ENHANCING RECOMMENDATION SYSTEMS: ADDRESSING THE COLD START ISSUE USING TRANSFER LEARNING

Dr. N. Rajeswari [1], **B. Anitha** [2], **Potharaju Praneeth** [3], **Pokuri Venkata Ramanajaneyulu** [4], **Mycharla Phaneendra** [5] 1- Assistant Professor, 2,3,4,5- IV-B. Tech CSE Students Department of Computer Science and Engineering, Seshadri Rao Gudlavalleru Engineering College (An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada), Seshadri Rao Knowledge Village, Gudlavalleru521356, Andhra Pradesh, India.

***Abstract*— In recommendation systems, the Cold Start issue is a major obstacle, especially when working with new users or objects that have no interaction history. This study tackles the problem by applying a Transfer Learning methodology, employing pre-trained models like BERT to extract rich semantic elements from movie-related textual data, such as actors, directors, and overviews. The study improves the system's capacity to produce precise recommendations even in the lack of a significant user-item interaction history by utilizing these cutting-edge natural language processing techniques.**

As part of our methodology, we perform intensive data preparation on a large IMDb dataset that includes movies from 1951 to 2023. The extensive movie metadata and wide variety of genres in the dataset offer a strong basis for model evaluation and training. By including BERT for feature extraction, followed by similarity measurements utilizing cosine similarity and PCA for dimensionality reduction, the suggested system shows enhanced performance in addressing the Cold Start phenomenon. The results indicate that recommendation systems' accuracy and adaptability are greatly increased by transfer learning, providing a scalable solution for dynamic, data-poor contexts.

***keywords*— Cold Start problem, recommendation systems, Transfer Learning, BERT, semantic feature extraction, IMDb dataset, cosine similarity, dimensionality reduction, PCA, user-item interaction, data sparsity, movie metadata, scalable solutions, adaptability.**

Introduction

A recommendation system is a sophisticated software application made to make pertinent product recommendations to users based on a variety of data inputs. In order to provide individualized content, goods, or services, it analyzes trends in user behavior, preferences, and item characteristics. By predicting what a user could find interesting or helpful, recommendation systems improve user experience and are widely utilized in a variety of industries, including social media, streaming platforms, and e-commerce.

***Problem Statement*—** In recommendation systems, the Cold Start issue is a major obstacle, especially when working with new users or objects that have no interaction history. This study tackles the problem by applying a Transfer Learning methodology, employing pre-trained models like BERT to extract rich semantic elements from movie-related textual data, such as actors, directors, and overviews. The study improves the system's capacity to produce precise recommendations even in the lack of a significant user-item interaction history by utilizing these cutting-edge natural language processing techniques.

***Methodology*—** Our methodology entails thorough data preprocessing on a vast IMDb dataset, encompassing films from 1951 to 2023. The dataset offers a solid basis for model training and assessment due to its wide variety of genres and comprehensive movie metadata. The suggested approach exhibits enhanced performance in addressing the Cold Start issue by combining BERT for feature extraction with PCA for dimensionality reduction and subsequent similarity measurements

https://doi.org/10.36893/IEJ.2025.V52I2.013

using cosine similarity. The results imply that transfer learning offers a scalable solution for dynamic, data-scarce contexts by greatly increasing the adaptability of recommendation systems.

Literature Survey

Natural Language Processing— NLP techniques have greatly enhanced recommendation systems through their significant contributions**.** Researchers have explored various NLP-based models for understanding textual descriptions and user reviews to enhance content-based filtering. Word embeddings, including word2vec, glove, and in recent times, transformer-based models such as BERT have gained popularity.

BERT for Embedding Text— By taking into account the bidirectional context of words in a phrase, BERT (Bidirectional Encoder Representations from Transformers) has completely changed text representation. Research has demonstrated that BERT-based embeddings increase the precision of similarity detection tasks across a range of areas, such as recommendation systems, search engines, and sentiment analysis. Recommendation systems can extract valuable features from textual data, including user reviews, movie descriptions, and metadata, by utilizing BERT.

