PlotMatch

Uniqueness in an age of unoriginality

William Teng, Adeline Chin
william.teng@berkeley.edu, avch@berkeley.edu

W266 - Natural Language Processing with Deep Learning
UC Berkeley MIDS

Abstract

With the emergence of mainstream media and streaming services, more movies and plot ideas have been released to the public than ever. Nowadays, many 'new' movies are continuations, adaptions from books, or reworks of classic films. Our project develops a novel approach to evaluate how similar a submitted movie plot is to existing films. By analyzing extractive summaries of raw plot synopses, we employ a two-tower retrieval architecture alongside a K-Nearest Neighbors predictive model. This method generates ranked lists of movies whose plots align closely with the input. Traditionally, understanding nuanced linguistic relationships has posed a significant challenge in language processing, however, with advancements in computational tools and methods, we are now better equipped to perform analyses that closely mirror human judgment and comprehension. This paper details PlotMatch's methodology and performance in offering plot similarity assessments.

1 Introduction

Narrative innovation is pivotal in preserving the originality and integrity of storytelling within cinema and home entertainment, an industry that supplies a highly consumed form of storytelling and is also worth over 100 billion USD in 2019[1]. As this industry continues to expand, authors and screenwriters face the increasing challenge of distinguishing their work in a market saturated with derivatives and clichés. Despite the industry's reliance on creativity, it frequently faces criticism for perceived laziness and unoriginality, often being accused of pursuing formulaic 'cash-cow' strategies that prioritize financial gain over artistic inspiration[2]. The fundamental challenge lies not in avoiding the reuse of themes, which is inevitable given the vast history of storytelling, but in innovatively building fresh and engaging content.

Although the core design of PlotMatch is to identify uniqueness by identifying similar films, one of the concepts underlying PlotMatch is derived from the pervasive inadequacies observed in some existing film recommendation systems, which often yield imprecise suggestions. For instance, platforms like Netflix might recommend an entire genre of horror films in response to a request for movies similar to "The Omen (1976)," despite the user's interest in specific storylines or character similarities. Recognizing this shortfall, the development of PlotMatch involved a sophisticated approach where movie tags and specific entities—such as themes, plot elements, and character types—are categorized. grouped and analyzed. This method allows for a more nuanced evaluation of similarity, substantially refining the performance of recommendations.

2 Related Work

The majority of related work regarding movie profiling has predominantly centered on summarization, with a particular emphasis on extractive rather than abstractive techniques. This preference aligns with the notion that movie summarizations are heavily dependent on specific keywords that denote types of characters, distinct time periods, and historical events. Given the relative ease of generating accurate summaries from preexisting plots, recent scholarly efforts have pivoted towards enhancing the accuracy of these summaries using graph models from scripts (Papalampidi et al., 2021) and even full-length movies (Gorinski and Lapata, 2015). Over time, summarization techniques such as T5 (Xue et al., 2021), entity recognition (Sharma et al., 2021), and even zero-shot classification (Gubelmann et al., 2022) have been developed.

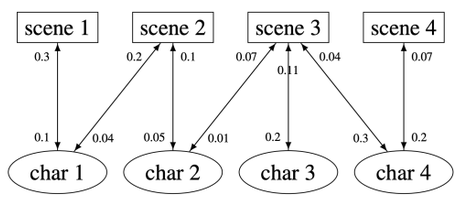

Figure A. Example of bipartite graph connecting scenes to characters (Gorinski and Lapata, 2015).

In order to measure the accuracy of extractive summaries, common metrics are the use of distances and the correctness of model outputs to answer specific questions related to movie plots (Gorinski and Lapata, 2015). In particular, Gorinksi and Lapata have developed categories to define distances with graph models, including the notion of Total Agreement, Partial Agreement, and Mean Distance.

