Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis MASTER IN DATA SCIENCE AND MARKETING ANALYTICS

Alleviate Data Sparsity and Cold Start Problems with Transformer-based Hybrid Recommender System

Zhujun Tan (**610923**)

Supervisor:	**Carlo Cavicchia**
Second assessor:	**Radek Karpienko**
Date final version:	July 14, 2023

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Acknowledgements

I would like to express my gratitude to Dr. Carlo Cavicchia for his unwavering guidance and support throughout this journey. I am truly grateful to have had him as my supervisor, as he consistently provided timely and valuable feedback, pushing me to choose challenging topics for my master's thesis.

Abstract

This study aims to investigate the effectiveness of three Transformers (BERT, RoBERTa, XLNet) in handling data sparsity and cold start problems in the recommender system. We present a Transformer-based hybrid recommender system that predicts missing ratings and extracts semantic embeddings from user reviews to mitigate the issues. We conducted two experiments on the Amazon Beauty product review dataset, focusing on multi-label text classification under sparse data and recommendation performance in user cold start settings. Our findings demonstrate that XLNet outperforms the other Transformers in both tasks, and our proposed methods show superior performance compared to the traditional ALS method in handling data sparsity and cold start scenarios. This study not only confirms transformers' effectiveness under cold start conditions in recommender systems but also highlights the need for further study and improvement in the fine-tuning process.

Contents

List of Figures

Chapter 1

Introduction

Recommendation systems play a significant role in the e-commerce business since they can improve customer satisfaction through personalized recommendations and grow the business online (Alamdari et al., 2020). Collaborative Filtering (CF) is widely recognized as one of the most popular and effective methods for recommendation systems, employed by major companies such as Amazon, eBay, and YouTube (Blake, 2017; Feng et al., 2021). However, to have high-quality recommendations, the company must overcome the data sparsity and cold start problem which are restrictions of the CF algorithm (Su & Khoshgoftaar, 2009). The issue of data sparsity arises when current users contribute only a few ratings, leading to sparse interactions between users and items. On the other hand, when a new user enters the system, the system still cannot make recommendations due to a lack of ratings or limited purchase history (Su & Khoshgoftaar, 2009). In this regard, data sparsity exacerbates the cold start problem, as the sparse data availability further hinders the system's ability to generate accurate and reliable suggestions for new users (Gope & Jain, 2017; Rashid et al., 2002, 2008; du Boucher-Ryan & Bridge, 2005; Roy & Dutta, 2022). Failing to provide personalized recommendations to new users can result in a loss of interest and their departure from the business platform (Gope & Jain, 2017; Feng et al., 2021). Consequently, effective solutions are required to resolve or alleviate the new user cold start problem to establish engagement from users and increase conversion rate in e-commerce businesses. Hybrid filtering methods have been introduced to combine the strengths and overcome the limitations among collaborative filtering, content filtering, and other pure recommender methods (Jannach et al., 2010; Roy & Dutta, 2022; Isinkaye et al., 2015; Lika et al., 2014). Previous research has shown that the combination of multiple techniques improves the accuracy of the model and alleviates data sparsity and cold-start problems (Basiri et al., 2010; Kardan & Ebrahimi, 2013; Rubtsov et al., 2018; Penha & Hauff, 2020; Feng et al., 2021). Recently, researchers have acknowledged user reviews as valuable sources for recommending processes and incorporated them into recommendation generation (Chen et al., 2015). Various studies have combined the benefits of hybrid recommender systems and customer reviews. For example, researchers proposed to use sentiment analysis on reviews and predict the rating score to overcome the cold start problem (Cumbreras et al., 2013; Osman et al., 2021). Nevertheless, the method failed to provide the true meaning of the sentences in customer reviews since it predicted the missing scores from only negative and positive words in the sentences. This limitation in traditional text analytic methods has led to the emergence of Transformers such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019). The models can capture contextual information and excel in complex linguistic tasks such as text generation, machine translation, and question-answering (Géron, 2022). As a result, it is interesting to explore the application of different transformer architectures (e.g., BERT, XLNet, and RoBERTa) based on user reviews in recommender systems and evaluate their effectiveness in facilitating recommendations in cold start scenarios. This leads to a research question in the following:

How effectively can Transformers alleviate data sparsity and user cold start problems in the recommender system?

In order to address the main research question, several sub-questions have been formulated, including:
1. Which Transformers perform best at multi-label text classification of missing ratings based on user reviews?
2. Does incorporating sentence embeddings of user reviews as side information improve the recommender system in cold-start settings?
3. Does a specific type of sentence embedding provide better results in recommender systems? And what are the reasons behind the variations?

The study is relevant for managers in several ways. First, this research provides potential solutions to handle the limitations of the traditional recommender system for managers. Managers can use insights from this study to overcome data sparsity and cold start problems in their own systems and data sets. Second, the findings of this study demonstrate the effectiveness and the limitations of different architecture of Transformers. Managers can leverage this knowledge and adapt the most appropriate Transformer into their system more efficiently under the company's business requirements and goals. Third, this study demonstrates the benefits of incorporating reviews into the recommender system. Managers can adapt their own systems and data sets to improve the personalization and relevance of the recommendations to their customers. Consequently, By improving the quality of recommendations, businesses can anticipate a notable boost in sales and revenues and reduce search costs (Hinz & Eckert, 2010). Furthermore, implementing such innovative recommender system technology positions the business at the forefront of the industry, showcasing its commitment to staying ahead of the curve in meeting customer needs and preferences.

The current literature on recommender systems primarily focuses on rating prediction and sentiment analysis techniques (Feng et al., 2021; Cumbreras et al., 2013; Osman et al., 2021; C. N. Dang et al., 2021). However, these approaches have limitations as they can be inaccurate and biased. Additionally, recent research in recommender systems has explored the use of Transformers to address cold start problems, with a particular emphasis on sentiment analysis and rating prediction (C. N. Dang et al., 2021; E. Dang et al., 2022; Zhuang & Kim, 2021). However, there is still a lack of research fully exploiting the potential benefits of transformer models in recommender systems. Furthermore, existing studies often rely on traditional word embedding techniques such as Word2Vec and LightFM word embeddings (Kula, 2015; Nguyen et al., 2020), which focus on individual words and may fail to capture the underlying preferences expressed in user reviews. This limitation highlights the need for novel approaches that leverage the semantic context of user reviews and incorporate both ratings and sentence embeddings to enhance the quality of recommendations. Overall, our study fills a crucial gap in the literature by addressing the limitations of previous approaches and offering a novel method that leverages Transformers, ratings, and sentence embeddings to improve the accuracy and the relevance of recommendations. By advancing the understanding and application of Transformers in the context of recommender systems, our research contributes to the existing body of knowledge in this field.

In this paper, we conducted two experiments to investigate the effectiveness of the proposed method under data sparsity cold start settings. We dropped the ratings by 30% to simulate the data sparsity problems. To answer the first sub-question, we compared BERT, RoBERTa, and XLNET on multi-label classification for predicting missing rating values based on user reviews to answer the first research question. We evaluated the performances based on accuracy, weighted F1, and Area Under ROC Curve. To address the second and third sub-questions, we incorporated sentence embeddings from three Transformers and rating results from the best model into collective matrix factorization. The accuracy of the recommendations was evaluated using three widely used rank metrics: precision and recall at N, as well as mean average precision (mAP). We compared results of three Transformer-based Hybrid Approaches with Alternating Least Square (ALS) technique.

The thesis is structured as follows: In Chapter 2, we discuss the relevant literature review concerning recommender systems and the current techniques used to solve cold start problems. Chapter 3 presents the descriptive analysis of the data and the data preprocessing steps undertaken before experimenting. Chapter 4 focuses on the methodology employed in the experiments and provides an overview of the evaluation metrics used. In Chapter 5, we present the results obtained from both experiments. Moving on to Chapter 6, we provide the interpretation behind the results, which directly relate to our research questions, and discuss the academic and managerial implications of our study. Additionally, we thoroughly examine the limitations and future research directions. Finally, Chapter 7 concludes the thesis by summarizing the key findings, answering the research questions, and providing limitations and recommendations for future research. Furthermore, this chapter emphasizes the contribution of our study to the field of recommender systems under the context of cold start scenarios.

Chapter 2

Literature Review

This section aims to review the current literature on related topics in the research questions and identify gaps in the existing literature, particularly in recommender techniques under cold start settings. The review focuses on the three most prevalent recommender systems: Collaborative Filtering (CF), Content-based Filtering (CBF), and hybrid filtering, along with previous techniques used to address cold start problems in each method, and transformer neural networks.

2.1 Collaborative Filtering

2.1.1 Overview of the method

People rely on recommendations from others in everyday life through conversations, news, and online articles. Recommender systems capture this social process to filter and find the most relevant or interesting products for users, recommending them accordingly (Su & Khoshgoftaar, 2009). The fundamental idea of collaborative recommendation systems is to utilize historical user behavior and current community opinions to predict which products current users would likely find appealing (Jannach et al., 2010).

There are two types of collaborative approaches: memory-based and model-based approaches. In the early work of memory-based CF methods, the similarity between users and items is calculated through user rating data to make recommendations (Resnick et al., 1994) Consequently, there are two types of memory-based CF methods: user-based and item-based CF (Roy & Dutta, 2022).

In user-based CF, the system identifies user neighborhoods or similar users to the active user based on their interactions with other items in the past (Roy & Dutta, 2022; Jannach et al., 2010). As a result, the user rating on the new item is computed based on their similar users. The intuitive assumption is that if users have had similar preferences in the past, they should also have consistent preferences over time. On the other hand, the item-based CF system calculates the user rating for the new item using the weighted average of ratings across similar items in user history. The consumer will be given the option to purchase the new item if it has a high similarity score to the previously rated items. Due to their simplicity and efficiency, memory-based approaches have been frequently employed on e-commerce websites (Linden et al., 2003; Hofmann, 2004). However, these methods can be slow since they use the complete dataset every time they generate new predictions. Additionally, the computation based on similarity values may undermine the credibility of the results when there are limited interactions between users and products (Su & Khoshgoftaar, 2009).

To overcome the limitations of memory-based recommendations, model-based CF techniques have been proposed. These techniques include machine learning and data mining approaches to learn patterns from the provided data and generate predictions for unrated items (Breese et al., 1998; Su & Khoshgoftaar, 2009; Roy & Dutta, 2022).

