Movie Recommendation Using Emotion Analysis & Topic Modeling

C. Lukwesa, D. Mooren, H. Shahoud, M. Abdulfattah

1 Introduction

The field of recommendation systems e.g. for movies, music, and other products, has been improving rapidly in recent years, thanks to advances in computing that allow for storing enormous amounts of data on web users as well as innovations in Machine Learning. There are two main approaches to recommendation systems: A non-personalized approach recommends the most popular products to any user based on predefined scores, whereas a personalized recommendation approach utilizes the data/behavior of users with similar behavior to make recommendations (Aamir and Bhusry, 2015). Both approaches have been successfully applied for commercial purposes and they involve implementing data mining and dynamic machine learning algorithms to learn recommendations.

Where these two approaches differ is in the amount of control the user has over the system. This project is an attempt to develop a recommendation system using the non-personalized recommendation approach. More specifically, we apply unsupervised machine learning to do three high level Natural Language Processing (NLP) tasks: Sentiment (Emotion) Analysis, Topic Modeling and Named Entity Recognition (NER), which are subsequently used to build a movie recommendation system. The following sections formulate the motivation behind the project, introduce the used data set, discuss the applied approaches and present the results. Furthermore, we discuss the limitations of the applied approach and suggest potential improvements for future research.

2 Motivation

The personalized recommendation approach has multiple commercial advantages, as it dynamically adapts user's recommendations. There are five different personalized recommendation techniques one could use (Burke, 2002); these include collaborative filtering, demographic filtering, utility-based and knowledge-based recommendation. However, these techniques seem to be unable to grasp the user's true preference since they merely make inferences(Wakil et al., 2015).

Therefore, in this project, we try to give the user more control over what is recommended to them by explicitly accounting for how they would like to feel, instead of relying on inter-user data. User emotions are ever changing and, as such, a user's movie preferences may also change in tandem. A system based on profiles of other users is incapable of capturing these fluctuations in emotions (Wakil et al., 2015). In addition, we wanted to provide the user with the possibility of exploring more movies which involve topics that are similar to ones contained in any other movie that the system initially recommends (based on a selected emotion). This would give the user more freedom to discover a wider range of movies.

3 Data Set

This project makes use of the Movie Plot Synopsis with Tags (MPST) data set found on Kaggle. It contains about 14000 instances of movies, and each instance contains information pertaining to the movie's title, genres, and an extended plot synopsis. In this project, we perform the above mentioned NLP tasks on the plot synopsis text of a movie to derive both emotions and topics associated with that movie. We read the data using Pandas. When cleaning the data we discovered that the MPST data set contained no null or invalid values. For the purpose of this project, we used only the title, IMDb ID and the plot synopsis information.

4 Approach

As mentioned earlier, there are several high level text mining techniques needed in order to realize the user application for our movie recommendation system. First, there is the issue of deriving emotions from the plot synopsis so that we can provide the user with movies based on their emotion choice preference. Secondly, we want the user to be able to select a movie for which they want movie titles with similar topics. As will be discussed in this section in detail, the text mining techniques necessary for this are: emotion analysis, topic modeling and named entity recognition. For each of these techniques, we made use of unsupervised transformer models, namely different BERT models retrieved from Hugging Face.

4.1 Emotion Analysis

For this task, we made use of the DistilBERT model to analyze the emotional content of movie plot synopses. According to (Sanh et al., 2019), this model is a smaller and more efficient version of the generic BERT model. In their research, the authors were able to reduce the size of the generic BERT by 40%, while retaining 97% of its language processing capabilities. To properly utilize the DistilBERT model, we first partitioned the plot synopsis of each movie into sentences using the spaCy library. This was required since the used model accepts strings with 512 characters maximally. Then, we fed each sentence to the fined-tuned DistilBERT classifier to detect emotions. This model returns scores for mainly six emotions: joy, anger, fear, sadness, love and surprise. The emotions scores sum up to 1, with a higher score for an emotion indicating a greater presence of that emotion in the movie's plot synopsis.