[1] https://www.boxofficepro.com/mpa-2019-global-box-office-and-home-entertainment-surpasses-100-billion/
[2] https://filmoriginality.weebly.com/unoriginality-is-ruining-film.html

Even with the development of specific summarization techniques, there is little current work on the subject of accurately comparing and defining similarities between generated summaries from different sources, which is what we hope to accomplish with this project. An adjacent project known as LOGTELL (Furtado et al., 2008) has been created that manages plots for generative storytelling, highlighting the importance of consistency of entities and relationships, where consistency could translate towards similarity.

3 Methods

3.1 Task

The objective of this research is to devise a system that enables storytellers to assess their narratives against a corpus of established literary and cinematic works. The proposed methodology employs a query architecture that summarization, categorization, Named Entity Recognition (NER), and similarity matching techniques. This system identifies potentially analogous works within a dataset of films from IMDb. Considering the subjective nature of narrative originality, the model is designed to generate predictions of stories with similarities rather than flag exactly if the submission was unique. Subsequently, human evaluators can apply their judgment to assess the subjective uniqueness of these stories. Specifically, PlotMatch utilizes a subset of movie plots to generate and rank a list of the top 10 similar movies. These results are then analyzed using relevancy metrics to determine the accuracy and appropriateness of the similarity rankings produced.

3.2 Database Tower

PlotMatch fundamentally uses the two-tower architecture for deep retrieval. The two-tower encoder model is an embedding-based search where one tower produces the query embedding and the second computes the candidate embedding. After creating the database of items, the first step is to train both neural network towers with the labeled query and database item pair. Then, the database items are mapped to the embedding space. This enables the query embedding to be computed against the database tower and vector similarity searches are performed[3]. Figure B below visualizes the final encoder model.

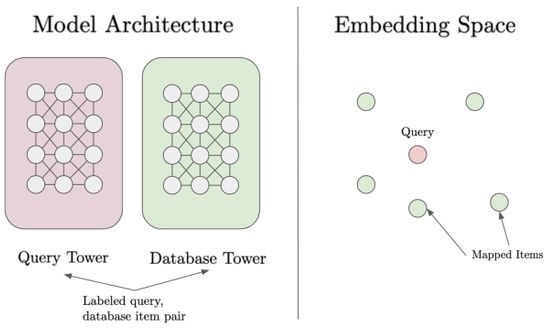

Figure B. Two-tower encoder model diagram

The movie dataset used for the database tower is called MPST (Movie Plot Synopses with Tags) which is a compilation of movie plot extracts gathered from IMDb and Wikipedia by ACL Anthology containing over 14,000 records. When exploring our hybrid model, both the database tower and the query tower are augmented with methods such as identifying entities, settings, and objects mentioned in the document as well as predicting respective film genres.

3.3 Zero-Shot Augmentation

From the raw plot synopses, we decided to use a genre classification method by leveraging the bart-large-mnli (Bidirectional and Auto-Regressive Transformers Large Natural Language Inference) model that uses pre-traind NLI models as zero-shot sequence classifiers[4]. This classification method is an encoder-decoder model, which encodes the incoming document to have contextualized tokens and the decoder component outputs a sequence based on the encoder's outputs. Once loaded, the zero-shot classification pipeline was used to classify sequences into any of the class names specified. In the case of this project, the classes were a list of genres. When first applying this model and attempting to generate all genres, we encountered the issue of abundant specificity, genres such as "thriller-horror" and "zombie-horror", which would be detrimental to the classification. We ultimately decided to manually create a shortlist of predefined genres to prevent overfitting. The list includes a total of 50 genres.

Prior to structuring querying, it was necessary to understand what requirements would be useful for the plot reference dataset to pre-process or what is required of the input. It was decided that a single sentence for the user to input would often be too vague and hence result would vary wildly due to the general nature of the plot. Therefore it was decided that the user is recommended to submit at least 3 sentences for meaningful retrieval.

