Yoga Recommender System using RoBERTa Model

Madhav Parmar
madhav.parmar@hotmail.com

Abstract

The integration of wellness and medical knowledge has opened new avenues in personalized healthcare technologies. This research introduces a cross-domain Yoga Recommender System that leverages disease information derived from medical prescriptions to suggest relevant yoga practices. By employing the RoBERTa language model, the system effectively maps textual medicine inputs to appropriate yoga recommendations, addressing the cold-start and data sparsity problems typical in single-domain recommender systems. The proposed approach utilizes collaborative filtering, content-based methods, and transfer learning within a cross-domain framework to enhance personalization. Evaluation using metrics such as Label Ranking Average Precision (LRAP) and cosine similarity demonstrates the effectiveness of the system, achieving a recommendation accuracy of 97.2%. This work highlights the potential of combining healthcare and wellness domains using advanced NLP techniques to build intelligent, adaptable, and user-centric recommender systems.

1 Introduction

Recommender systems have become a cornerstone of modern technology, enabling personalized user experiences across various domains, such as e-commerce, entertainment, healthcare, and education. These systems leverage vast data to provide tailored suggestions, enhancing user engagement and satisfaction. However, traditional single-domain recommender systems often face challenges such as data sparsity, cold-start problems, and limited understanding of user preferences across diverse domains.

To address these limitations, Cross-Domain Recommender Systems (CDRS) have emerged as an advanced approach, enabling the transfer of knowledge between related or unrelated domains. By integrating data and features from multiple domains, CDRS enhances the accuracy and diversity of recommendations, thereby offering more comprehensive solutions. For instance, a system recommending yoga based on a user's search for specific medications exemplifies the power of CDRS in connecting seemingly disparate domains to improve user outcomes.

Recommender systems estimate users' preferences on items and recommend items that users might like proactively [6, 10]. Recommendation models are usually classified into three categories [6, 11]: collaborative filtering, content-based, and hybrid recommender systems. Collaborative filtering makes recommendations by learning from user-item historical interactions, either explicit (e.g., user's previous ratings) or implicit feedback (e.g., browsing history). Content-based recommendation is based primarily on comparisons across items' and users' auxiliary information.

This work explores the foundational aspects of recommender systems, the evolution of cross-domain methodologies, and their application in scenarios like healthcare and wellness. The study aims to identify key challenges, analyze existing approaches, and propose innovative strategies to overcome limitations, particularly in addressing the cold-start problem and data sparsity. By delving into metrics like Label Ranking Average Precision (LRAP) and techniques like Cosine Similarity, this work underscores the importance of accurate and effective evaluation measures for developing robust recommender systems.

Through this exploration, the research aspires to contribute to the growing field of cross-domain recommendations, paving the way for more intelligent and user-centric systems that can adapt to complex and dynamic user needs.

2 Literature Review

2.1 Introduction to Recommendation Systems

A recommendation system, also known as a recommender system, is an information filtering technology that assists users in finding relevant items or content based on their preferences, interests, and past behavior. It is widely used in various domains, including e-commerce, entertainment, social media, and more. The primary goal of a recommendation system is to provide personalized recommendations that enhance user experience, engagement, and satisfaction.

Recommendation systems leverage advanced algorithms and techniques to analyze large datasets, such as user profiles, item attributes, and historical interactions. These systems strive to understand user preferences and interests, as well as the characteristics of items or content, to make accurate and relevant recommendations.

Evaluation of recommendation systems is essential to measure their performance and effectiveness. Metrics like precision, recall, accuracy, and mean average precision are commonly used to evaluate the quality of recommendations and compare different algorithms.

2.2 Fundamental Steps to the Recommender System

(i) **Define the objective**
Identify the purpose of your recommendation system. Determine what you want to recommend and to whom, whether it's products, books, articles, or any other items.

(ii) **Pre-process data**
Collect relevant data about users and items. Cleanse and pre-process the collected data. This involves handling missing values, removing noise, normalizing data, and transforming it into a suitable format for analysis.

(iii) **Implement personalization**
Customize recommendations for individual users by incorporating user-specific features

and preferences. Personalization can significantly enhance the user experience and increase engagement.