Thereafter, we averaged the emotion scores for all sentences of each plot synopsis. This was needed due to the maximum text length limitation mentioned earlier. We then stored the obtained emotions and their values together with the corresponding movie instance in our data so that we could later use this information in the user application. In order to better utilize the new emotion scores for the task of recommendation, we stored the score of each individual emotion as a separate column in our movies CSV file. Consequently, when the user selects emotions in the application, these columns are sorted in descending order and the sorted list of movies is returned, with the top movies having the best-matching emotion scores.

4.2 Topic Modeling and Named Entity Recognition

In order to provide the user a wider range of movies with the same topic, we needed to form clusters of movies with similar topics. Since there are no labels in the data, a generative approach was needed, which led initially to the use of Latent Dirichlet Allocation (LDA) (Zhang and Zhang, 2022). This model, however, performed poorly on our data set and yielded nonsensical topic words for the derived topics.

We subsequently decided to perform the clustering of movies by first generating word embeddings for each movie synopsis using a transformer model. Because the embeddings for each synopsis had a very large number of dimensions (768), which would have likely resulted in poor clustering results (Grootendorst, 2020), we used another package, UMAP, to reduce the dimensionality of the embeddings to five. From our testing, this seemed to be the optimal number for retaining enough aspects about each synopsis to yield good movie clusters.

For clustering, we initially used the sklearn HDBscan package, which clustered the movies based on the euclidean distance between the word embeddings of each movie synopsis. We also made use of tf-idf scores across clusters, by treating each cluster of movies as a single document, to generate the most important words for each cluster, i.e. the topic words of the cluster. However, we discovered that, in addition to the majority of movies being classed as outliers (i.e. not belonging to any cluster), several of the topic words generated by this method consisted only of names of actors and/or characters from the plot synopsis. This may have been due to the fact that distinct names do not tend to occur in multiple synopses and, because of the use of tf-idf scores, are therefore seen as important words.

Because as many as 9000 movies, out of our data set of about 14000, were not being assigned to any cluster, this would have resulted in our recommendation system, which aims to present the user with movies similar to a selected one, performing poorly. To remedy this, we opted to use k-means clustering, which assigns every single instance to some cluster. For this, we used a preset number of 20 clusters. However, the issue of some clusters consisting of mainly actor and character names as topic words was still present with k-means. We decided to deal with this by using Named Entity Recognition to identify then remove these names from the synopses.

For the Named Entity Recognition (NER), we made use of another transformer model, a BERT NER model, which assigns IOB tags to entities (Sang and De Meulder, 2003). We then removed all entities labeled with 'B-PER' or 'I-PER' tags so that the synopsis texts used for clustering would not contain any identified person names, whether actor or character. Word embeddings of the resulting synopses, i.e. with person names removed, were then clustered using k-means into 20 different topics. Each cluster therefore represents movies that have similar underlying topics.

5 Application

We created an application that recommends movies based on the emotions, e.g. 'joy', that the user wishes to feel when they watch a movie. Additionally, the app is intended to allow the user to find movies similar to any other recommended movie based on topic similarity. The functionality of this user interface relies on the aforementioned text mining techniques. Upon opening the application, the user is presented with a menu of six emotions from which they can select at most three emotions that they want a movie to be associated with. This is done by means of a checkbox (Appendix A, Figure 2). The selection of emotions is ordered, with the first selected emotion being ranked the most important for retrieving results: this is indicated to the user through the display of a number e.g. '1' to indicate the first choice, in front of each selected emotion.

Based on the emotion analysis we performed beforehand on our data set of movie synopses, the application retrieve movies with the highest scores on the selected emotions, e.g. if a user picks 'Anger' first and then 'Joy', it will first look for the movies with the highest anger rating and then, within that subset of movies, look for the highest joy rating. The number of movie recommendations that are displayed in the app is determined by the recommendation limit set by the user, via a drop-down menu displayed next to the emotions menu, with 10 as the default.