The notion of dynamically generating genre tags and conducting entity extraction across the dataset tower dynamically holds theoretical promise and would allow for flexible swapping in data, enabling adaptability to diverse datasets. However, the computational demands associated with each iteration were considerable, with processing times exceeding 17 minutes to rebuild. Notably, a significant portion of this computational burden was attributed to the genre tagging process utilizing the zero-shot capability of the Facebook/bart-large-mnli model. Consequently, a decision was made to pre-tag the dataset to streamline computational efficiency. Furthermore, T-5 summarization techniques were employed to distill key plot points from narratives, facilitating the computation of cosine similarity metrics relative to the user's plot pitch.

3.4 Entity Extraction

Using the raw plot synopses provided in the original dataset, we also extracted lists of entities for each plot using spaCy's English pipeline

[3] https://cloud.google.com/blog/products/ai-machine-learning/scaling-deep-retrieval-tensorflow-two-towers-architecture
[4] https://huggingface.co/facebook/bart-large-mnli

en_core_web_md[5] and created four separate categories of entities: Characters (char), Locations (loc), Temporal Settings (temp), and Items. Table 1 outlines the specific labels for each category.

char	loc	temp	items
PERSON	GPE	DATE	FAC
ORG	LOC	TIME	PRODUCT
		EVENT	

PERSON: People, including fictional.
ORG: Companies, agencies, institutions, etc.
GPE: Countries, cities, states.
LOC: Non-GPE locations, mountain ranges, bodies of water.
DATE: Absolute or relative dates or periods.
TIME: Times smaller than a day.
EVENT: Named hurricanes, battles, wars, sports events, etc.
FAC: Buildings, airports, highways, bridges, etc.
PRODUCT: Objects, vehicles, foods, etc. (Not services.)

Table 1. spaCy labels of extracted entities and descriptions

For each plot synopsis, the result for each category was a nested list of entity extractions. Each result was then flattened to a single list and padded with zeroes using a custom function so that all extractions were the same length.

3.5 Vectorization

Given that the raw plot synopses are in the form of a natural language paragraph represented as a string, we used the sentence-transformer model all-roberta-large-v1 to encode the raw data. Converting the raw text into structured numerical vectors is crucial for computational analysis because it enhances semantic understanding, ensures compatibility with our retrieval algorithm.

This model was pre-trained from roberta-large and maps text in the form of sentences or paragraphs to a dimensional dense vector space of size 1024[6]. This encoder model was also used on the generated lists of extracted entities and genres, which had been previously flattened and normalized to yield lists of size 1024.

When first experimenting with the vectorization for PlotMatch, we attempted to generate summaries from the raw plot synopses using the T5 encoder-decoder model in hopes that it would act as a first pass to extract the most relevant ideas from the plot and potentially decrease the processing time of the rest of the steps. However, even with extensive fine-tuning and optimization, we found that the generated summaries excluded entities that the entity extraction pipeline picked up with the raw synopses. Furthermore, the subsequent steps' processing times did not significantly decrease with the T5 model, so we decided to work with the raw plot synopses for the baseline model.

4 Base Model

The foundational architecture of our model employs a K-Nearest Neighbors (KNN) approach, specifically utilizing a two-tower embedding framework that leverages cosine similarity to rank resultant outputs. Cosine similarity measures the cosine of the angle between vectors. This measurement quantifies the orientation similarity between vectors, making it particularly suitable for text analysis where the magnitude of the vectors may not correspond directly to their semantic similarity.

In the context of plot similarity evaluations performed by PlotMatch, cosine similarity offers a method for assessing the closeness of textual documents in a multi-dimensional vector space.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \, ||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

This algorithm requires both the query and the database tower to be encoded. The encoding does take a significant amount of time given the size of the database tower, so the base model begins by preprocessing the MPST dataset's plot. Using Netflix as a user experience reference point that often uses the top 10 as the K value for their listings, our K value for KNN will also be 10. Using these parameters, we established our baseline model using embedding (Kenter and de Rijke, 2015).