One of the most successful methods in terms of prediction accuracy in collaborative filtering is Matrix Factorization (Takács et al., 2008) which includes techniques such as Singular Value Decomposition (Abdi, 2006), Non-negative Matrix Factorization (NMF) (Lee & Seung, 2000), Probabilistic Matrix Factorization (PMF) (Salakhutdinov & Mnih, 2008), and Max-Margin Matrix Factorization (MMMF) (Rennie & Srebro, 2005; Srebro et al., 2004). Additionally, other model-based approaches are latent factor models (Krestel et al., 2009; Sarwar et al., 2000; Goldberg et al., 2001), clustering CF models (Ungar & Foster, 1998; Xue et al., 2005; Crespo et al., 2011), and Linear Singular Value Decomposition (Zahálka et al., 2015; Ayata et al., 2018). Although model-based methods aim to address data sparsity issues encountered in memory-based methods, certain limitations and data sparsity problems persist in some of these approaches. Latent factor models, for instance, still suffer from data sparsity due to their heavy reliance on user-item interactions. Proposed clustering methods often employ greedy techniques, which makes it difficult to find optimal clustering solutions, especially when dealing with large datasets (Su & Khoshgoftaar, 2009). Furthermore, linear SVM models, although not commonly used in collaborative filtering, may rely heavily on labeled data or user-item ratings for training and may exhibit low accuracy in certain datasets (Roy & Dutta, 2022). The next section discusses recent research in CF methods that aim to alleviate cold start problems.

2.1.2 Techniques to Alleviate Cold-start Problem

According to Zhang (2015), the traditional CF algorithms based on ratings have two main challenges which are data sparsity and the limited interpretability of numerical ratings. The author argued that textual reviews can provide product features and customer opinions and gain more insights than just a numerical rating. As a result, by extracting the pair of product features and the customer's opinion in each review, the algorithm worked under cold-start settings (Zhang, 2015). The author proposed generating a review corpus based on each product by integrating all the reviews and categorizing them into a specific product domain such as mobile phone. Consequently, phrase-level sentiment analysis was conducted on each review corpus to find each product feature word that is associated with its sentiment polarity. As a result, the algorithm was able to generate recommendations based on identified user features and sentiment lexicon.

The objectives of the proposed method were to alleviate cold-start problems and provide explanations of recommendations results. However, this paper only proposed the framework without implementation. Furthermore, with the construction of user/item profiles, the method was computationally expensive due to the large amount of data in real-world applications. In addition, this method may not work well when users provide either product features or their opinions in reviews.

C. N. Dang et al. (2021) proposed a new method to improve collaborative filtering by incorporating deep-learning sentiment analysis to predict the rating. The authors implemented a Bidirectional Encoder Representations from Transformers (BERT) pre-trained model on review text to get word embeddings from Amazon fine food and movie reviews. The authors fed feature vectors into the combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models and obtained five sentiment classes (very positive; positive; neutral, negative; very negative) based on each review. Consequently, the authors put the classification results into five user-based collaborative filtering methods, which were Singular Value Decomposition (SVD), Non-Negative Matrix Factorization (NMF), and optimized Singular Value Decomposition (SVD++), and compared the proposed methods with all CF algorithms without sentiment analysis. The rating prediction results showed that the proposed method provided a higher accuracy compared with the results from CF traditional methods without sentiment analysis (C. N. Dang et al., 2021). Moreover, the evaluation from top-N recommendations also demonstrated that the proposed methods yielded higher Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) compared with the baseline models (C. N. Dang et al., 2021)).

The experiment by C. N. Dang et al. (2021) demonstrated a significant improvement in collaborative filtering when incorporating deep learning-based sentiment analysis in the models. However, the process of getting sentiment results by exploiting three different deep learning models may not be efficient and can be time-consuming, especially when dealing with large datasets.

2.2 Content-based Filtering

2.2.1 Overview of the method

One of the challenges of applying collaborative filtering is to find a data set with the complete ratings provided by users (Isinkaye et al., 2015). Moreover, relying solely on the ratings makes the algorithms unable to provide personalized recommendations based on specific preferences for each user (Jannach et al., 2010). Content-based filtering (CBF) uses item descriptions and user profiles, which extract item features based on the user's past purchases, to recommend items. This approach involves finding positive similarities between two documents and leveraging those similarities to generate meaningful recommendations (Isinkaye et al., 2015; Lops, 2014; Roy & Dutta, 2022). The widely used methods for text analysis are Term Frequency Inverse Document Frequency (TF/IDF), Probabilistic classifiers, and nearest neighbor models (Isinkaye et al., 2015). As a result, CBF can recommend new items when there are no ratings, and it can quickly adjust the recommendation with user preferences. However, the effectiveness of the system relies on in-depth item descriptions and user profiles before utilizing the algorithms (Jannach et al., 2010; Roy & Dutta, 2022). Moreover, since the algorithm analyzes solely item features based on the user's previous purchases, the recommendations are limited to similar items in the past, ignoring other potential options (Isinkaye et al., 2015). The next section explores techniques that aim to alleviate cold-start problems and situations when there are not enough ratings provided by users.

2.2.2 Techniques to Alleviate Cold-start Problem

According to Lops (2014), most content-based filtering (CBF) techniques suffered from language ambiguity problems, where words or sentences in user and item profiles can have multiple meanings. The authors proposed a comprehensive survey of semantic techniques that can overcome such challenges and improve the accuracy of recommender systems. One of these techniques was Explicit Semantic Analysis (ESA), which was a top-down approach that utilized open encyclopedias like Wikipedia to improve item and user representations (Lops, 2014). The authors explained that Wikipedia was a collection of concepts and could be used to define the relationship between terms and articles in order to represent the meaning of the text. The sparse matrix was constructed, where the columns represent the concepts or titles of Wikipedia, and the rows represent the terms or words in the articles, capturing the relationship (Lops, 2014; Gabrilovich & Markovitch, 2014). Consequently, semantic interpretation vectors of individual terms in sentences or entire documents were computed based on the concepts in Wikipedia (Lops, 2014). This process helped to improve the ambiguity of words and uncover their meaning in a given context without the need for advanced technology tools (Lops, 2014; Gabrilovich & Markovitch, 2014). For example, with the proposed method, the sentence ”Researchers have found a potential cure for cancer” is related to concepts such as oncology, medical research, breakthroughs, medicine, and treatment options. While the proposed method generated more meaningful representations and worked in cold start settings, the technique heavily depended on free external knowledge, which may lack some concepts and may not be entirely accurate. As a result, the presence of outdated and incomplete concepts could affect the quality of recommendations.

In the recent work on CBF by Nguyen et al. (2020), a novel approach was proposed to address the lack of sufficient ratings in datasets. The experiment was conducted by using a crowdsourcing system to create an IMDB movie dataset through OMDB open API (Nguyen et al., 2020). Nguyen et al. (2020) extracted features titles, genres, directors, actors, and movie descriptions, and apply Word2Vec to the movie descriptions to obtain a semantic vector for movie descriptions. Consequently, the authors converted features, which are titles, genres, directors, and actors, into vectors and implement Jaccard similarity and vector soft cosine-based similarity to obtain the similarity between features and movies. In the experiment, movies that received similar scores of more than 5 for a given user were labeled with 1. The recommendation list included the movie list with the label. The evaluation metrics in the experiment were recall, accuracy, precision, and F1 (Nguyen et al., 2020). The authors compared the result of the proposed method with K-Nearest Neighbor (KNN) and Correlation-based Items Similarity (CIS). The result of the experiment showed that the proposed method overcome baseline models in all metrics in cold-start conditions (Nguyen et al., 2020). The experiment demonstrates the effectiveness of word embeddings in content-based filtering in a sparse data set. However, the recommendation list generated by the proposed method consists of movies similar to ones previously watched by users. As a result, it may overlook other potentially diverse options.

2.3 Hybrid Filtering

2.3.1 Overview of the method

Hybrid filtering methods have been introduced to combine the strengths and overcome the limitations among collaborative filtering, content filtering, and other pure recommender methods (Jannach et al., 2010; Roy & Dutta, 2022; Lika et al., 2014; Isinkaye et al., 2015). Previous research has shown that the combination of multiple techniques improves the accuracy of the model and alleviates data sparsity and cold-start problems (Basiri et al., 2010; Kardan & Ebrahimi, 2013; Rubtsov et al., 2018; Penha & Hauff, 2020; Feng et al., 2021). Hybrid filtering algorithms have mainly three hybrid designs: monolithic, parallelized, and pipelined-based designs(Jannach et al., 2010).

First, monolithic designs exploit either feature combination or feature augmentation in one recommender system and use the result as input into another recommender system (Jannach et al., 2010). For example, content-boosted collaborative filtering by Melville et al. (2002) applies feature augmentation and predicts pseudo-user ratings based on collaborative and content algorithms.

Second, parallelized hybrid designs apply two or more recommender techniques independently and combine results from each system into a final list of recommendations. The designs include three strategies which are mixed, weighted, and switching (Burke, 2002). A mixed strategy combines the results from two or more recommender systems and provides the recommendation list from each technique. For example, Zanker et al. (2007) propose a mixed hybrid method that yields different recommended bundles based on different techniques for different product categories in the tourism industry. Consequently, a weighted hybrid strategy applies the relative weight to each algorithm to combine the predictive power of all techniques (Jannach et al., 2010; Roy & Dutta, 2022). Claypool et al. (1999) implement a dynamic weighting scheme on the output of collaborative and content-based algorithms in the news industry. The switching strategy implements different recommender techniques based on the quality of recommendation results and user profiles. For instance, Billsus and Pazzani (2000) implement a switching hybrid based on two content-based algorithms and one collaborative filtering in the news domain. The algorithm will switch to collaborative filtering when it cannot find any related articles for users when using a content-based nearest-neighbor system (Billsus & Pazzani, 2000).

Third, pipelined hybrid designs exploit the strengths of each recommender algorithm and build on top of each other results sequentially before providing the final recommendation list for users (Jannach et al., 2010). For example, the Fab system developed by Balabanovic and Shoham (1997) builds a collaborative filtering recommender on top of the results from the content-based algorithm in online news. The content-based component builds user models based on users' interests in online articles (Balabanovic & Shoham, 1997). Consequently, the collaborative component recommends articles to users based on their nearest neighbors (Balabanovic & Shoham, 1997).

2.3.2 Techniques to Alleviate Cold-start Problem

One of the most widely used methods for hybrid filtering in e-commerce business is LightFM developed by Kula (2015)). The author proposes a new word embedding method to encode item and user content features and incorporate them into collaborative filtering with user-item interaction. Unlike Word2vec and GloVe embeddings, LightFM embeddings are based on user-item interactions. For instance, word embeddings of ball gowns and pencil skirts are closed together when they are liked by the same user. The model predicts the user and item based on the dot product of the user and item representations adjusted with user and item biases (Kula, 2015). The author trained the model with asynchronous stochastic gradient descent (Recht et al., 2011) on sparse datasets, which are MovieLens and CrossValidated. The author adjusted the data by removing 20% of the interactions. The result shows that LightFm with both item and user features performs better than traditional matrix factorization and a content-based model which uses a latent topic technique from item features (Kula, 2015). The experiment demonstrates the effectiveness of integrating metadata into CF algorithms in cold start settings (Kula, 2015). The algorithm is able to recommend immediately after new items and users enter the system because of the linear combination of user and item features. However, this method assumes that each feature has the same influence on users based on the sum of user and item representations. Moreover, it may provide inaccurate recommendations when dealing with irrelevant and low-quality content. As a result, the content component has to be preprocessed properly to reflect item attributes and user preferences.