(iv) **Select recommendation algorithm**
Choose an appropriate recommendation algorithm based on your objective and available data. Common algorithms include collaborative filtering, content-based filtering, matrix factorization, and deep learning-based models.

(v) **Incorporate feedback**
Continuously gather feedback from users to enhance the recommendation system. Feedback can be collected through explicit ratings, implicit feedback (e.g., clicks, purchase history), or user surveys. Incorporate this feedback to refine the model and enhance recommendations.

2.3 Types of Recommendation Systems

- **Content-based Filtering**
  This type of recommendation system uses the characteristics and attributes of items to make recommendations. It analyses user preferences and recommends items that are similar to the ones the user has liked in the past. For example, if a user has searched medicine (e.g., Asthalin), the system will recommend yoga (e.g., Cat-cow, low lunge) which benefits the user.

- **Collaborative Filtering**
  Collaborative filtering-based recommendation methods [13] make full use of the behavior information and preference information generated by the user in the past without using the user's personal information and product description information, such as the user's rating of the item to generate the recommended item. Collaborative filtering considers the preferences and behavior of multiple users to make recommendations. It identifies similarities in user preferences and recommends items that users with similar tastes have liked. Recommendations for items that are new to the catalog are therefore considerably weaker than more widely rated products, and there is a similar failing for users who are new to the system [4]. There are two main types of collaborative filtering:

  – *User-based Filtering:* This approach finds users with similar preferences and recommends items to users based on similarity to other users.

  – *Item-based Filtering:* This approach identifies similar items based on user preferences and recommends items that are similar to the ones a user has liked.

- **Hybrid Recommendation Systems**
  Hybrid systems combine multiple recommendation techniques to provide more accurate and diverse recommendations. For example, a hybrid system may use both content-based and collaborative filtering approaches to leverage the advantages of both methods.

2.4 Challenges and Issues

This proposal's coverage of conceptual work has identified issues in existing research. The following details highlight potential strategies for the mentioned issues.

2.4.1 Functional Issues

Functional issues refer to problems associated with the functionality of recommendation system algorithms. These issues can affect the effectiveness and usability of the recommender system.

- **Long Tail**
  The system usually recommends popular items neglecting unpopular ones. On the other hand, items generally have different levels of exposure to users. Hence, the recommendation system may be skewed towards particular items favored. Training new recommender systems from interaction data available by the previous model creates a feedback loop that usually affects diversity in recommendations.

- **Cold Start Problem**
  With the entry of a new user or item, the system lacks awareness of the user's preferences or item's ratings, making it difficult to suggest accurate recommendations with limited knowledge or minimal interaction. Cold start can be problematic in multiple domains because a service in one domain needs to communicate with a service in another. Session-based recommendation systems alleviate cold start problems by providing personalized recommendations based on the user's current session or browsing behavior.

- **Shieling Attack**
  Shieling attack is a type of attack where a malicious user profile and item description is injected to alter the review and rating decision of the recommender system. Such an attack distorts the recommender process to promote or demote a particular product.

2.4.2 Non-Functional Issues

Non-functional issues in recommendation systems are not directly related to functional requirements but impact overall performance, usability, and reliability.

- **Performance**
  The recommendation system requires significant computational resources when dealing with large and complex information.

- **Scalability**
  With explosive data growth, the recommendation system should be able to provide timely and relevant recommendations.

- **Accuracy**
  The system's performance on user-item interaction is evaluated based on different accuracy measures such as MAP, MRR, Precision, and Recall.

- **Data Sparsity**
  This refers to situations where available data about user interaction with items is sparse.

3 Cross-Domain Recommendation

Cross-domain recommendation transfers knowledge across domains based on similarity of users and items. It combines information (e.g., reviews and ratings) from multiple source domains and transfers it to a target domain, primarily to overcome the drawbacks of single-domain recommendation systems (SDRS). SDRS are often unable to capture the full spectrum of a user's interests and evolving preferences. Generally, domains are defined at four levels: attribute level, type level, hem level, and system level. At each level, different information sets are available, and cross-domain recommender systems (CDRS) often utilize information from different levels to handle cold-start problems in recommendations. Some existing works highlight the need for CDRS to alleviate data sparsity issues as well.