One limitation we found during the initial development of the base model is the use of extract token matches. This is a rudimentary start, however, a major issue is that it fails when a token of the same spelling can have multiple meanings.

An example of this occurred when testing a prompt for the film "Fight Club (1999)" which flagged 'soap' as a noun, where the character Tyler Durden is referred to as a 'soap salesman', among the returned matching films was "Intolerable Cruelty (2003)" because of the character, Donovan Donaly, who is referred to as "a TV soap opera producer". This highlights a few key disadvantages of static tokens which will also be an issue if we proceed with static word embedding. It also limits other words that may relate but fall outside of the corpus of tokens captured in the fields.

Sampling with 120 plots of user-generated movie plots based on the actual movie, and using the IMDb ID as the label, we used the embeddings of the raw plot synopses as inputs for the KNN model in hopes that the same movie would appear first in the ranking or, at the very least, within the top ten. In the early stages, this was our human-interpretable evaluation metric. Below is an example of the output of the baseline model for the input "Schindler's List (1993)" where the first line is the movie that is predicted to be the most similar.

[5] https://spacy.io/models/en
[6] https://huggingface.co/sentence-transformers/all-roberta-large-v1

imdb_id	title	cos_sim
tt0108052	Schindler's List	0.2714
tt0038769	Die Mörder sind unter uns	0.2578
tt0363163	Der Untergang	0.2412
tt0346293	Hitler: The Rise of Evil	0.2245
tt0092978	Escape from Sobibor	0.2064
tt0161860	Nirgendwo in Afrika	0.1429
tt0071688	Jakob, der Lügner	0.1185
tt0108211	Stalingrad	0.1095
tt0099776	Europa Europa	0.1022
tt1280548	Spielzeugland	0.0842

Table 2. Example of baseline output for "Schindler's List (1993)"

5 Hybrid Model

The hybrid model extends the foundational architecture of the base model by incorporating a K-Nearest Neighbors (KNN) model to process the plot synopses, extracted entities, and genres associated with each film. This adaptation generates five additional sets of cosine similarity values and corresponding lists of movies ranked according to these values based on: genre classification, characters, locations, temporal settings, and items.

Upon evaluating performance metrics for each category independently, it was observed that the embeddings derived from plot synopses outperformed those based on genre or character traits. This finding reinforces the designation of this configuration as our base model. The relatively lower performance of the entity embeddings can be attributed to the variability in the number of entities extracted across different categories. For instance, a significant number of films yielded no extractions for the 'items' category, resulting in cosine similarity scores of zero, except when compared to other films with an identical absence of items, where the score would then default to one.

The method used to combine the rankings from all categories was to add on top of the originally generated list of 10 similar movies from the baseline model. For each of the 10 movies, their IMDb ID was matched with the keys in the dictionaries of cosine similarity values for each category. If the predicted similar movie appears in the ranking for a category, the cosine similarity value is added to the value for the baseline ranking, and the ranking is updated. Table 3 outlines an example of how an intermediate ranking is generated based on the values from the character category.

Baseline	Char	Updated
Pred 1: 0.56	+0.32	Pred 1: 0.86
Pred 2: 0.50	+0.00	Pred 3: 0.70
Pred 3: 0.49	+0.21	Pred 9: 0.62
Pred 4: 0.47	+0.10	Pred 4: 0.57
Pred 5: 0.47	+0.00	Pred 2: 0.50
Pred 6: 0.42	+0.00	Pred 5: 0.47
Pred 7: 0.31	+0.15	Pred 7: 0.46
Pred 8: 0.28	+0.16	Pred 8: 0.44
Pred 9: 0.26	+0.36	Pred 6: 0.42
Pred 10: 0.25	+0.00	Pred 10: 0.25

Table 3. Example of combination of cosine similarity values

6 Results and Discussion

6.1 Metrics

To determine the effectiveness of the KNN model and evaluate the hybrid against the baseline model, we chose four metrics that each encompassed a different attribute of the semantic processing. An arbitrary scoring mechanism weights how well the pitched submission matches with the dataset's genres, then if they are overlapping entities using extract string matching and finally ranked by cosine similarity. All models return a sorted list based on their respective relevance mechanisms.