Feng et al. (2021) proposed a new hybrid model that combines Probabilistic Matrix Factorization (PMF) for a rating-oriented approach and Bayesian Personalized Ranking (BPR) for a pairwise ranking approach. The proposed method aimed to alleviate new user cold start problems by utilizing both explicit and implicit feedback data. The authors used SVD to predict unrated items based on user-historical ratings. Consequently, predicted ratings and historical ratings were incorporated into the PMF model to extract explicit features. On the other hand, the authors exploited the BPR model to extract implicit features from implicit data, such as browsing history and click-through rate. Both features were fused and trained through a linear combination objective function that incorporates both models. The proposed method was evaluated on Movielens 100k, Movielens 1M, FilmTrust, and Ciao datasets in cold start settings where the number of ratings is less than 20 for each user. The results showed that the proposed method outperformed baseline models in cold start settings based on Precision, Recall, Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR).

Nevertheless, when setting the number of ratings for one item per user, the proposed algorithm was not able to provide recommendations to users due to extreme data sparsity. The research by Feng et al. (2021) demonstrated the effectiveness of using hybrid recommendations on both explicit and implicit data to alleviate new user cold start problems. Nevertheless, the authors could include content information such as item descriptions and review text to improve the accuracy of the recommendations in the future.

2.4 Transformer Neural Networks

Bahdanau et al. (2014) introduced the attention mechanism which allows the decoder to focus only on the relevant parts of encoded sequences before making predictions. The mechanism aggregated encoder inputs and calculates attention weights to determine the significance and relevance of each input at each step (Géron, 2022). As a result, it has overcome the short-term memory and vanishing gradients problems in Recurrent Neural Networks (RNNs). The attention mechanisms surpassed other state-of-art performances, especially in long sentences. This advancement has contributed to a revolution in deep learning until today.

Vaswani et al. (2017) extended the concept of attention mechanism and developed a groundbreaking neural network architecture called the Transformer. The model was based solely on self-attention without using recurrent or convolutional layers. The architecture includes multi-head attention layers in both encoder and decoder allowing the model to process multiple aspects of input information simultaneously. As a result, Transformer can capture long-range dependencies and handle complex sentences more accurately compared to previous natural language processing methods (Géron, 2022).

According to Devlin et al. (2019), the Transformer architecture had a significant limitation in language processing. The model could process tokens only from left to right within the self-attention layers. As a result, this unidirectional constraint limited the model to truly understand the meaning of the words within the context. The authors proposed Bidirectional Encoder Representations from Transformers (BERT) to alleviate the limitation in transformers by exploiting the Mask Language Model (MLM). To illustrate, the algorithm randomly masked words in each sentence and predicted the masked words based only on the context (Géron, 2022). The pre-trained BERT model has been trained on more than 3,000 million words from BookCorpus and English Wikipedia. The pre-trained model can be fine-tuned for downstream tasks to leverage previous knowledge on more specific tasks in different datasets. The authors demonstrated that BERT achieved state-of-art results in eleven NLP tasks based on GLUE, MultiNLI, and SQuAD datasets.

Following the BERT paper, many pre-trained language models have been proposed to achieve more state-of-art NLP tasks. XLNet was one of the models developed by (Yang et al., 2019). XLNet overcame the limitations of previous pre-trained language models by implementing permutation language as the training objective and improving the architecture of autoregressive language modelings based on the Transformer-XL model (Yang et al., 2019; Dai et al., 2019). The permutation language objective allowed the autoregressive model to utilize the context from both left and right sides and does not need to corrupt the data by masking words (Yang et al., 2019). As a result, the proposed algorithm combined the benefits of autoregressive language modeling and bidirectional context language.

Liu et al. (2019) argued that BERT was undertrained and has an unoptimized training process. Consequently, the authors introduced a Robustly Optimized BERT Pretraining Approach (RoBERTa). The model was trained on over 160 GB of uncompressed text with longer sentences. Moreover, the authors removed the next sentence prediction objective and exploit dynamic masking in the training process to increase the efficiency of the model. Experimental results on benchmark datasets demonstrated that the masked language objective in BERT was as effective as in recently proposed models like XLNet.

The latest research in recommender systems has shown potential progress in addressing cold start problems. However, there has been limited research that fully exploits the benefits of transformer models. It is interesting to explore the application of different transformer architectures (e.g., BERT, XLNet, and RoBERTa) in recommender systems and evaluate their effectiveness in facilitating recommendations in cold start scenarios.

Chapter 3

Data

The dataset is originally from the Amazon Review Data in 2018 (Ni et al., 2019). We used the Beauty category of the Amazon datasets, which are widely regarded as benchmarks in research papers. The dataset consists of 371,345 rows and 12 columns, with 324,038 unique user IDs and 32,586 unique item IDs. Moreover, the average number of words in the review text is 64 words.

3.1 Data Descriptive Statistics

Our goal is to investigate whether our proposed method can effectively handle situations where there are limited historical user ratings and scarce data. Therefore, the relevant variables are ratings, user ID, item ID, and review text. Figure 3.1 shows that the rating variable is highly imbalanced among users, with more than 60% of users giving a rating of 5. Figure 3.2 presents a bar chart displaying the number of users and their ratings, where more than 99% of users in the dataset have rated less than 10 times.

Moreover, the distribution of beauty products in Figure 3 illustrates that most items have been rated by only a few users or have received few ratings. The analysis between userID and ItemID suggests that the cold start setting is likely to appear in the data set due to limited historical user ratings.

Figure 3.3 shows a word cloud comparison between users who rated more than 3 (green) and users who rated less than 3 (orange). From the word cloud, it can be observed that words such as ”love,” ”skin,” ”hair,” ”body,” and ”perfect” appear frequently when users rate more than 3. On the other hand, words like ”money,” ”shave,” ”bad,” and ”waste” seem to occur more often when users rate less than 3.

3.2 Data preprocessing

The duplicated rows and the missing values of review text were dropped to get more accurate recommendations and facilitate the deep learning part using transformers. Moreover, we did one-hot encoding on the rating variable to 5 numerical variables (1-5). Consequently, users who rated more than 20 ratings were removed to construct cold start settings (Feng et al., 2021). The first experiment focuses on rating classification based on the reviews from transformers. As a result, the data is split randomly with a 70:15:15 approach into training, test, and validation data sets.

We excluded ratings randomly for 30% of users in the test set to depict a more challenging cold start scenario for the first experiment. Moreover, dummy variables were created for the rating variable, which is a necessary step for multi-label classifications in transformer models.

The second experiment aims to investigate whether sentence embeddings from transformers can enhance recommender systems when there are new users entered the system. As a result, we split the data based on unique user IDs to make sure that users in the training set and test set are not overlapped. Table 3.1 shows two final data sets for the second experiment with unique users, unique items, and the number of ratings.

Datasets	Unique Users	Unique Items	Number of ratings
Training set	1,548	1,814	5,700
Test set: users $\notin$ train, items $\in$ train	389	527	1,094

