction. Since it is the most frequent film, our model seems to have a bias towards assigning the drama label to movies. The highest accuracy out of all 5 models – with almost 0.1 difference – seem to be the model that combines the overview and the poster. Perhaps, the increased accuracy is related with the fact that dramas are one of the most frequent film genres and while it is hard to convey drama in the title, drama can definitely be portrayed in a longer sequence of text, such as a movie summary. The ground truth labels and the predictions can be found in table 2

Taking a closer look at one of the movies in our data, *Brighton Rock*, a crime, thriller and drama film, is labeled as a thriller and a drama when using title and image as input, but not as a crime film. However, the model was able to predict all 3 labels correctly when using the overview and the image as features, confirming that longer sequences of meaningful text are a more informative feature than the titles. The cleaned overview text for that movie contains quite many key-words that could help with the correct prediction of the crime genre (e.g. 'gang', 'criminal', 'vicious', 'interwar', 'underbelly'). The poster and the overview of the movie can be found in Appendix B. Another genre that scored rather low accuracy is comedy. The model that uses overview, title, production companies and poster displays the lowest accuracy (0.5781). Similar to the drama genre, comedy, the second most frequent genre in the dataset, is assigned quite frequently to movies resulting in a greater number of false positives because of the existence of bias towards the comedy genre in the data. A possible explanation for this could be that the features that are used are quite different between them. A movie may not have a very funny title, so the comical content is more likely to be extracted from the overview. As a result, such contradictory information may confuse the model. On the other hand, the highest accuracy (0.711) is scored by the model that uses only the overview and the poster, which seems to prove the point we made above – that the overview carries substantial semantic information contributing to the prediction of the desired output.

Quite interesting are the metrics for the thriller genre. The lowest accuracy (0.7391) comes from the model that uses the titular and the visual features as input. The model with the best performance (0.7901) is the one that combines all available features, something which can be easily explained if we take into consideration that both title and overview carry important semantic information and that certain companies specialize in producing movies of such a niche movie genre.

Looking at one of the movies in the test data, *Coming Soon*, explains the low accuracy of the model that uses all possible features as input in the comedy genre. *Coming Soon* is a horror and thriller movie but it was classified as a comedy by this model. This tendency, which can be explained by the number of comedies in the dataset, increases the number of false positives. On the other hand, the model that uses only the overview and the poster predicted both labels correctly without misclassifying this movie as a comedy. Its summary contains some words that could function as indicators for its actual genres (e.g. 'horror', 'scare','ghost' etc.). Its poster and overview can be found in Appendix C.

Another case that is worth mentioning is the movie *Beethoven's 2*, whose genres are comedy and family. While the prediction of the model that uses all features is an exact match, the model that uses the production company and the poster as input predicts the two genres correctly, however it also assigns 3 additional labels to the movie. The model with the least success though is the one that uses the title and the poster by only predicting the comedy genre right. Words that are included in the overview, such as 'fathered','family', 'kid', could potentially help predicting the family genre. The poster can be found in Appendix D.

4 Conclusion

In this paper, we combined data of different modalities in various ways to perform a classification task. We reached the conclusion that the method we combine them may affect the performance of the model to a certain extent. Also, particular features seem to have a tendency of being more informative than others. Finally, to give a preliminary answer to the question we cited in Section , our experiments showed that sometimes fewer features can achieve better results.

In the future, we would like to explore the use of contextual word embeddings on the textual data, so as to assign each word a representation based on its context, a method which has shown to improve performance in various NLP tasks.

We would also want to investigate if using pre-trained image features would improve the results.

On top of that, it would be of interest to explore more parameters for training. Perhaps, actors or directors could be a good indicator for movie genre prediction since it is quite common for both to be associated with films of a particular genre.

References

John Arevalo, Thamar Solorio, Manuel Montes y Gómez, and Fabio A. González. 2017. Gated multimodal units for information fusion.

Wei-Ta Chu and Hung-Jui Guo. 2017. Movie genre classification based on poster images with deep neural networks. In *Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes*, MUSA2 '17, page 39–45, New York, NY, USA. Association for Computing Machinery.

Dana Dannélls and Shafqat Virk. 2020. Ocr error detection on historical text using uni-feature and multi-feature based machine learning models.

Li Deng. 2016. Deep learning: from speech recognition to language and multimodal processing. *APSIPA Transactions on Signal and Information Processing*, 5.

Ali Mert Ertugrul and Pinar KARAGOZ. 2018. Movie genre classification from plot summaries using bidirectional lstm.

Francisco Herrera, Francisco Charte, Antonio J. Rivera, and Mara J. del Jesus. 2016. *Multilabel Classification: Problem Analysis, Metrics and Techniques*, 1st edition. Springer Publishing Company, Incorporated.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Yusuke Nakano, Hiroaki Ohshima, and Yusuke Yamamoto. 2019. Film genre prediction based on film content and screenplay structure. In *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications amp; Services*, iiWAS2019, page 151–155, New York, NY, USA. Association for Computing Machinery.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Heung-Il Suk and Dinggang Shen. 2013. Deep learning-based feature representation for ad/mci classification. volume 16, pages 583–90.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator.

Howard Zhou, Tucker Hermans, Asmita Karandikar, and James Rehg. 2010. Movie genre classification via scene categorization. pages 747–750.

Appendix A Confusion Matrices for the Drama Genre

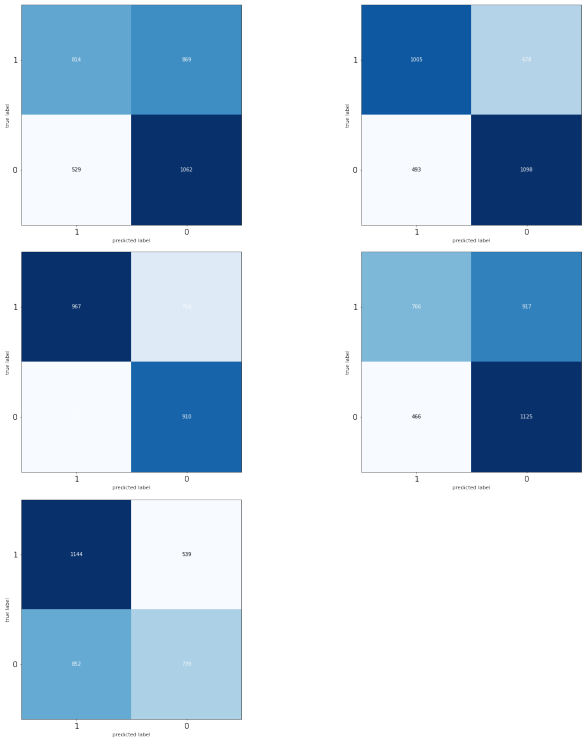

Figure 4: Confusion matrices for the drama movie genre across the different models. From left to right: (a) title and poster, (b) overview and poster, (c) production companies and poster, (d) title, overview and poster, (e) title, overview, production companies and poster.

Appendix B *Brighton Rock* Movie Poster and Overview

Figure 5: centring activity gang assorted criminal particular leader vicious young hoodlum known pinkie film main thematic concern criminal underbelly evident interwar brighton

Appendix C *Coming Soon* Movie Poster and Overview

Figure 6: kind scene horror film scare ghost appears totally unexpectedly main character see ghost sneaking behind end find main character actually ghost along none compare feeling arriving home alone suddenly stuck feeling dejavu

Appendix D *Beethoven's 2nd* Movie Poster and Overview

Figure 7: beethoven back time whole brood he met canine match missy fathered family problem missy owner regina want sell puppy tear clan apart beethoven newton kid save day keep everyone together