Movies matching the user's preferences are displayed as a list, with each movie's title shown together with a clickable link to the movie's IMDb page, allowing the user to find more information about a recommended movie (Appendix A, Figure 3). In addition to this feature, there is a button displayed with each movie's information which, when the user clicks on it, shows a side panel with a list of movies containing similar topics to the selected movie (Appendix A, Figure 4). This similarity is based on the aforementioned topic modeling, with movies in the same cluster as the selected movie being treated as similar to that movie.

6 Results

Because we created a user-oriented application, the best way to analyze the performance of both our emotion analysis and topic modeling systems would be to get user feedback on the movies that the app recommends to them. This would involve, for example, designing a questionnaire that asks the user, after they have presumably watched the movies that the app recommends to them, to rate how well the recommended movie matched the emotions they selected in the app as well as whether any suggested similar movies were actually related to their initial selected movie. However, because we cannot perform such evaluation for this project, we instead rely on our own qualitative analyses of the results obtained from both the emotion analysis and topic modeling.

For the emotion analysis part, the first step to evaluate the performance of our system was to specify a set of movies which we have watched and are present in the data set. Subsequently, we performed the emotion analysis on them and retrieved the respective scores. We then compared, discussed and researched whether these scores were accurate and consistent, based on our own viewing of these movies. Evidently, the comparison and discussion were subjective and depended upon our interpretation and understanding of the selected movies. Moreover, we looked at the genres of a subset of these movies, read their plot synopses and then compared this information with the six emotions the model yielded to see whether there was an overlap. This was straightforward since most genres are easily identified and the plot synopses are descriptive enough to be related to the basic, predicted emotions.

For example, the highest emotion scores for the "Forrest Gump" movie were: *joy* $\approx$ *0.49* and *anger* $\approx$ *0.18*. And because we had previously watched this movie, we confirmed that these scores were fairly representative of the overall emotional content of the movie. Moreover, reading the plot synopsis, on which the classifier runs, we could see joy, optimism and frustration as the main themes. This interpretation aligned with the scores of the classifier. See the "Forrest Gump" plot synopsis in the dataset.

In order to gauge the performance of our topic modeling, we derived the top 10 topic words for each cluster of movies, i.e. common words that appear across multiple synopses in the cluster and are therefore likely used by the clustering algorithm to determine that different synopses belong to the same topic. Overall, there does appear to be a clear theme across every set of topic words; for example, one cluster has the topic words 'earth', 'planet', 'space', 'alien', etc. and another cluster groups 'police', 'gang', 'prison', 'crime', etc. together. That such semantically-related words appear in the same sets of topic words indicates that the clustering algorithm did indeed successfully group together movies that treat similar topics.

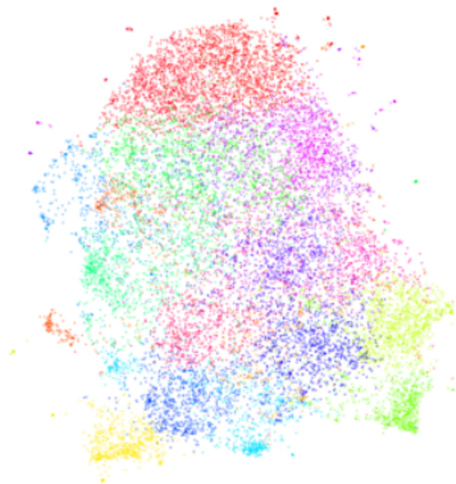

Figure 1: K-means model result of 20 topic clusters

However, there were also a few clusters were there seemed to be no explicit relation between the topic words of that cluster, e.g. in one cluster, the words 'french', 'father', 'king', and 'war', were grouped together. It is therefore unclear from looking at such topic words whether the movies grouped together in these clusters are actually similar in terms of their topic matter. It may be the case that there does actually exist a strong similarity between the movies in such clusters and that the topic words are just a poor representation of the underlying topic of these movie synopses.

Another possible explanation for the seemingly unrelated topic words in some clusters is that the k-means clustering performed poorly here, which is likely, given that this method forces every movie instance into some cluster rather than having outliers. This is unlike the HDBscan clustering model that we initially used, which deemed about 9000 out of our 14000 instances of movies to be outliers rather than assigning them to a cluster. The k-means clustering method may have therefore clustered together movies that have no similarity between them.