Table 3.1: The overview of training and test sets

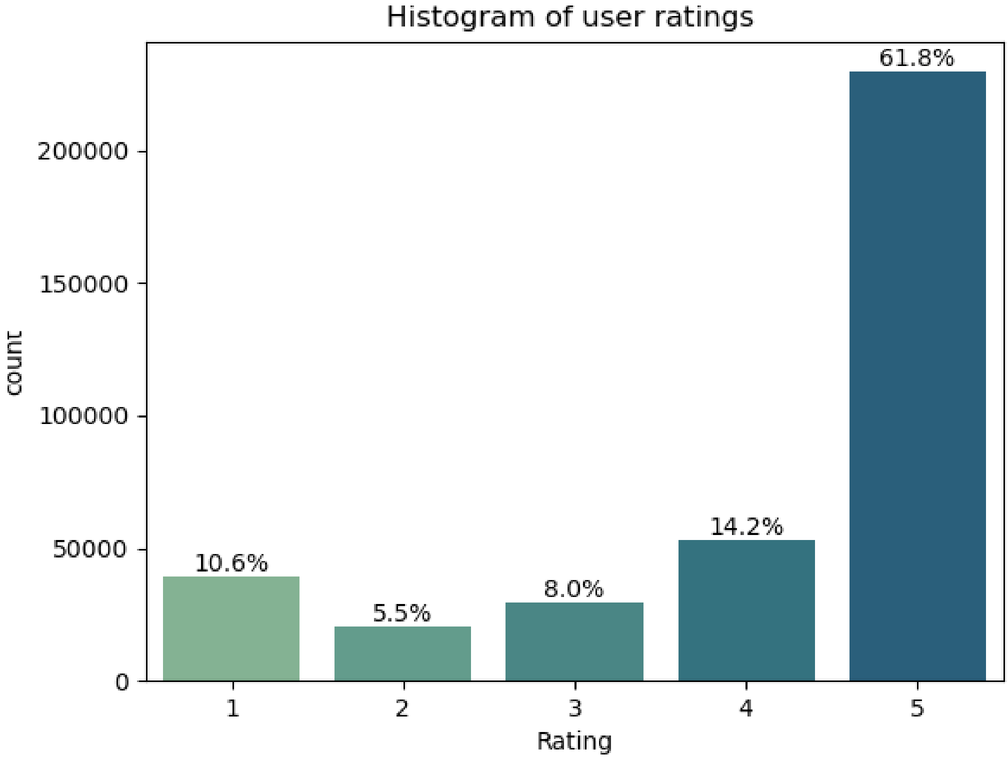


Figure 3.1: The histogram of user ratings

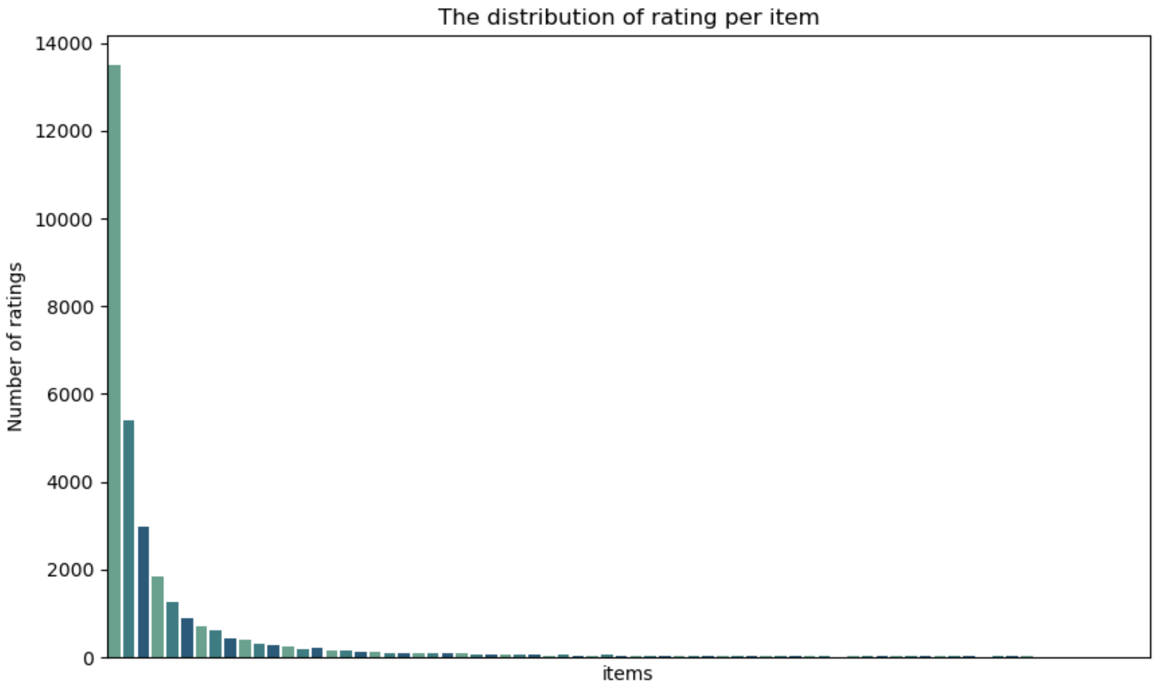


Figure 3.2: The distribution of rating per item

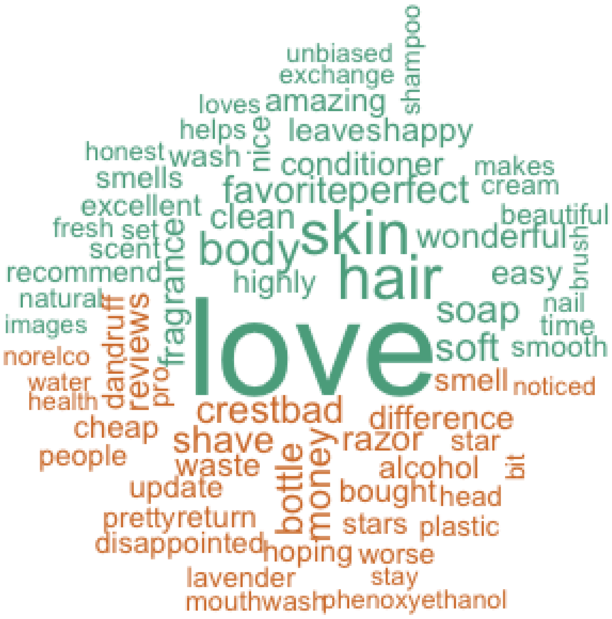


Figure 3.3: The world cloud comparison

Chapter 4

Methodology

This section compares BERT, RoBERTa, and XLNet in detail. We explore the strengths of these transformers and discuss their benefits in the recommender system. Consequently, we present the proposed method and two experiment setups.

4.1 Transformer Neural Networks

Long-term memory cells such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been introduced to improve short-term memory problems in RNNs (Géron, 2022). However, the problems persist when the networks process more than 100 input sequences at each step, such as long sentences (Géron, 2022). The main issue of long input sequences is the exploding gradient problem during backpropagation which compromises the ability to learn the information of neural networks. Moreover, the sequential processing of RNNs leads to inefficient computation and the inability to process in parallelization. Transformers mitigate these challenges by exploiting self-attention mechanisms and parallel processing in their architects. Figure 4.1 (left) depicts the original architect of the transformer developed by Vaswani et al. (2017). The aim of our first experiment is to process customer reviews and classify the rating labels for users who forgot to rate the products. As a result, we focus on the encoder at the left component of transformers which is the main part of BERT, RoBERTa, and XLNet. The model processes each word independently and sequentially across the layers while still capturing the meaning of the words in each sentence due to multi-head attention layers and positioning encoding in encoder and decoder layers as shown in Figure 4.1 (left).

4.1.1 Multi-head Attention

The model utilizes the dot product techniques developed by Luong et al. (2015) for the attention mechanism. However, to avoid exploding gradients on the softmax operation, the dot product is scaled by the square root of input dimensions in transformers. The attention function maps each input in three different roles (Vaswani et al., 2017). First, the value (V) represents the input vector that the algorithm uses in a weighted sum to derive output. Second, the query (Q) is the input vector of the current output matched against every other input vector. Third, the key (K) denotes the other input vector against which the query is matched. The attention formulation is as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.1}$$

Figure 4.1 (right) demonstrates the multi-head attention layer, which consists of multiple self-attention layers computed in parallel. The values, queries, and keys go through separate linear transformations to project word representation in different subspaces. Consequently, The algorithm is able to focus on different word characteristics and incorporate them into a stack of self-attention layers. All outputs obtained from the scaled dot products in each sub-attention head are concatenated and go through a final linear transformation. The multi-head attention layer is essential since it captures different relationships among words within the sentences. For example, in the sentence “The shampoo was not that bad.” The word “bad” is inverted by “not” and moderated by “that.” It also describes the property of shampoo. Therefore, the model can classify a higher rating by not based only on the word “bad” in the review. By utilizing different self-attention operations in a multi-head layer, the model can effectively capture the meaning of words correctly (Géron, 2022).

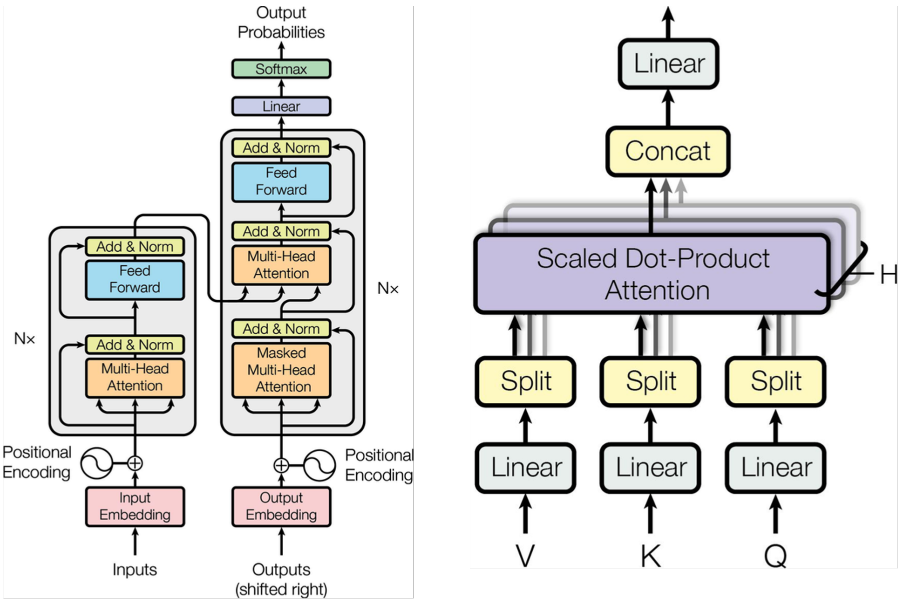


Figure 4.1: The Transformer Architect (Left) and Multi-head Attention (Vaswani et al., 2017)(Right)

4.1.2 Positional Encodings

The order of words influences the overall meaning of each sentence. For example, consider the sentences "The customer service was terrible, but the eye cream was excellent!" and "The eye cream was excellent, but the customer service was terrible!" By rearranging the word order, the meaning of the two sentences are opposite. In order to allow the self-attention blocks to capture the sequential structure without relying on recurrent or convolutional mechanisms, the positional encoding vectors are added on word embeddings before passing through a multi-head attention layer. Transformers use a sinusoidal positional encoding method to calculate fixed positional encodings for each word within sentences based on the combination of sine and cosine curves. This method provides a unique position for each word and allows the model to learn the relative position of words within the sentences. Based on the equation as follows, an input sequence has a length of L and p^{th} is the position of the word within the sequence, i is the dimension index map of each positional embedding in different frequency waves, and d indicates the dimension of the output embeddings. Moreover, a sine function corresponds to even positions $(2i)$ and a cosine function $(2i + 1)$ corresponds to odd positions.

$$PE(p, 2i) = sin\left(\frac{p}{1000^{\frac{2i}{d}}}\right) \tag{4.2}$$

$$PE(p, 2i+1) = cos\left(\frac{p}{1000^{\frac{2i}{d}}}\right) \tag{4.3}$$

According to Devlin et al. (2019), the Transformer architecture has a significant limitation in language processing. The model can process tokens only from left to right within the self-attention layers. As a result, this unidirectional constraint limits the model to truly understand the meaning of the words within the context.

4.2 BERT

BERT exploits the encoder component of the original transformers since the objective of the model is to learn bidirectional contextual representations of words, rather than generating sequence outputs. We used a pre-trained BERT model which had already been trained with large language datasets and fine-tuned it with the Amazon review dataset. It is more efficient and cost-effective to use the pre-trained models compared to building the model from scratch (Devlin et al., 2019). The model was trained on two unsupervised tasks simultaneously: Mask Language Model and Next Sentence Prediction (Devlin et al., 2019). This section discusses the training process for the model, the dataset preparation, and the fine-tuning process for our downstream task.

4.2.1 The pre-training

Mask Language Model (MLM)

Before the training phase, approximately 15% of the words in input sentences were corrupted Devlin et al. (2019). To address the word mismatch issue between pre-training and fine-tuning, 80% of the corrupted words were replaced with special [MASK] tokens, 10% of the words were replaced with random tokens from the vocabulary, and the rest were unchanged (Devlin et al., 2019). The model was trained on random sequences of 512 tokens from English Wikipedia and Bookcorpus (Devlin et al., 2019). The cross-entropy loss then was computed over the masked words (Devlin et al., 2019). The masking task aims to encourage the model to understand and learn contextual representations of words, as it needs to predict the masked words based on surrounding words within each sentence (Géron, 2022). Figure 4.2 depicts the final input after summing positional, segments, and token embeddings.

Next Sentence Prediction (NSP)

The goal of this training was to allow the model to understand the relationship between two sentences which was a limitation in traditional language models (Devlin et al., 2019). In order to prepare for the training and handle downstream tasks, the authors proposed a specific input representation process. First, the input sentences were tokenized by using Wordpiece tokenization developed by Wu et al. (2016). The tokenizer split the word into characters and then merged the characters back to form subwords based on vocabulary from the token dictionary. Additionally, classification tokens [CLS] and separation token [SEP] were added at the beginning of the sentence and at the end of the sentence, respectively (Devlin et al., 2019). Second, sentence embeddings were added to indicate whether each token was in sentence A or sentence B. Lastly, positional embeddings, as discussed in the previous section, were included. Figure 4.2 shows the input preparation process by summing the token, sentence, and positional embeddings. The data preparation allowed the model to distinguish each sentence during training. The training began with the model receiving pairs of sentences and learning to predict the subsequent sentences in each pair based on the original corpus (Devlin et al., 2019). Additionally, during training, 50% of second sentences were replaced with random sentences (Devlin et al., 2019).

