**Yoga Recommender System from disease 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’ preference on items and recommend items that users might like to them proactively [6, 10]. Recommendation models are usually classified into three categories [6, 11]: collaborative filtering, content based and hybrid recommender system. 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 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

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, 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.

2.1 Fundamental steps to the Recommender System

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

**(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.

2.2 Types of Recommendation Systems

**(i) 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 (eg. Asthalin), the system will recommend yoga (eg. Cat-cow, low lunge) which benefits the user.

**(ii) Collaborative Filtering** Collaborative filtering-based recommendation methods [13] makes 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.

**(iii) 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.

2.3 Challenges and Issues

This proposal's coverage of conceptual work has identified the issues in existing research work. The main purpose of following details is to highlight the potential strategy for mentioned issues.

2.3.1 Functional Issues

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

**(i) 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 the particular items favored. In these settings, training new RSs from interaction data available by the previous model makes a feedback loop that usually affects the diversity in recommendations also.

**(ii) Cold Start Problem** With the entry of a new user or item, the system isn't aware of the user preferences (lack of historical interaction with the user)/ Item's rating (lack of feedback/rating from consumers), so it becomes difficult for the systems to suggest accurate recommendations with limited user/item knowledge (or minimal interaction). A cold start problem can be problematic when it comes to 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 recommendation based on the user's current session or browsing behavior.

**(iii) 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 alters the recommender process to promote and demote a particular product.

2.3.2 Non-Functional Issues

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

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

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

(iii) **Accuracy** The performance of the recommendation system on user-item interaction is evaluated based on different accuracy measures such as MAP, MRR, Precision, and Recall.

(iv) **Data Sparsity** It refers to the situation where the available data about user interaction with an item is sparse.

3. Cross-Domain Recommendation

Cross Domain Recommendation transfers knowledge across domains based on similarity of user and item. It combines information (for e.g., review and rating) from multiple source domains and transfers it to target domain, primarily to overcome the drawback of single domain recommendation system (SDRS). SDRS is unable to capture the full spectrum of user's interest and evolving preferences. Generally, domains are defined at four levels: Attribute level, Type level, Hem level, and System level. In each level, different information sets are available, and CDRS often utilizes information from different levels of a domain to handle Cold-Start problems in recommendation. Few existing works highlight the needs of CDRS to alleviate data sparsity issues as well.

Cross-domain recommendation was proposed to combat the long-standing data sparsity problem, which leverages feedbacks or ratings from multiple domains to improve recommendation accuracy in a collective manner. [12]

In recent years, Multilayer Perceptron (MLP), a Neural Network ecosystem, is used to learn non-linear mapping functions across domains. In these, MLP takes the user latent factor in source domain as input and user latent factor in target domain as output. Different approaches, such as Knowledge transfer, Feature Engineering, Hybrid model, Meta and Transfer learning have also been implemented to develop the Cross-domain recommendation system for different application domains.

For example: Imagine an online platform that recommends both medicine and yoga. A user who frequently searched medicine for 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 use shared features like the medicine specification and yoga posture characteristics to suggest yoga that match their need.

We cannot estimate a promising latent factor for it to make recommendations unless we have enough information about a user or item in the target domain. An affine latent factor for it in the target domain can be obtained, nevertheless, by using the mapping function between the source and target domains in addition to the latent factor learned for it in the source domain. For instance, if we have a user interface (*Û*it) in the target domain, we can use the following equation to determine its affine factor. [2]

*Û*it = *f* (*U*is ; *θ*),

where *U*is is its latent factor in the source domain and *f* (· ; *θ*), is the mapping function. Next, the recommendation is done base on this affine latent factor.

**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.

3.1 Current approaches of Cross-domain Recommendation System

Cross-domain recommendation aims to provide personalized recommendations across multiple domains. There are several approaches used in developing such a system.