7 Limitations and Future Research

One of the main limitations of this project is the lack of sufficient computing power. Running the advanced BERT models on the available GPUs required long waiting times and restricted us to using the plot synopsis of movies instead of the full scripts. However, there are multiple online resources from which full scripts of movies could be obtained, and these are likely to be much more informative, in terms of emotion analysis and topic modeling, than the plot synopsis texts that we used in this project. Therefore, a future improvement of the application could be to apply our approach using full movie scripts and more computing power. Since such scripts are a better representation of the content of a movie, this would likely yield more accurate emotion scores and topic clusters.

Furthermore, with the current state of the application, it is rather difficult to quantitatively evaluate the performance of emotion analysis since our approach relies on unsupervised techniques. One possible improvement is to manually annotate movie synopses with emotion scores (gold labels). In this extension, the performance of the emotion analysis model could be evaluated by comparing the similarity of the model's scores to those of the annotators.

As for topic modeling, it is difficult to verify whether or not the topic words generated by topic analysis for each movie cluster are accurate. Apart from user feedback, a possible way to validate correctness is to manually, similar to the emotions, annotate each plot synopsis then check whether or not the topic is correct. One way to improve this process would be to first create gold labels (human annotated labels) for a subset of the data, then use those labels to create machine annotated labels, thus creating silver labels. This could speed up the process since each individual instance does not have to be manually annotated.

8 Conclusion

In this project, we have utilized NLP techniques in order to build a movie recommendation system. We first pre-processed the dataset then applied emotion analysis and topic modeling, as well named entity recognition. For both techniques, we leveraged the pre-trained models from the hugging face platform. Moreover, we built an application with a user interface in order to allow a user to discover movies based on a set of emotions that they would like to experience from watching a movie. The user can also rely on the app to find movies that are similar to a specific movie of their selection. Finally, we discussed some limitations and future improvements with regard to the NLP techniques used, computation power and evaluation metrics.

9 Division of the work

Chileshe

1. *Coding*: performed Named Enitity Recognition as well as carrying out the necessary data pre-processing for NER and topic modeling; did filtering of person named entities from synopses; trialed different NER models (spaCy, NLTK, BERT) to find the best one

2. *Analysis*: analyzed the cluster sizes and topic words generated by different clustering algorithms with different parameter settings to find the best algorithm and parameters for the clustering task

3. *Reporting*: wrote the Results and Topic Modeling/NER sections; contributed to the Application section; did proofreading and editing/corrections

Denise

1. *Coding*: set up and tested the LDA algorithm on our data set, performed Topic Modeling with BERTopic, and set up different clustering methods(HDBscan, DBscan, K-means, Hierarchical Clustering) with multiple parameters

2. *Analysis*: analyzed the cluster sizes and topic words generated by different clustering algorithms with different parameter settings to find the best algorithm and parameters for the clustering task

3. *Reporting*: wrote the Motivation and Data Set sections; contributed to the Topic Modeling/NER, Application and Limitation & Improvements sections

Hasan

1. *Coding*: emotion anaylsis and related data processing; set up the user application using ReactJS in the front-end and the Python Flask library for the backend
2. *Analysis*: analyzed the performance of the emotion transformer DistilBERT and accounted for edge case scenarios (e.g. maximum length of accepted string)
3. *Reporting*: wrote the Conclusion, contributed to the Emotion Analysis section and the emotion analysis part of the Results section; did proofreading

Mohamad

1. *Coding*: performed data cleaning and prepossessing; applied the emotion analysis using DistilBERT; processed the obtained results using Pandas; set up the backend functionality of the app
2. *Analysis*: inspected the results of emotion analysis and cleared up odd outputs
3. *Reporting*: wrote the Introduction, Emotion Analysis and Limitations sections; contributed to the Motivation section and the emotion analysis part of the Results section; did proofreading

References

[Aamir and Bhusry2015] Mohammad Aamir and Mamta Bhusry. 2015. Recommendation system: state of the art approach. *International Journal of Computer Applications*, 120(12).

[Burke2002] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370.