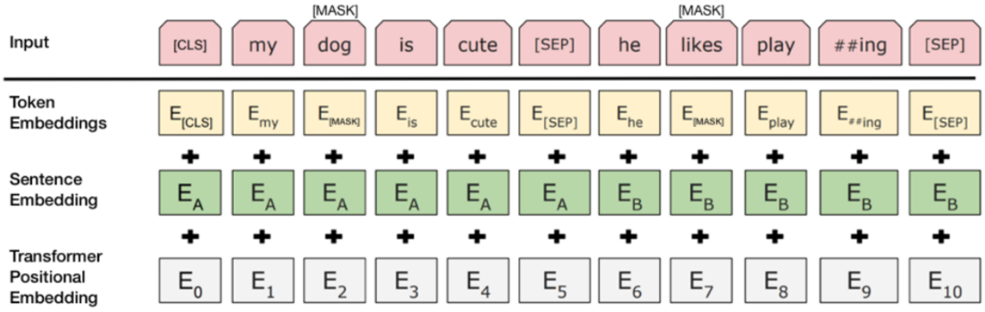

Input
[CLS]
my
[MASK]
dog
is
cute
[SEP]
he
[MASK]
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} $E_{[MASK]}$ E_{is} E_{cute} $E_{[SEP]}$ E_{he} $E_{[MASK]}$ E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Sentence Embedding
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Transformer Positional Embedding
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 4.2: The input preparation for BERT (Devlin et al., 2019)

4.2.2 Downstream Task

Data preparation for downstream task

In the first experiment, our goal was to train transformers to classify the rating based on users reviews. We tokenized the review text based on steps discussed in the NSP section. We truncated the sentences and set the max length to 128 due to efficiency. Moreover, we added special padding tokens to shorter sentences in the data to solve the unequal length of sentences which can create an issue for tensors. All of these processes were done by loading "bert-base-uncased" tokenizer from the transformer library.

After the data preparation for BERT, four final outputs were acquired: input_ids, token_type_ids, attention_mask, and labels. The Input_ids define the indices of each token. It added "101" and "102" to represent special tokens of [CLS] and [SEP] respectively. The toke_type_ids is used to differentiate a pair of sentences with the value 0 and 1. Attention_mask is a binary tensor which has a value of 0 and 1. It indicates the unpadded positions or 1 which the model should attend to. Label is a binary tensor of 0 and 1. It indicates a rating label for each review text.

Multi-label Text Classification

The pre-trained model which is "bert-base-uncased" was acquired through the Transformer library. The dropout layer and a linear layer were added on top of the final hidden state of [CLS] token for a sequence classification task provided in the library. As a result, the layer takes the output from the token and applies the linear transformation to get final logits for each class. We specified a learning rate of 2e-5, training and evaluating batch sizes of 16 and 32 respectively, and conducted training for 4 epochs, as recommended based on the paper (Devlin et al., 2019).

4.3 RoBERTa

Liu et al. (2019) used the BERT architectures and improved the pre-training process by removing next sentence prediction and implementing dynamic masking and a byte-level BPE tokenizer. The next sentence prediction (NSP) is ineffective since the training incorporates the masked language model (MLM) as a part of its training (Lan et al., 2019). As a result, NSP also benefits from the learning objective of MLM and makes the topic prediction aspect of NSP easier (Lan et al., 2019).

4.3.1 The pre-training

Dynamic Masking

Token masking in the original BERT implementation was performed once before the training (4.2). The input sequences were duplicated ten times to generate ten different positions for each sequence for 40 epochs during the training (Liu et al., 2019). As a result, the model saw the same four masking patterns from each input sequence during the training. This prevents the model from generalizing well to new masking positions since it becomes too familiar with the same masking patterns (Liu et al., 2019). On the other hand, Dynamic masking was implemented in the RoBERTa model to generate random masking patterns for each input sequence and helps to alleviate the problem.

Byte-Pair Encoding (BPE)

Byte-Pair Encoding is originally a data compression technique that breaks down words into subwords based on the most frequent pairs in the training corpus (Sennrich et al., 2015). The technique is widely used to handle large vocabularies in natural language processing. The byte-level implementation in BPE allows the tokenizer to encode every text input without introducing "unknown" tokens based on the vocabulary of 50,000 units and ensure comprehensive context without compromising the performance (Liu et al., 2019)

4.3.2 Downstream Task

Data preparation for downstream task

We followed a similar approach to BERT, except that we utilized the "roberta-base" tokenizer, which incorporates a byte-level Byte-Pair Encoding (BPE) tokenizer. Additionally, RoBERTa does not employ token_type_ids, which are used to separate sentences in BERT. Instead, we inserted "$< /s >$" in the input to separate the sentences. This adjustment reduces the complexity in data preparation while serving the same purpose. Therefore, the outputs after tokenization are input_ids, attention mask, and labels.

Multi-label Text Classification

We acquired "roberta-base" as our pretrained model and added sequence classification head on top of the pre-trained model. The head consists of three main layers: a dense layer, an activation function layer, and a output projection layer. The dense layer reduce the dimensionality of the input features. The activation function layer introduces the non-linearity to the output from the dense layer, and helps it learn the complex patterns of the data. The output projection layer is the linear transformation layer which generates the final logits for each class. We followed the same fine-tuning steps as in BERT.

4.4 XLNet

XLnet is a generalized auto-regressive pre-trained model. It aims to combine the benefits and overcome the limitations of the autoregressive (AR) model and autoencoding (AE). An autoregressive model aims to each word from the words before it by factorizing the product over the words where the probability of each word is dependent on the words before it (Yang et al., 2019). However, the model can only estimate the probability distribution of sentences in either forward or backward directions within the sentence. As a result, the model is unable to capture the contextual information from all positions. On the other hand, autoencoding or BERT aims to predict the masked tokens by training on the corrupted version of input sequences. The absence of the special mask tokens during the finetuning leads to the discrepancy between a pre-train and finetune. Moreover, BERT assumes that the masked tokens are independent from other given unmasked tokens which is uncommon in the natural language.

XLNet exploits all permutations of the factorization order and autoregressive formulations to address the limitations in AR and AE methods. As a result, it can capture the dependencies between words more effectively and capture bidirectional context without relying on data corruption.

4.4.1 Permutation Language Objective

The training objective decomposes the log probability of the next word into the sum of log probabilities over each word in the sequence, considering all possible permutations of input sequences. As a result, the model can capture contextual information based on different positions in the sequence. The proposed permutation objective is as follows:

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim Z_T}\left[\sum_{t=1}^{T} \log p_\theta(x_{z_t}|\mathbf{x}_{\mathbf{z}<t})\right] \tag{4.4}$$

Where T is the length of the input, z is a permutation of indices in $[1, T]$, x_t is the t-th token in the sequence, x_{z_t} is the token at position $z(t)$ in the permutation, $x_{z<t}$ denotes the tokens before position t in the chosen permutation $\mathbf{z}$. $\mathbf{z} \sim Z_T$ is the summation of all permutations. It maximizes the log probability of the next word or x_{z_t} given the preceding $x_{z<t}$.

4.4.2 The model architecture

The problem with the permutation language objective function in XLNet is that the model cannot capture the dependencies between the target word and its specific position effectively. In the standard parameterization, the model predicts the same probabilities for the next possible word, regardless of the target positions in different permutations (Yang et al., 2019). Consequently, XLNet introduces the two-stream self-attention mechanism by incorporating the content stream and the query stream attention into the hidden states of the model to address this issue (Yang et al., 2019).

Two-Stream Self-Attention

There are two requirements that we need to take into consideration. First, the query representation $g(x_{z<t}, z_t)$ should predict the token x_{z_t} based on the position z_t and not the content x_{z_t}(Yang et al., 2019). Second, when predict the other token x_{z_i}, $g(x_{z<t}, z_t)$ should encode x_{z_t} to provide full contextual information and take other the relevant contextual information into account (Yang et al., 2019). The two types of representations in the model are as followed:

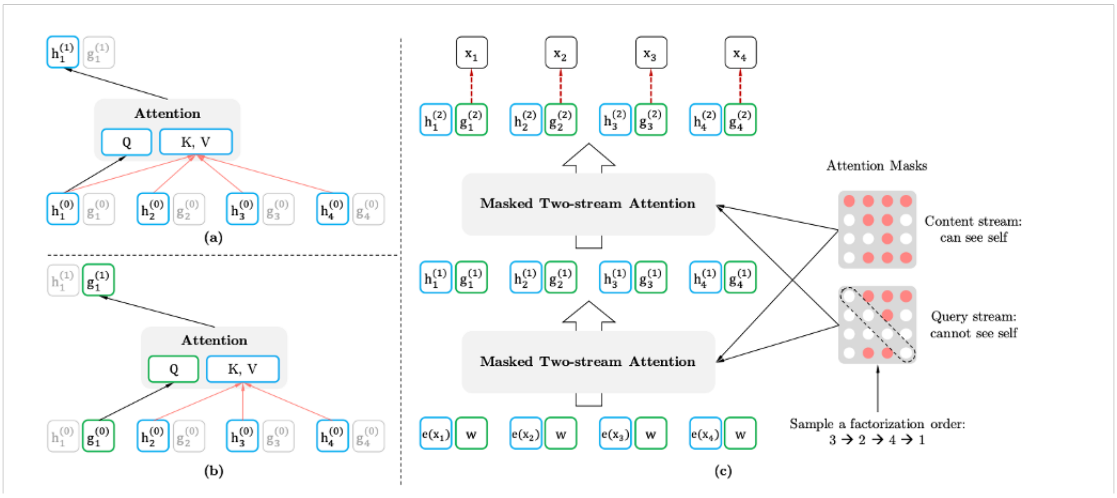
1. The content representation or $h_\theta(x_{z\leq t})$: It encodes both the context or the tokens before position t $(x_{z<t})$ and the token at position z_t (x_{z_t}) (Yang et al., 2019).

2. The query representation or $g_\theta(x_{z<t}, z_t)$: it has access to the context or the tokens before the position t $(x_{z<t})$ and the target position of x_{z_t}(Yang et al., 2019).