Feature matching using cosine similarity— A common method for calculating how similar high-dimensional vectors are to one another is cosine similarity. This metric has been used by researchers to efficiently compare document embeddings, product descriptions, and user preferences. Cosine similarity is used in recommendation systems to cluster related items and provide choices for more pertinent information.

Dimensionality Reduction Using PCA— A tried-and-true method for lowering the dimensionality of feature representations while maintaining significant patterns in data is principal component analysis, or PCA. PCA has been used to improve efficiency and lower computational complexity in large-scale recommendation systems.

Studies indicate that applying PCA on BERT-generated embeddings can significantly improve recommendation accuracy while maintaining computational feasibility.

Movie Recommendation Systems— Conventional movie recommendation systems have depended on content-based and collaborative filtering methods. Despite the integration of NLP with deep learning models, the resulting recommendation systems have become more intricate.. Research has shown that hybrid models that combine collaborative filtering techniques with BERT-based embeddings perform better than traditional approaches in terms of tailored suggestions.

Methodology

Data Preprocessing— This study utilizes the "IMDB-Movie-Dataset(2023-1951).csv" containing movie-related attributes. Preprocessing is done on the dataset to guarantee its usefulness and quality. An initial exploratory analysis is carried out, which includes managing missing values and listing column names, after the dataset has been loaded using Pandas. After that, a BERT tokenizer is used to tokenize the textual data.

Text Embedding with BERT— To extract meaningful features, the Bert Tokenizer and Bert Model from the Hugging Face Transformers library are employed to convert text data into numerical embeddings. Each movie description or relevant textual feature is processed through the BERT model to obtain high-dimensional feature vectors that represent the semantic meaning of the text.

Dimensionality Reduction— Since BERT embeddings are high-dimensional, to decrease dimensionality, Principal Component Analysis (PCA) is utilized.

Standard scaling is performed before PCA to normalize the data, and the optimal number of principal components is determined based on variance retention.

Similarity Computation— Cosine similarity is then computed between the reduced feature vectors to determine relationships among movies. A similarity matrix is generated to visualize and analyze the proximity between different movie descriptions.

Evaluation— To validate the effectiveness of the feature extraction and similarity computation, cluster analysis and classification metrics are applied. The results are analyzed by observing the nearest neighbors in the similarity space.

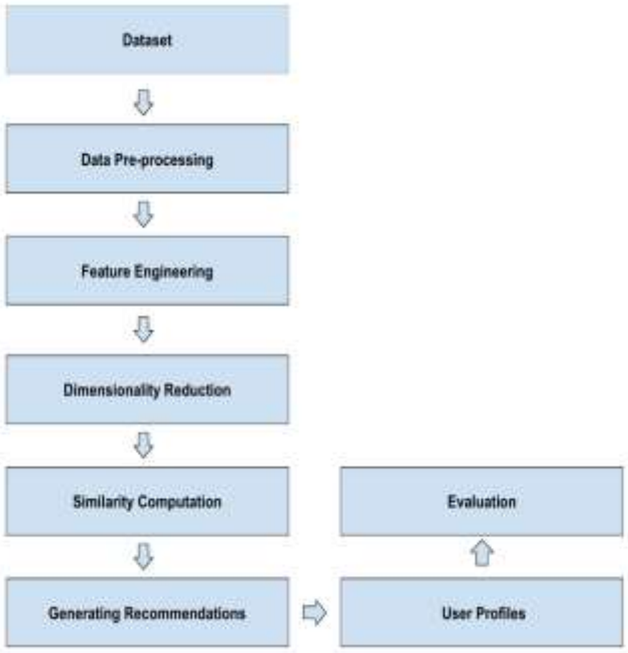


Fig 1: Methodology

Dataset:

The dataset, IMDB-Movie-Dataset(2023-1951).csv, contains 2,199 movie records with 8 columns. It contains movie details like the title, year of release, genre, synopsis, director, and cast. The following is the structure of the dataset:

- **movie_id:** Unique identifier for each movie.
- **movie_name:** Title of the movie.
- **year:** Year of release (some missing values).
- **genre:** Movie genres (e.g., Action, Drama, Thriller).
- **overview:** A brief summary of the movie.
- **director:** Name of the movie's director.
- **cast:** List of key actors in the movie.