Cross-domain recommendation was proposed to combat the long-standing data sparsity problem by leveraging feedback or ratings from multiple domains to improve recommendation accuracy collectively [12].

In recent years, multilayer perceptron (MLP), a neural network architecture, has been used to learn non-linear mapping functions across domains. In these approaches, MLP takes the user latent factor in the source domain as input and outputs the user latent factor in the target domain. Different methods, such as knowledge transfer, feature engineering, hybrid models, meta-learning, and transfer learning, have also been implemented to develop cross-domain recommendation systems for various application domains.

For example, consider an online platform recommending both medicine and yoga. A user who frequently searches for medicine related to pneumonia may have their preferences transferred to the yoga domain. The system might use the user's medicine requirements to recommend relevant yoga poses or utilize shared features like medicine specifications and yoga posture characteristics to suggest yoga that matches their needs.

We cannot estimate a promising latent factor for making recommendations unless we have enough information about a user or item in the target domain. However, an affine latent factor in the target domain can be obtained by using the mapping function between source and target domains along with the latent factor learned in the source domain. For instance, if we denote the latent factor in the target domain as $\hat{U}_{it}$, it can be computed as:

$$\hat{U}_{it} = f(U_{is}; \theta), \tag{1}$$

where U_{is} is the latent factor in the source domain, and $f(\cdot; \theta)$ is the mapping function. The recommendation is then made based on this affine latent factor.

3.1 Example Process

(i) **Collect Data**
Gather user requirements and medication details.

(ii) **Analyze Preferences**
Identify the user's disease.

(iii) **Transfer Knowledge**
Use this preference to recommend relevant yoga.

(iv) **Refine Recommendations**
Continuously adjust recommendations based on new data from both medicine and yoga.

3.2 Current Approaches of Cross-Domain Recommendation Systems

Several approaches are used in developing cross-domain recommendation systems, including:

- Hybrid Approaches
- Shared Nearest Neighbor
- Cross-domain Collaborative Filtering
- Joint Factorization Models

3.3 Working Steps of Cross-Domain Recommendation System (CDRS)

(i) Cross-domain recommendation systems collect data on user-item interactions from multiple domains. Data may be selected based on the match of user profiles and item attribute information.

(ii) Common techniques like collaborative filtering, content-based methods, and matrix factorization are used to extract latent features from data.

(iii) Transfer learning techniques are used for latent feature mapping across multiple domains.

(iv) Finally, the recommendation model is used to generate recommendations from multiple domains.

3.4 Key Concepts

- **Enhanced Data Utilization**
  By integrating data from various domains (e.g., medicine and yoga), cross-domain systems can make more informed recommendations. For example, if a user is diagnosed with insomnia, the system might suggest yoga therapy based on this related information.

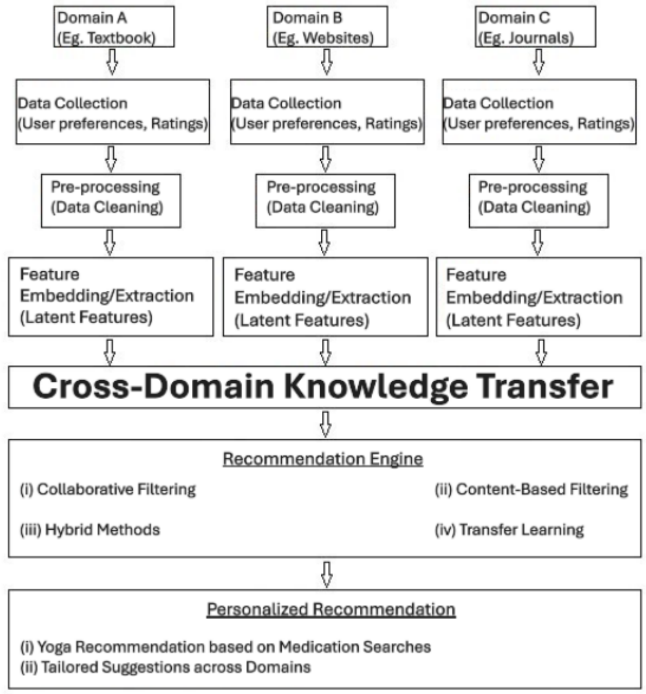

Figure 1: Diagram illustrating the Cross-Domain Recommendation System (CDRS) architecture.