In each attention layer, both h_{z_t} and g_{z_t} are updated schematically and we use the last layer of query representation g_{z_t} to compute the probability of the next word. The re-parameterize objective function which focus on target position aware is as follows:

$$p_\theta(X_{z_t} = x|x_{z<t}) = \frac{exp(e(x)^T g_\theta(x_{z<t}, z_t)}{\sum_{x'} exp(e(x')^T g_\theta(x_{z<t}, z_t))} \tag{4.5}$$

In the given Figure 4.3 with a factorization order of 3, 2, 4, 1, in order to predict the content representation of x_1, we need h_1 for our query and consider h_1 to h_4 for our key and value from previous hidden layer since x_1 is the last position in input sequence. On the other hand, figure (b) shows that we cannot see the content of x_1 or $h_1^{(0)}$ when predicting the query representation $g_1^{(1)}$. Therefore, we need g_1, h_2, h_3,and h_4 from previous hidden layer to predict the next word. Figure (c) depicts the overview of the permutation language training with two-stream attention. In the first layer, the model initialize a trainable vector as w for query representation $g_i^{(0)}$ and word embedding as $e(x_i)$ for content representation $h_i^{(0)}$ (Yang et al., 2019).



Attention
Q
K, V
(a)
(b)
x₁
x₂
x₃
x₄
Masked Two-stream Attention
Masked Two-stream Attention
Attention Masks
Content stream: can see self
Query stream: cannot see self
e(x₁)
e(x₂)
e(x₃)
e(x₄)
w
Sample a factorization order: 3 → 2 → 4 → 1
(c)

Figure 4.3: Content Stream Attention (a), Query Stream Attention (b), and Masked Two-Stream Attention (c) (Yang et al., 2019)

Partial prediction

The concept of partial prediction is introduced to address the optimization problem of permutation training objective (Yang et al., 2019). Instead of predicting all tokens in the sequence, the algorithm predict only the last tokens in the factorization order (Yang et al., 2019). The finalized permutation objective is as follows:

$$\max_{\theta} \quad \mathbb{E}_{\mathbf{z}\sim Z_T}[\log p_\theta(\mathbf{x}_{\mathbf{z}>c}|\mathbf{x}_{\mathbf{z}\leq c})] = \mathbb{E}_{\mathbf{z}\sim Z_T}\left[\sum_{t=c+1}^{|z|} \log p_\theta(x_{z_t}|\mathbf{x}_{\mathbf{z}<t})\right] \tag{4.6}$$

Where z is the factorization order and split by the cutting point c. $z > c$ denotes to target subsequence and $z \leq c$ denotes to non-target subsequence. We maximize the log likelihood of predicting the target subsequence based on the context from non-target subsequence.

4.4.3 Downstream Task

Data preparation

Similar to BERT, we need to tokenize our review data before downstream tasks in XLNet. We added special tokens [CLS] and [SEP] at the beginning and ending of the sentence as recommended from the paper (Yang et al., 2019). However, the token pattern is different since XLNet does not rely on the Next Sentence Prediction (NSP) task like BERT. The token pattern is as follows: sentence A + [SEP] + sentence B +[SEP] +[CLS]. We truncated the sentences, set max length to 128, and added special padding tokens to solve the unequal length of sentences. We tokenized our review data by importing "xlnet-base-cased" tokenizer from transformers library.

Multi-label text classification

The pre-trained model used in our experiment is "xlnet-base-cased" which was acquired through the Transformers library. We added the linear layer on top of the final hidden state for the sequence classification task. To ensure a fair comparison, we specified the same learning rate, training and evaluation batch sizes, and number of epochs to be the same as those used for BERT.

4.5 Sentence Embeddings

In the second experiment, we investigated if sentence embeddings can improve the recommender system as a side information when there are limited user-item interactions in the system. The idea is that rating is unable to represent user preferences entirely, and users have a different standard. Moreover, we are aware that previous literature has not fully exploited the semantic textual similarity of user reviews, and we realized that it can be more informative than sentiment analysis and topic modeling.

However ,The BERT architecture is inefficient to solve semantic similarity search since it uses a cross encoder for pairs of sentences. Our reviews have almost 10,000 sentences, and it can take almost 65 hours to compute the task (Reimers & Gurevych, 2019). To solve the issue, we put the mean pooling layer on top of the BERT model to get a fixed 768 dimensional output from each review. Consequently, we normalized the embeddings and computed cosine similarity between each user review to find the similarity between each user based on their reviews.

Our approach for obtaining sentence embeddings is based on the paper by Reimers and Gurevych (2019). The authors proposed adding a pooling operation layer on top of the model to aggregate the contextualized word embeddings into a single vector representation and obtain fix-length sentence embeddings. By applying a pooling layer, we obtained meaningful compact representations for the entire sentences. Moreover, we used the same method to obtain the sentence embeddings from RoBERTa and XLNet as suggested by the authors.

4.6 Collective Matrix Factorization

In order to address the second and third sub-research questions, the recommender system should meet three specific requirements. Firstly, the system should be able to utilize both sentence embeddings information and ratings to generate recommendations. Secondly, it should be a fast and effective method to generate recommendations from high dimensional data. Third, it should provides a framework that allows us to compare different type of word embeddings. Consequently, the collective matrix factorization technique developed by Singh and Gordon (2008) fits well with our scenario. The hybrid collaborative technique extended the low-rank factorization model and incorporate user or item attributes. The technique factorizes the user-item interaction matrix and user and item attributes as additional matrices and jointly factorize them (Singh & Gordon, 2008). As a result, the model can learn to capture the latent features and capture underlying patterns between users, items, and users preferences (Cortes, 2018). Since more than 50% of item descriptions in our data set are missing, we decided to experiment with only user attributes from user reviews.

The objective function is to find the factorized matrices A, B, C, and D that minimize the loss function between the actual observed values in the interaction matrix and the predicted ones from the factorized matrices. As a result, the process is able to make precise predictions for unknown entries, capture underlying patterns in the data, improve the effectiveness of recommendations (Cortes, 2018). The collective matrix factorization is as follows:

$$\min_{A,B,C,D} \|I_x(X - AB^T)^2\| + \|U - AC^T\|^2 + \|I - BD^T\|^2 \tag{4.7}$$

Where I_x is the indicator function of a partially observed matrix, X is the user-item interactions or product ratings, A is user factor matrix, B is item factor matrix, U is user attributes matrix, and C is user attributes factor matrix.

4.7 The Proposed Method and Evaluation Metrics

We developed the method to alleviate cold start problem by leveraging transformers to classify the missing rating and encode the review into contextual sentence embeddings as a side information for the recommender system. The method should be effective when there is missing rating data and limited user reviews. We incorporated both rating and embeddings into collective matrix factorization to get the recommendations for new users. We conducted two experiments to evaluate the effectiveness of the proposed method. Figure 4.4 depicts the visualization of our proposed method.

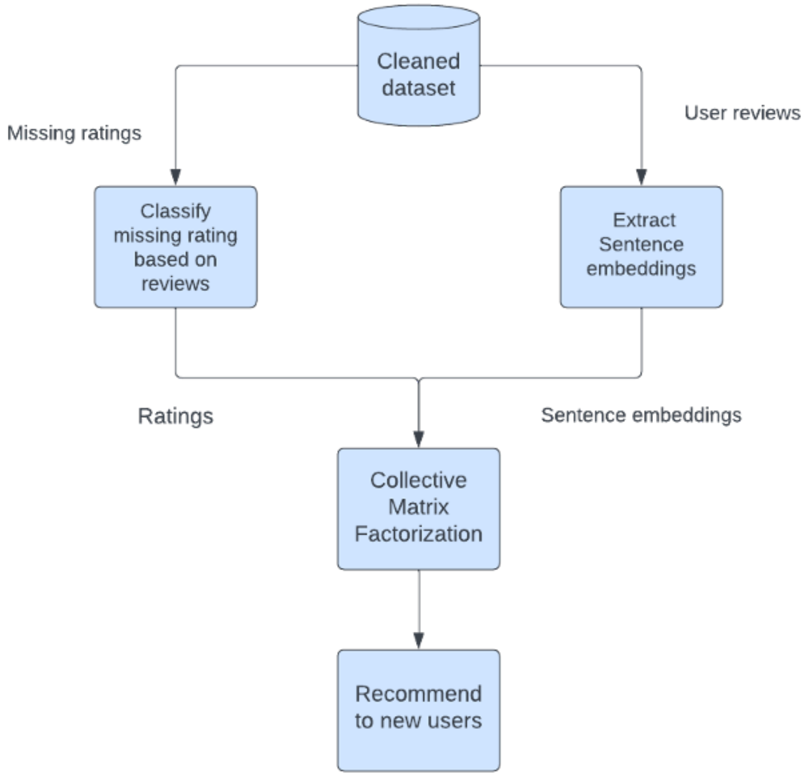

Cleaned dataset
Missing ratings
User reviews
Classify missing rating based on reviews
Extract Sentence embeddings
Ratings
Sentence embeddings
Collective Matrix Factorization
Recommend to new users

Figure 4.4: the System Architecture of the Proposed Recommender System

4.7.1 Multi-label Classification

To answer the first research question, we conduct the first experiment by comparing BERT, RoBERTa, and XLNET on multi-label classification on missing rating values in the user reviews. We evaluated the result by using accuracy, weighted F1, and Area Under ROC Curve.

Accuracy

The metric gives the overall accuracy of the model. The metrics is defined as followed:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{4.8}$$

Weighted F1

For multiclass classification, we used One-vs-Rest or OvR to compare each class to the rest of the classes. Therefore, we have F1 score for each class separately. Consequently, we calculate weighted F1 by multiplying each class by the weight or the proportion of the actual occurrence for each class in the dataset. Then, we summed all five values from five different classes to get the final weighted F1. The metric is defined as followed:

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{4.9}$$

$$weightedF1 = \frac{\sum(w_i F1_i)}{\sum w_i} \tag{4.10}$$

Area under the curve for the Receiver Operating Characteristic (AUC-ROC)

The metric is calculated by plotting the true positive rate against the false positive rate across different threshold. If AUC-ROC is more than 0.5, it indicates that the model classify better than chance.

4.7.2 Recommender system

To answer the second and third question, we incorporated sentence embeddings from three transformer models and rating results from the best model into collective matrix factorization. Since we ensured that there was no overlap between the users in the test and training datasets, the recommendation on new users were based solely on the side information. We evaluated the accuracy of recommendations by using three widely used rank metrics for recommendations which are precision and recall at N, and mean average precision (MAP) (Feng et al., 2021). The higher of these metrics, the better performance of the recommendations at ranking relevant items. We compared results from three transformers with Alternating Least Square (ALS) which is a classic matrix factorization as our base line model.