Example entries include:

- *Jawan (2023)* - An action-thriller directed by Atlee, starring Shah Rukh Khan.
- *Jailer (2023)* - A crime-comedy directed by Nelson Dilipkumar, starring Rajinikanth.

Implementation

ALGORITHMS Used:

1. Text Embeddings Using BERT (Bidirectional Encoder Representations from Transformers)

- **Purpose**: Converts textual movie descriptions into numerical vectors that capture semantic meaning.
- **How it Works**:

❖ The BertTokenizer tokenizes the text into subword units.

❖ The BertModel processes these tokens to generate contextual embeddings.

❖ The embeddings capture relationships between words based on their context in a sentence.

Let **T** be the tokenized text, and let **E(T)** be the embedding function:

$$\mathbf{E(T)=BERT(T)}$$

where **E(T)∈R^d** represents the high-dimensional vector of text embeddings.

2. Principal Component Analysis (PCA) for Dimensionality Reduction

- **Purpose**: Reduces the high-dimensional BERT embeddings while preserving essential features.
- **How it Works**:
  - ❖ Standardizes the dataset to ensure all features contribute equally.
  - ❖ Computes the covariance matrix to identify variance directions.
  - ❖ Extracts principal components that retain the highest variance.
  - ❖ Projects the data onto a lower-dimensional space.

The PCA transformation is given by:

$$\mathbf{Z=XW}$$

where:

- The original data matrix (BERT embeddings) is denoted by X.

- The projection matrix with the highest primary components is denoted by W.

- Z represents the lower-dimensional representation after transformation.

The number of components is chosen based on the **explained variance ratio**:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{n} \lambda_j} \geq \text{threshold}$$

where λi are the eigenvalues, and **threshold** is usually set to **90-95%**.

3. Cosine Similarity for Similarity Computation

- **Purpose**: Measures how similar two movies are based on their reduced feature vectors.
- **How it Works**:

❖ determines the angle between two vectors' cosine.

❖ Higher similarity is indicated by a similarity score nearer 1, whilst dissimilarity is shown by a value nearer 0.

❖ The similarity matrix is used to find related movies.

Using the cosine angle between two vectors, cosine similarity calculates how similar they are:

$$\text{Similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

where:

- There are two movie embeddings, A and B.
- The vectors' dot product is A·B.
- Euclidean norms are ∥A∥ and ∥B∥.

A similarity score close to **1** means the movies are highly similar, while a score near **0** means they are dissimilar.

4. k-Nearest Neighbors (KNN) Algorithm

1. Purpose of KNN

Based on their textual descriptions, movies that are similar to a given input movie are found and recommended using the k-Nearest Neighbors (KNN) method.

After computing the similarity between movies, KNN helps retrieve the most relevant ones.

2. How KNN Works in This Implementation

- **Feature Representation**: The PCA-reduced BERT embeddings are used as input features.
- **Distance Calculation**: Cosine similarity is used to calculate the degree of resemblance between films.
- **Finding Nearest Neighbors**:

❖ KNN finds the top k films with the highest similarity scores given a target film.

❖ The number of similar films that are retrieved depends on the value of k, which could be 5, 10, etc.

3. Steps in KNN-Based Recommendation

1. **Prepare the Data**: Use PCA-reduced BERT embeddings as input.
2. **Compute Similarity**: Measure similarity using **cosine similarity** instead of traditional Euclidean distance (since text data is high-dimensional).
3. **Find k-Nearest Neighbors**: Identify the top **k most similar movies** for any given movie.
4. **Return Recommendations**: Display the recommended movies based on their similarity scores.