(i) Hybrid Approaches

(ii) Shared Nearest Neighbor

(iii) Cross-domain Collaborative Filtering

(iv) Joint Factorization Model

3.2 Working Steps of Cross-domain Recommendation System (CDRS)

(i) Cross-domain recommendation system collects data on user-item interaction from multiple domains. Data may be selected according to the match of user profile, 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 technique is used for latent features mapping across multiple domains

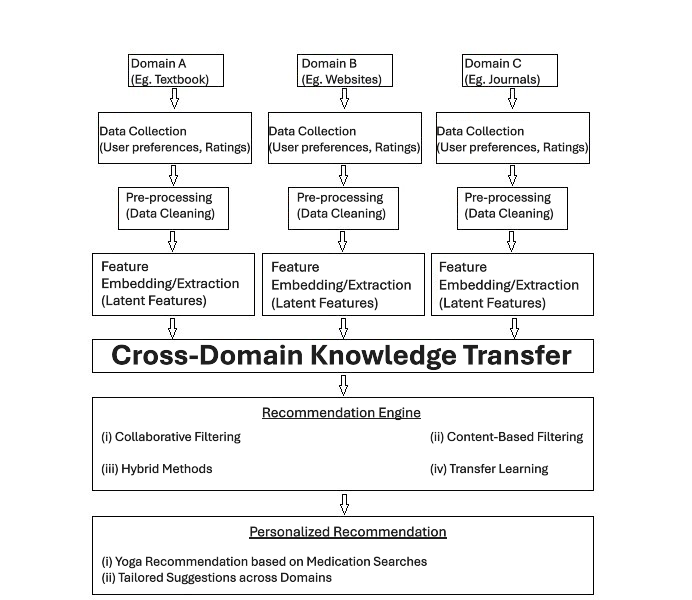
(iv) Finally, the recommendation model is used to generate recommendation from multiple domains.

**3.2.1 Key Concepts**

**(i) 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.

**(ii) 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 yoga.

**(iii) 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.



**Related Work**

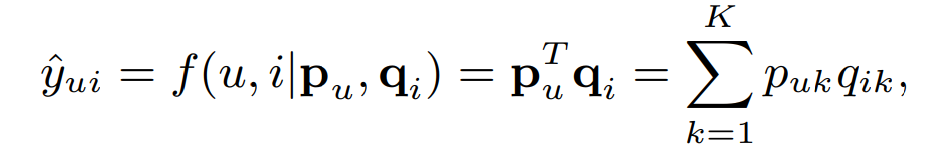
RoBERTa is an advanced language representation model that builds on the foundation of BERT. It enhances BERT. By utilizing more data, larger batch sizes, and longer training sequences, leading to improved performance.

The RoBERTa training process enhances the published BERT results on both GLUE and SQuAD during training data. RoBERTa matches the 88.4 on the public GLUE leaderboard with a score of 88.5 after being trained for a longer period of time over more data. [12, 9] On 4/9 of the GLUE tasks—MNLI, QNLI, RTE, and STSB—it sets a new state-of-the-art.

Cross domain recommendation can assist target domain recommendation with the knowledge learned from source domains, provides a desirable solution for these problems.[1]

Matrix factorization characterizes both items and users by vectors of 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 vector of latent features. Let **p**u and **q**i denote the latent vector for user u and item i, respectively; MF estimates an interaction yui as the inner product of **p**u and **q**i: [5]

**

The accuracy for model to recommend yoga is 97.2.

**Result**

In this study, we developed a yoga recommender system using RoBERTa model. The model was trained to recommend yoga on the basis of medicine, both share a common domain of disease from which they apply CDRS.

**References**

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

[2] Man, Tong & Shen, Huawei & Jin, Xiaolong & Cheng, Xueqi. (2017). Cross-Domain Recommendation: An Embedding and Mapping Approach. 2464-2470. 10.24963/ijcai.2017/343.

[3] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 52, 1, Article 5 (January 2020), 38 pages. https://doi.org/10.1145/3285029

[4] Burke, R. (2007). Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72079-9\_12

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[6] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In Recommender systems handbook. 1–34

[7] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in Computer, vol. 42, no. 8, pp. 30-37, Aug. 2009, doi: 10.1109/MC.2009.263.

[8] A. K. Pal, A. R. Parashar, A. Yadav, A. Yadav and M. Pachori, "Yoga, Recommendation, System using, Machine Learning Techniques," 2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N), Greater Noida, India, 2024, pp. 772-777, doi: 10.1109/ICAC2N63387.2024.10895752.

[9] Liu, Yinhan & Ott, Myle & Goyal, Naman & Du, Jingfei & Joshi, Mandar & Chen, Danqi & Levy, Omer & Lewis, Mike & Zettlemoyer, Luke & Stoyanov, Veselin. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. 10.48550/arXiv.1907.11692.

[10] Gediminas Adomavicius and Alexander Tuzhilin. 2005. 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 17, 6 (2005), 734–749.

[11] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. Recommender systems: an introduction.

[12] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In SIGKDD ’08, 2008.

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