Precision and recall at N

Precision at N evaluates if the top N recommended items are relevant. For example, if our precision at top 3 is 75%, it means that 75% of the list are relevant to the user. On the other hand, recall focus on the relevant items instead. If recall at 3 is 30%, it indicates that only 30% of the all relevant items are found in the top 3 results. The metrics is define as followed:

$$Precision@N = \frac{\sum_{u \in U^T} |I_u^N \cap I_u^T|}{N|U^T|} \tag{4.11}$$

$$Recall@N = \frac{1}{|U^T|} \sum_{u \in U^T} \frac{|I_u^N \cap I_u^T|}{I_u^T} \tag{4.12}$$

Where I_u^N is the recommendation list of items from the recommendation system,I_u^T is the ranking list of actual items users bought, $U^T$is the set of users in the dataset, and N equal to $|I_u^N|$

Mean average precision (mAP)

The metric considers both precision and average precision at each position in the ranked list. As a result, mAP is more sensitive to the position of the item that the user is interested in. It is defined as followed:

$$mAP = \frac{1}{|U^T|} \sum_{u \in U^T} \sum_{N=1}^{|I_u^T|} \frac{1}{N} \sum_{k=1}^{N} \delta(i_u^k \in I_u^T) \tag{4.13}$$

Where $\delta(i_u^k \in I_u^T)$ is an indicator function. When $i_u^k \in I_u^T$ is true, $\delta(i_u^k \in I_u^T) = 1$. i_u^k is the actual items ranking at k position.

Chapter 5

Results

This chapter presents key relevant findings derived from two experiments prior to the discussion of the interpretations. In the first experiment, we examined the learning curve of the models based on training loss and validation loss from multi-label text classification of user ratings and evaluated three model performances. In the second experiment, we evaluated the effectiveness of the proposed recommender method in cold start scenarios with baseline models.

5.1 Multi-label text classification

5.1.1 The Model Learning Curves

We initially set the number of epochs to 4 at the beginning of the training process, following the recommendation in the paper. However, we monitored the behavior of each model by analyzing the training loss and validation loss. Eventually, we decided to select the models at epoch 2. Figure 5.1, 5.2, and 5.3 provides valuable insights as they show that the training losses were consistently higher than the validation losses and gradually decreased with each epoch, indicating the models were learning. However, for all three models, the validation loss started to increase after the second epoch. This increase indicated that the models were overfitting to the training data and struggling to generalize effectively to the validation data set beyond 2 epochs.

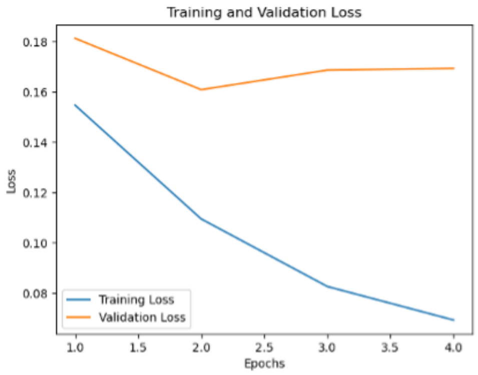


Figure 5.1: BERT model learning curves

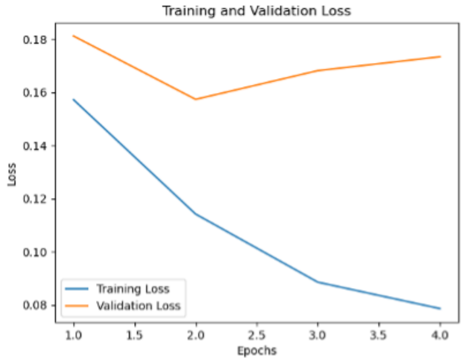


Figure 5.2: RoBERTa model learning curves

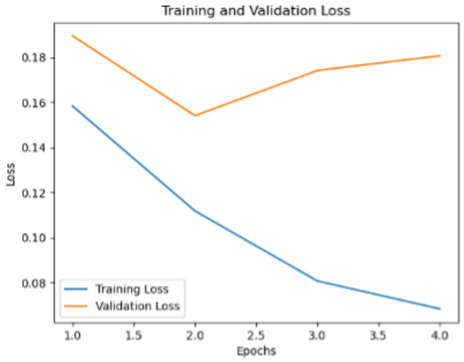


Figure 5.3: XLNet model learning curves

5.1.2 The Model Performance

Table 5.1 demonstrates the performances from three models on a test data set based on accuracy, weighted F1, and Area under the curve for the Receiver Operating Characteristic (AUC-ROC). It is important to note that we evaluated based on the missing 30% rating rather than the entire rating column. There was not much difference between BERT and RoBERTa performances across three metrics. BERT yielded an accuracy of 84% for accuracy which was slightly higher than the accuracy of RoBERTa at 83.6%. Both models achieved a weighted F1 of 0.830. However, BERT obtained a slightly higher AUC-ROC score at 0.746 compared to RoBERTa's score of 0.746. On the other hand, XLNet outperformed both models across the three metrics. It achieved an accuracy of 84.9%, which is 1.1% and 1.6% higher than BERT and RoBERTa, respectively. Moreover, it obtained a weighted F1 score of 0.840, which is 1.2% higher than BERT and RoBERTa. In terms of AUC-ROC scores, it achieved 0.750, which was slightly higher than BERT and RoBERTa by 0.54% and 0.40%, respectively.

Model	Accuracy	Weighted F1	AUC-ROC
BERT	0.840	0.830	0.747
RoBERTa	0.836	0.830	0.746
XLNet	**0.849**	**0.840**	**0.750**

Table 5.1: Model performance on 5-label rating classification

5.2 Recommender system

In the second experiment, we evaluated the performance of our proposed method in comparison to the baseline models which are Collective Matrix Factorization with BERT, RoBERTa, and XLNet sentence embeddings, as well as Alternating Least Squares (ALS) with and without missing values datasets. The complete rating results from XLNet were used, and the data was split into non-overlapping users into training and test sets to simulate cold start scenarios for users. The detailed results are presented in Table 5.2.

Table 5.2 demonstrates that ALS with the original dataset with missing values performed poorly in making recommendations compared to other models. The precision at 3 and 5 were 28.9% and 26.1%, respectively. Moreover, the recall at 3 and 5 was 26%. The mean average precision at 3 and 5 were 28.9% and 26.1%, respectively. However, we also investigated the performance of the ALS algorithm using a complete rating dataset (ALS_XL) obtained from the best model in the first experiment. The inclusion aimed to determine if sentence embeddings improve the recommendation performance or are a crucial component in our proposed method, as discussed in section 4.7. ALS generated slightly worse results than other transformer-based methods. To illustrate, It achieved 88.4% accuracy when recommending the top 3 items to users. However, the accuracy of the recommendations declined to 87.6% when recommending the top 5 items to users. The algorithm provided 78.2% for recall at rank 3 and increased to 87% for recall at rank 5. In terms of mean average precision (mAP), ALS achieved 86.9% and 85.9% at ranks 3 and 5, respectively.

The BERT-based system provided slightly better overall results than ALS based in Table 2. The precision at rank 3 and rank 5 reached 88.5% and 87.8%, surpassing ALS at 0.23%. The recall at rank 3 was 78.4% and increased to 87.2% for rank 5. The recall values were 0.26% higher at rank 3 and 0.35% higher at rank 5 compared to ALS. Furthermore, the mean average precision at rank 3 and rank 5 were 87.2% and 86.2% which were 0.35% and 0.26% higher than ALS values. The mean average precision (mAP) values highlighted the effectiveness of the proposed method in consistently providing relevance across different recommendation lists.

Compared with the BERT-based system, the RoBERTa-based system provided a slight improvement in precision at rank 3. The precision at rank 3 was 88.7%, which is 0.23% higher than the system with BERT sentence embeddings and 0.45% higher than ALS. This indicates that the system with RoBERTa generates more accurate recommendations within the top 3 items. However, in terms of recall and mean average precision (mAP), it achieved similar results to BERT sentence embeddings.

The system with XLNet sentence embeddings achieved the best result among all the models. When compared with ALS with the original dataset with 30% missing rating values, the XLNet-based system obtained significant improvements. It achieved 207% higher for precision at 3, 237% higher for precision at 5, 202% higher for recall at 3, 237% higher for recall at 5, 202% higher for mean average precision at 3, and 230% higher for mean average precision at 5. When compared with ALS without missing rating (ALS_XL), the XLNet-based system achieved slight improvements. It obtained 0.45% higher for precision at 3, 0.46% higher for precision at 5, 0.26% higher for recall at 3, 0.23% higher for recall at 5, 0.35% higher for mean average precision at 3, 0.26% higher for mean average precision at 5. The system with XLNet sentence embeddings showed an improvement in precision at rank 3 and 5 compared to RoBERTa and BERT. Its precision values at rank 3 and 5 were 88.7% and 88%, which were 0.11% and 0.23% higher than the precision values of RoBERTa. Moreover, its precision values at rank 3 and 5 also were higher than BERT's for 0.23% for both ranks. The results indicate that the system with XLNet sentence embeddings achieves superior precision compared to other transformer models. However, it provided similar performances in recall and mean average precision (mAP) to other transformers.

Method	Prec@3	Prec@5	Recall@3	Recall@5	MAP@3	MAP@5
ALS	0.289	0.261	0.260	0.259	0.289	0.261
ALS_XL	0.883	0.876	0.782	0.870	0.869	0.859
BERT based	0.885	0.878	**0.784**	**0.872**	**0.872**	**0.862**
RoBERTa based	0.886	0.878	**0.784**	**0.872**	**0.872**	**0.862**
XLNet based	**0.887**	**0.880**	**0.784**	**0.872**	**0.872**	**0.862**

Table 5.2: Model performance on recommendation

Chapter 6

Discussion

6.1 Interpretation of the results

Our findings offer valuable insights into each of these sub-questions and collectively contribute to a comprehensive understanding of the effectiveness of transformers in recommender systems under user cold-start settings.

From our first experiment, we made two main discoveries. Firstly, the performance of RoBERTa and BERT was comparable, showing minimal differences. Secondly, XLNet outperformed BERT and RoBERTa in the multi-label classification task. In the multi-label text classification task, we expected that RoBERTa will outperform BERT based on the paper by Liu et al. (2019). However, the performance of RoBERTa was similar to that of BERT based on accuracy, weighted F1, and AUC-ROC metrics. It seems that the removal of NSP (Next Sentence Prediction) and the implementation of dynamic masking during the pre-training process may not have had a substantial impact on multi-label text classification. Furthermore, it is important to note that Liu et al. (2019) primarily evaluated RoBERTa on tasks related to semantic similarity and question answering, which may explain the limited benefits observed in our specific task of multilabel text classification. On the other hand, XLNet outperformed two models by no more than 1.6% across three metrics. It demonstrates that the XLNet architecture may be more effective in this specific dataset and the task. However, XLNet was trained on more diverse and larger datasets than BERT and RoBERTa during pre-training (Yang et al., 2019). Therefore, we cannot conclude that XLNet's architecture is superior in this specific fine-tuning task.