The first metric is Mean Average Precision (mAP). Usually used for object detection tasks, mAP is also used to evaluate the performance of information retrieval, which is the first step in the overall process of extracting relevant data from plot synopses. mAP is calculated by finding the Average Precision (AP) for each class, then taking the average over all classes. The following equations outlines how this metric is calculated and has a range of 0 to 1[7].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

The second metric is Normalized Discounted Cumulative Gain (NDCG). While also used in the evaluation of information retrieval, NDCG is useful when determining the correctness of rankings. NDCG compares output rankings against an ideal where the top results are the most relevant. The range for NDCG is from 0 to 1 where 1 indicates a ranking with the ideal order and lower values representing lower quality rankings[8]. Below is the derivation of NDCG, with DCG as the Discounted Cumulative Gain and Gain being the relevance score.

$$\text{DCG} = \sum_{i=1}^{\text{ranks}} \frac{\text{Gains}}{log_2(i+1)}$$

$$\text{NDCG} = \frac{\text{DCG}}{\text{DCG}_{\text{ideal}}}$$

The third metric is Kendall's tau. This value measures the relationship between two instances of ranked data. Kendall's tau is considered a correlation metric, where -1 corresponds to perfect negative correlation and 1 corresponds to perfect positive correlation. In the equation below, C represents the number of concordant pairs and D represents the number of discordant pairs[9].

$$\tau = \frac{\text{C-D}}{\text{C+D}}$$

The last metric chosen to evaluate our models is Spearman's rho. Similarly to Kendall's tau, this is a correlation metric that evaluates the similarity

[7] https://kili-technology.com/data-labeling/machine-learning/mean-average-precision-map-a-complete-guide
[8] https://www.evidentlyai.com/ranking-metrics/ndcg-metric
[9] https://www.statology.org/kendalls-tau/

of two rankings, with the range of possible values from -1 to 1. The following equation is the calculation for Spearman's rho where d_i is the difference between the two ranks of each observation and n is the total number of observations[10].

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2 - 1)}$$

6.2 Results

As expected, the hybrid model containing the updated ranking of movies using the combined cosine similarity values of the plot synopsis, genre, and entity embeddings performed better for all metrics than the baseline model. Table 4 below shows the specific values for each model and metric.

	mAP	NDCG	τ	ρ
Baseline	0.2513	0.3333	0.0794	0.1150
Hybrid	0.7853	0.9950	0.1879	0.2671

Table 4. Final metric values

Surprisingly, the mAP and NDCG values for the baseline model were fairly low and the tau and rho values were near 0. Because the baseline model encoded the raw plot synopses, we expected the metrics to be higher, especially the NDCG value. However, it makes sense that the tau and rho values were near 0 as the baseline model had no additional features to improve the specific ranking of predicted movies.

In the evaluation of the hybrid model, the Normalized Discounted Cumulative Gain (NDCG) value approached 1, signifying that the model effectively ranked the most relevant titles at higher rankings. Despite this, the incremental improvements in Kendall's tau and Spearman's rho metrics from the baseline to the hybrid model were minimal, with values remaining near zero. This suggests that although the model positions individual movies appropriately within the rankings, the overall correlation with expected rankings of similar movies is weak. The initial expectation was that similar movies would output similar rankings, however, based on these results, that was not the case.

Remembering that both the baseline and the hybrid models were tested with a subset of the movies from the database to test whether that same movie would appear in the ranked output, this means that the hybrid model was indeed able to output the same movie that had been inputted, however, the rest of the ranking may have not been as similar as we had hoped.

What this suggests is that the retrieval performance is significantly improved when significant entities, settings, and items are emphasized. Thus for PlotMatch, the differentiation is to weigh these extracted features more when applying retrieval.