- **Cold-Start Problem Mitigation**
  New users or items with insufficient data can benefit from existing data in related domains. For instance, a new user's medicine prescription might be used to recommend yoga even before they have searched for yoga.

- **Domain Definitions**
  Domains can be defined in various ways, such as broad categories (e.g., medicine vs. yoga) or more specific subcategories (e.g., allopathy vs. homeopathy). Understanding the context and nature of these domains is crucial for effective recommendations.

4 Related Work

RoBERTa is an advanced language representation model that builds on the foundation of BERT. It enhances BERT by utilizing more data, larger batch sizes, and longer training sequences, leading to improved performance. The RoBERTa training process has improved published BERT results on both GLUE and SQuAD benchmarks by training over larger datasets and for longer periods [9,12]. RoBERTa matches a score of 88.4 on the public GLUE leaderboard, achieving 88.5 after extended training. It sets new state-of-the-art results on 4 out of 9 GLUE tasks, including MNLI, QNLI, RTE, and STSB.

Cross-domain recommendation can assist target domain recommendations with knowledge learned from source domains, providing a desirable solution to common problems in recommender systems [1].

Matrix factorization (MF) characterizes both items and users by vectors of latent factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation [6]. MF associates each user and item with a real-valued latent feature vector. Let $\mathbf{p}_u$ and $\mathbf{q}_i$ denote the latent vectors for user u and item i, respectively; MF estimates an interaction y_{ui} as the inner product of $\mathbf{p}_u$ and $\mathbf{q}_i$ [5].

$$\hat{y}_{ui} = f(u, i|\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u^T \mathbf{q}_i = \sum_{k=1}^{K} p_{uk} q_{ik},$$

In this study, the model achieved a recommendation accuracy of 97.2%.

5 Result

In this study, we developed a yoga recommender system using the RoBERTa model. The model was trained to recommend yoga based on medicine, as both share a common domain of disease, enabling the application of cross-domain recommendation systems (CDRS). The evaluation demonstrated that the system effectively bridges the healthcare and wellness domains, achieving a recommendation accuracy of 97.2%. This result highlights the potential of leveraging advanced natural language processing techniques within a cross-domain framework to provide personalized and accurate yoga recommendations tailored to individual medical conditions.

References

[1] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018, doi: 10.1109/ACCESS.2018.2880197.

[2] Man, Tong, Shen, Huawei, Jin, Xiaolong, and Cheng, Xueqi. "Cross-Domain Recommendation: An Embedding and Mapping Approach," *IJCAI*, 2017, pp. 2464–2470.

[3] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. "Deep Learning Based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys*, vol. 52, no. 1, Article 5, Jan. 2020.

[4] R. Burke, "Hybrid Web Recommender Systems," in *The Adaptive Web*, Lecture Notes in Computer Science, vol. 4321, Springer, Berlin, Heidelberg, 2007, pp. 377–408.

[5] Xiangnan He et al., "Neural Collaborative Filtering," in *Proc. of WWW '17*, 2017, pp. 173–182.

[6] Francesco Ricci, Lior Rokach, and Bracha Shapira, "Recommender systems: introduction and challenges," in *Recommender Systems Handbook*, 2015, pp. 1–34.

[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[8] A. K. Pal et al., "Yoga Recommendation System Using Machine Learning Techniques," in *1st Int. Conf. on Advances in Computing, Communication and Networking (ICAC2N)*, Greater Noida, India, 2024, pp. 772–777.

[9] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[10] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[11] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, Cambridge University Press, 2010.

[12] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. of SIGKDD*, 2008.

[13] R. Sharma, D. Gopalani, and Y. Meena, "Collaborative filtering-based recommender system: Approaches and research challenges," in *Int. Conf. on Computational Intelligence & Communication Technology*, 2017, pp. 1–6.