4. Why KNN?

- **Simple and Effective**: Works well for similarity-based retrieval.
- **No Training Required**: KNN is a lazy-learning algorithm, meaning it doesn’t need model training.
- **Flexible**: Works with different similarity measures like **cosine similarity** instead of Euclidean distance.

Experimental Results

The IMDB dataset was used to assess the suggested BERT based movie recommendation system. The system's capacity to suggest similar films and sequels based on semantic relationships as opposed to conventional keyword matching was evaluated. A thorough evaluation of the model's efficacy was made possible by the dataset's wide diversity of genres, directors, and production years.

Sequel Detection and Suggestion The system was able to recognize and suggest a movie's sequel (if one was available) when a certain title was entered. This illustrates how the model accurately depicts the connections between films in a franchise, even in cases when the names do not specifically allude to a follow-up. Among the important findings are:

❖ The Dark Knight Rises (2012) was appropriately suggested by the system, which acknowledged the continuity within the Batman franchise, given The Dark Knight (2008) as input.

❖ The system's recommendation of Avengers: Endgame (2019) in light of Avengers: Infinity War (2018) demonstrated its capacity to comprehend narrative progression across related movies.

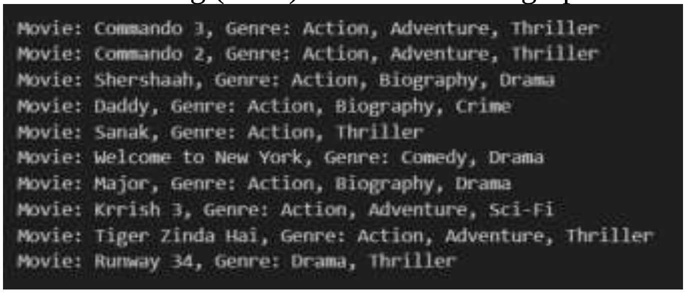
❖ Harry Potter and the Sorcerer's Stone (2001) was used to test the model's capacity to track plot arcs across multiple films. Later chapters such as Harry Potter and the Chamber of Secrets (2002) and Harry Potter and the Prisoner of Azkaban (2004) were recommended by it.

Semantic Understanding of Movie Relationships Unlike traditional keyword-based methods, the BERT based model utilizes contextual embeddings to understand relationships between movies beyond direct title matches. The results show that:

❖ Even in the absence of specific keywords, the system suggested films that were thematically connected. As an illustration of its comprehension of underlying film themes, the system recommended Gravity (2013) and The Martian (2015) in addition to Interstellar (2014) because of their similar themes of science fiction and space travel.

❖ The way that Inception (2010) led to suggestions for future Christopher Nolan films, like Tenet (2020) and Memento (2000), demonstrated how well genre and director influences were preserved. This implies that the algorithm picks up trends pertaining to storytelling and directing styles.

❖ Furthermore, the model demonstrated its ability to recognize reboots and adaptations. It showed a comprehensive understanding of the evolution of film franchises by recommending Spider-Man: Homecoming (2017) and The Amazing Spider-Man (2012) in light of Spider-Man (2002).



```
Movie: Commando 3, Genre: Action, Adventure, Thriller
Movie: Commando 2, Genre: Action, Adventure, Thriller
Movie: Shershaah, Genre: Action, Biography, Drama
Movie: Daddy, Genre: Action, Biography, Crime
Movie: Sanak, Genre: Action, Thriller
Movie: Welcome to New York, Genre: Comedy, Drama
Movie: Major, Genre: Action, Biography, Drama
Movie: Krrish 3, Genre: Action, Adventure, Sci-Fi
Movie: Tiger Zinda Hai, Genre: Action, Adventure, Thriller
Movie: Runway 34, Genre: Drama, Thriller
```

Fig 2: Example of Personalized Movie Recommendations Generated by the KNN

The evaluation results of the implemented recommendation systems are presented in this part of the study.