[Grootendorst2020] Maarten Grootendorst. 2020. Topic modeling with bert.

[Sang and De Meulder2003] Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

[Sanh et al.2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[Wakil et al.2015] Karzan Wakil, Rebwar Bakhtyar, Karwan Ali, and Kozhin Alaadin. 2015. Improving web movie recommender system based on emotions. *International Journal of Advanced Computer Science and Applications*, 6(2):218–226.

[Zhang and Zhang2022] Yilin Zhang and Lingling Zhang. 2022. Movie recommendation algorithm based on sentiment analysis and lda. *Procedia Computer Science*, 199:871–878.

Appendix

A

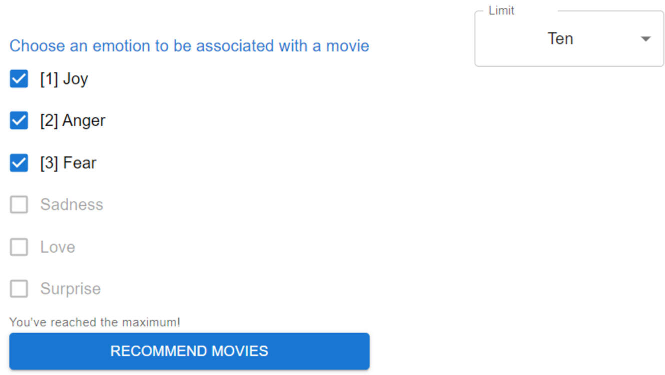

Figure 2: Emotion selection

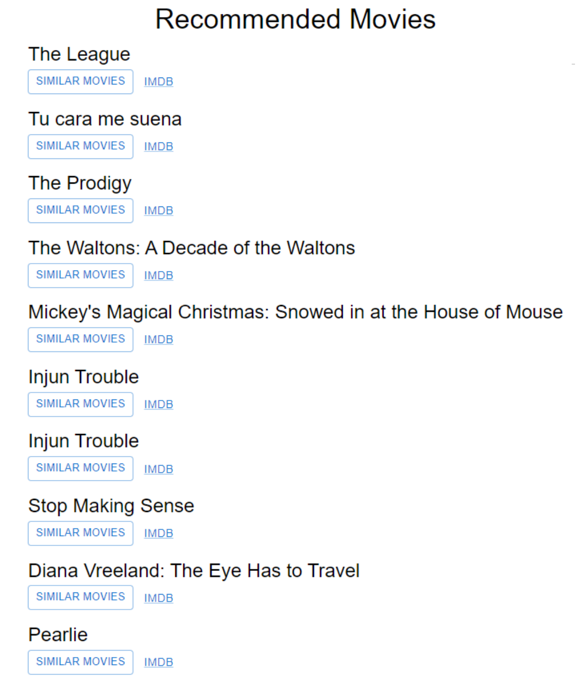

Figure 3: Recommended movies based on the selected emotions

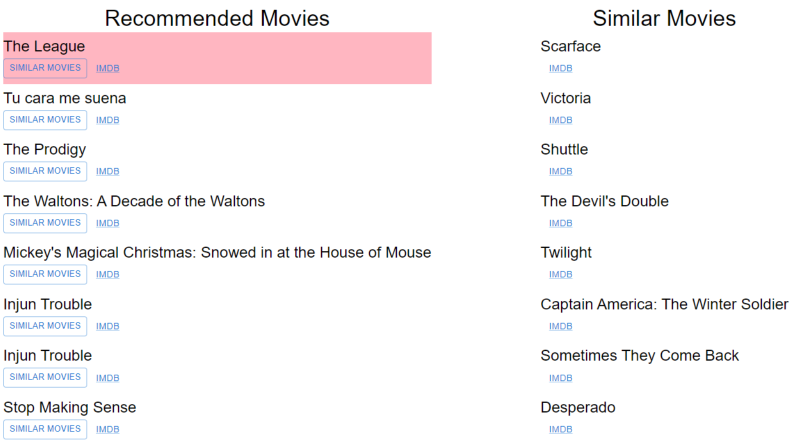

Figure 4: Similar movies based on the selected movie