Looking at the mAP values, the final precision analyzed in conjunction with the NDCG value for the hybrid model shows that the hybrid model was able to accurately retrieve similar movies from the database. This can be attributed to the use of the two-tower model architecture and creates the understanding that the retrieval methods of PlotMatch are acceptable. Given the time, if we were to improve these metrics, we would focus on fine-tuning the two-tower model which would likely impact the mAP and NDCG values the most.

7 Conclusion

The original use case of PlotMatch would be used for authors and film industry executives to evaluate screenplays and scripts to test if film submissions are likely to be too similar to existing work, to which PlotMatch has achieved some confidence by returning a ranked list based on similarity metrics which users can ultimately make a judgment call on.

Looking forward, the PlotMatch project holds potential for a range of applications beyond assessing the uniqueness of movie plots. The core architecture of PlotMatch, designed to identify and extract semantic similarities could be adapted for diverse fields. For instance, legal professionals might utilize this system to efficiently retrieve case law that shares pertinent legal precedents or thematic elements. Similarly, academic researchers could employ PlotMatch to identify scholarly articles that utilize similar methodologies or address similar research questions, thus facilitating a more efficient literature review process.

This capability to discern semantic similarities can significantly reduce the time and effort required for data-intensive tasks, enabling professionals across various disciplines to focus on higher-level analysis and decision-making.

For even more capability to expand on the entities retrieved, it would benefit from Wikification, a technique that enriches text with a knowledge base that can draw out further relationships and relative meaning. This would be particularly useful when compiling the database tower when the provided plots have a minimal description of certain features such as characters. For example, the film X-Men (2000) has several characters with abilities, however, the plot entry does not comprehensively list out all characters with their associated powers. By applying Wikification, these additional references can enrich the database tower and further improve upon PlotMatch's performance metrics.

With a little bit of re-working to include industry-specific requirements, PlotMatch could be used for a variety of amateurs and professionals alike, allowing us to uphold the uniqueness of our ideas.

[10] https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide-2.php

References

Furtado, A. L., Casanova, M. A., Barbosa, S. D., Breitman, K. K. (2008). Analysis and Reuse of Plots Using Similarity and Analogy. *Lecture Notes in Computer Science*, 355–368.

Gomaa, W. H., Fahmy, A. A. (2013). A Survey of Text Similarity Approaches. *International Journal of Computer Applications (0975 – 8887).*

Gorinski, P. J., Lapata, M. (2015). Movie Script Summarization as Graph-based Scene Extraction. *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, 1066–1076.

Gubelmann, R., Handschuh, S. (2022, January 19). Uncovering More Shallow Heuristics: Probing the Natural Language Inference Capacities of Transformer-Based Pre-Trained Language Models Using Syllogistic Patterns. ArXiv.org.

Kenter, T., de Rijke, M. (2015). Short Text Similarity with Word Embeddings. *Proceedings of the 24th ACM International Conference on Information and Knowledge Management.*

McKeown, K., Radev, D. R. (1995). Generating Summaries of Multiple News Articles. *Association for Computing Machinery, Inc. (ACM).*

Papalampidi, P., Keller, F., Lapata, M. (2021) Movie Summarization via Sparse Graph Construction. *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21).*

Sethi, P., Sonawane, S., Khanwalker, S., Keskar, R. B. (2017). Automatic text summarization of news articles. *2017 International Conference on Big Data, IoT and Data Science (BID).*

Sharma, A., Amrita, Chakraborty, S., Kumar, S. (2021). Named Entity Recognition in Natural Language Processing: A Systematic Review. *Proceedings of Second Doctoral Symposium on Computational Intelligence*, 817–828.

Wang, J., Dong, Y. (2020). Measurement of Text Similarity: A Survey. *Information 2020*, 11, 421.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C. (2021). *mT5: A massively multilingual pre-trained text-to-text transformer.* ArXiv:2010.11934 [Cs].