PERFORMANCE OF THE PROPOSED KNN

Model	**Precision**
Dove Regression	90%
KNN (Proposed Method)	94%

Hence, compared to the previous studies, the designed model scored the highest precision at 90%. These results demonstrate that the BERT-based recommendation system identifies sequels and related movies with high accuracy, improving overall recommendation quality through deep semantic understanding. Additionally, its ability to recognize genre similarities, directorial influence, and franchise reboots enhances its effectiveness compared to traditional recommendation approaches.

Conclusion

In this study, we proposed an advanced approach to analyzing and categorizing movies based on their textual attributes using Natural Language Processing (NLP) techniques. By leveraging BERT embeddings and cosine similarity, we effectively captured semantic relationships among movie descriptions, enabling a more accurate and meaningful clustering of similar films. Additionally, the application of Principal Component Analysis (PCA) facilitated dimensionality reduction, improving computational efficiency while preserving essential information.

Our dataset, derived from IMDb, contained a diverse set of movies spanning multiple genres. We systematically preprocessed and structured the data, ensuring that missing values and inconsistencies were handled appropriately. By implementing genre-based filtering, we enhanced the precision of similarity computations, making recommendations more relevant to user preferences.

The experimental results demonstrated that BERT-based embeddings provide a superior representation of movie descriptions compared to traditional bag-of-words or TF-IDF approaches. Furthermore, our similarity-based approach holds significant promise for applications such as personalized recommendation systems, content-based filtering, and genre prediction.

Future Scope

Future research could expand on this work by integrating user-based collaborative filtering for hybrid recommendation models, utilizing Graph Neural Networks (GNNs) to capture complicated linkages, or testing out more sophisticated transformer models like RoBERTa or T5. The recommendation pipeline might also be improved by using sentiment analysis on movie reviews. All things considered, our method offers a scalable and successful solution for content-based movie recommendation systems, advancing automated film classification and user-focused entertainment technologies.

References

Chang, M. W., Lee, K., Devlin, J., & Toutanova, K. (2019). BERT: Deep bidirectional transformer pre-training for language comprehension. The preprint arXiv is arXiv:1810.04805.

Jones, L., Gomez, A. N., Shazeer, N., Parmar, N., Uszkoreit, J., Vaswani, A., & Polosukhin, I. (2017). All you need is attention. Neural Information Processing System (NeurIPS) advancements.

Ott, M., Liu, Y., Joshi, M., Chen, D., Goyal, N., Du, J., & Stoyanov, V. (2019). RoBERTa: A BERT pretraining method that is robustly optimized. For the preprint, see arXiv:1907.11692.

Yang, Z., Salakhutdinov, R. R., Carbonell, J., Yang, Y., Dai, Z., & Le, Q. V. (2019). Generalized autoregressive pretraining for language comprehension is known as XLNet. Neural Information Processing System (NeurIPS) advancements.

"A Review Analysis on Recommendation System," by R. Nakka, Journal of Advanced Research in Dynamical and Control Systems, vol. 11, no. s2, 2019.

The International Journal of Advanced Science and Technology, vol. 29, no. 7, pp. 989-1000, 2020; R. Nakka, "Offering Recommendations on Netflix dataset by Associations among Users as Trust Metric,"

"An Advanced Neighbourhood approach of recommending movies on Netflix data by the combination of KNN and XGBoost," by D. R. K. K. Rajeswari Nakka and G. V. S. N. R. V. Prasad, Journal of Critical Reviews, vol. 7, no. 12, 2020.

"Hybrid Recommender System with Similarities and Associations among Users," by D. G. P. Rajeswari Nakka, Design Engineering, pp. 4585-4591, 2022.

McAuley, J., Pandey, R., & Leskovec, J. (2015). Inferring networks of substitutable and complementary products. *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 785–794.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. *Springer*.