In our second experiment, we identified three key findings. Firstly, incorporating complete ratings improved the performance of the traditional recommender system significantly (ALS and ALS_XL). Secondly, systems utilizing transformers outperformed baseline models, showcasing the overall enhancement achieved through transformer-based approaches. Lastly, while there were minor variations, the results from different transformers were relatively similar, with XLNet demonstrating a slight advantage. In the recommender system experiment, Alternating Least Square (ALS) with the original data set provided the worst results as expected. This incomplete rating information introduces data sparsity problems in the recommender system. As a result, the algorithm cannot capture the complete user-item interactions and lead to unreliable factorization results (Jannach et al., 2010). Moreover, it is even more challenging for the algorithm to predict the rating for unknown users in the test dataset. On the other hand, we anticipated that the ALS technique with the complete rating dataset (ALS_XL) would also provide significantly inferior results compared to our proposed method. In theory, matrix factorization struggles to generate accurate recommendations in cold start settings (Jannach et al., 2010; Roy & Dutta, 2022). Surprisingly, it was able to generate compatible performances in terms of precision, recall, and mean Average Precision (mAP). One of the possible explanations is that ALS generates recommendations for new users by identifying overlapping items from similar users in the training data sets based on the latent representations between existing users and new users. However, our proposed models still generated better results overall.

Another important point is that the performances of the three transformers are similar across all six metrics, with the XLNet-based system performing slightly better in precision. One possible explanation for this similarity in performance is the positive bias and limited information present in the reviews. Based on figure 3.1, the reviews in the data set mostly contain positive feedback (more than 60% rated 5 for items). Three transformers may capture similar overall sentiment information and result in similar overall sentence embeddings and lead to similar recommender performance. Moreover, figure 3.3 shows that the reviews predominantly contain limited set of words such as ”love,” ”skin,” and ”hair” for positive feedback, and ”money,” ”shave,” ”bad,” and ”waste” for negative feedback. As a result, these reviews may not sufficiently represent the unique preferences of each customer and have a narrow set of topics. As a result, this limits the ability of the models to capture substantial variations between users. However, the XLNet-based system still demonstrated better performances in handling cold-start scenarios and providing recommendations than the ALS technique. Our proposed method effectively addressed the challenges of data sparsity and cold start problems, outperforming the traditional ALS method.

6.2 Implications

6.2.1 Academic implications

First, we contribute by exploiting both ratings and reviews in our study. The existing literature mainly focused on predicting ratings of new customers based on purchase history and sentiment analysis techniques (Feng et al., 2021; Cumbreras et al., 2013; Osman et al., 2021; C. N. Dang et al., 2021). However, solely focusing on rating prediction can lead to inaccuracy recommendations since ratings are subjective and biased measures (Hu et al., 2009; Muchnik et al., 2013). Additionally, sentiment analysis oversimplifies user opinions and preferences, failing to capture the details present in user reviews. In contrast, our study addresses these limitations by incorporating both rating and user reviews in the recommender system, enabling a more comprehensive understanding of user opinions and specific preferences.

Second, our study contributes to the current literature by proposing a Transformer-based hybrid approach to address the data sparsity and cold start problem in the recommender system. Previous studies have explored traditional word embeddings, such as Word2Vec and LightFM embeddings (Kula, 2015; Nguyen et al., 2020). In accordance with the studies by Kula (2015) and Nguyen et al. (2020), embeddings based on user information improve the recommender system in cold start conditions. However, The studies focus on individual words rather than capturing the underlying meaning within user reviews. We acknowledge this limitation and emphasize the use of sentence embeddings to capture the user preferences and enhance the hybrid filtering system. Moreover, Our proposed method has not been previously explored in the literature, and it offers a promising solution to effectively handle new users and mitigate the cold start problem.

Third, contrary to the study by Liu et al. (2019), our experiments showed that RoBERTa did not outperform BERT, especially in the multi-label text classification task. It may be more efficient to use the BERT architecture since RoBERTa has been trained on more data and longer sequences while still providing similar performance in this context. This finding challenges the prevailing assumption and highlights the need for future research on the factors that impact the performance of transformer models in different scenarios.

In conclusion, our study contributes to the recommender field by proposing a novel hybrid recommender system with the help of transformers to handle both data sparsity and cold start problems. We stand apart from current literature by addressing its limitations and present a method that leverages the semantic context in user reviews and incorporates both ratings and sentence embeddings, improving recommendations.

6.2.2 Managerial implications

Based on the results from two experiments, our proposed method can effectively provide recommendations while addressing both data sparsity (limited information on ratings) and user cold start problems. However, the fine-tuning process in downstream tasks of transformers requires time (4-5 hours with a max length of 128 for each model) and computational resources. As a result, the proposed method is not recommended for a fast-paced environment like a fast fashion business that may require us to fine-tune the model with large data on a daily basis. Our proposed method is recommended for high-end luxury brand companies since they have exclusive, fewer customers and lower purchase frequency. The study can benefit the brands by improving customer loyalty and personalized experiences, which are significant selling points for luxury brand businesses (Gupta et al., 2023; Manthiou et al., 2020), through improved recommendation systems. Consequently, By improving the quality of recommendations, businesses can anticipate a notable boost in sales and revenues and reduce search costs (Hinz & Eckert, 2010). We recommend two strategies based on our findings. First, managers should fine-tune XLNet on their review datasets and extract sentence embeddings by also using XLNet since this model demonstrated the best performance among the three models. Second, managers should always encourage customers to leave detailed feedback since our method relies heavily on user feedback in order to have accurate recommendations. It is important to note that our method can perform well in the absence of rating information unlike other traditional methods such as ALS, but there must be at least one review from each customer in the system. However, these are general recommendations, and the brands should also consider their brand identities and strategic goals.

6.3 Limitations and future research

There are four main limitations in our study. First, we evaluated based on a small sample of 10,000 observations due to the computational limitations and the long training time. In the data preprocessing step, there were 324,033 rows left after filtering out users with more than 20 interactions and removing duplicate rows. Consequently, it took us more than 10 hours to fine-tune each model. As a result, we decided to sample the data for 10,000 rows based on rating, which allows us to train each model within five hours. However, a larger data set can improve the learning curves and mitigate overfitting of Transformers. Second, we evaluated our proposed method with one type of dataset. It would improve the certainty and generalizability of our performance if we included multiple datasets. Third, our study compares the three specific transformers (BERT, RoBERTa, and XLNet). As a result, it would improve the study to be more comprehensive to compare them with other text transformers, such as Electra and Albert. Fourth, we cannot conclude that XLNet is the best model based on the results. This is because the model was trained on more than ten times larger and more diverse data sets (Yang et al., 2019). XLNet should be built from scratch and trained on the same datasets as BERT for a fair comparison. However, due to time and resource constraints, we were unable to improve this. Nevertheless, our proposed method sheds light on the benefits of combining transformers and hybrid recommender systems in cold start settings, as well as their limitations.

Future research should take four main areas into account. First, it is recommended to reproduce this research on multiple and larger data sets. Second, future research can investigate other hybrid approaches that can incorporate both rating and sentence embeddings from transformers. For example, future studies can look into Factorize Machine-based neural networks (Guo et al., 2017), Neural Collaborative Filtering (He et al., 2017), and Deep Cooperative Neural Networks (Zheng et al., 2017). Third, we advised future studies to explore ways to improve the efficiency of the fine-tuning process. It would be beneficial for future research to investigate other techniques to reduce training time and computational resources. If this issue is mitigated, we can use the models for real-time recommendations with large data sets daily. Finally, future studies can include item descriptions and customer demographic information to improve the performance of the proposed method and capture more diverse preferences.

Chapter 7

Conclusion

This research aims to investigate the effectiveness of transformers models in handling data sparsity and user cold start problems in recommender systems. Based on the findings from the first experiment, we can conclude that XLNet outperforms other models at multi-label text classification based on user reviews when there are missing ratings. Moreover, based on the insights from the second experiment, incorporating sentence embeddings of user reviews as user-side information improves the recommender system in cold start settings. The result shows that our transformer-based models (BERT, RoBERTa, XLNet) outperform traditional ALS baseline models. To answer the last research question, we can conclude that three-sentence embedding from three transformers are similar, with XLNet showing a slightly better performance. The reasons behind the similar performances are the positive bias and limited information from the reviews. As more than 60% of users in the data rate 5 for the items, it indicates a positive bias toward the models toward higher ratings. Moreover, the reviews contain a limited range of topics since they consist of mainly 3-4 frequently occurring words. Therefore, three models can capture the same variations and overall sentiment from each review due to the positive bias and the lack of diversity in topics. The study not only shows the benefits of transformers in handling data sparsity and user cold start problems, but it also raises the question of the efficiency of the transformers in fine-tuning tasks. Three models were fine-tuned for nearly 15 hours using an 8-core Graphics Processing Unit (GPU) to get the results in the first experiment, emphasizing the time-consuming nature of transformers. Based on these conclusions, researchers can consider other fine-tuning strategies to improve efficiency and experiment with larger and more diverse data sets. Nevertheless, our study contributes to the field of recommendation systems by addressing data sparsity and cold start problems with Transformers-based hybrid systems. Our findings not only confirm Transformers' effectiveness under cold start conditions in recommender systems but also highlight the need for further study and improvement in the fine-tuning process.

References

Abdi, H. (2006). Singular value decomposition (svd) and generalized singular value decomposition (gsvd)..

Alamdari, P. M., Navimipour, N. J., Hosseinzadeh, M., Safaei, A. A., & Darwesh, A. M. (2020). A systematic study on the recommender systems in the e-commerce. *IEEE Access*, *8*, 115694-115716.

Ayata, D., Yaslan, Y., & Kamasak, M. E. (2018). Emotion based music recommendation system using wearable physiological sensors. *IEEE Transactions on Consumer Electronics*, *64*, 196-203.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, *abs/1409.0473*.

Balabanovic, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Commun. ACM*, *40*, 66-72.

Basiri, J., Shakery, A., Moshiri, B., & Hayat, M. Z. (2010). Alleviating the cold-start problem of recommender systems using a new hybrid approach. *2010 5th International Symposium on Telecommunications*, 962-967.

Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, *10*, 147-180.

Blake, M. B. (2017). Two decades of recommender systems at amazon..

Breese, J. S., Heckerman, D. E., & Kadie, C. M. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Conference on uncertainty in artificial intelligence.*

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, *25*, 99-154.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Annual international acm sigir conference on research and development in information retrieval.*

Cortes, D. (2018). Cold-start recommendations in collective matrix factorization. *ArXiv*, *abs/1809.00366*.

Crespo, R. G., Martínez, O. S., Lovelle, J. M. C., García-Bustelo, B. C. P., Gayo, J. E. L., & de Pablos, P. O. (2011). Recommendation system based on user interaction data applied to intelligent electronic books. *Comput. Hum. Behav.*, *27*, 1445-1449.

Cumbreras, M. Á. G., Ráez, A. M., & Díaz-Galiano, M. C. (2013). Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Syst. Appl.*, *40*, 6758-6765.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., & Salakhutdinov, R. (2019).

Transformer-xl: Attentive language models beyond a fixed-length context. *ArXiv*, *abs/1901.02860*.

Dang, C. N., García, M. N. M., & de la Prieta, F. (2021). An approach to integrating sentiment analysis into recommender systems. *Sensors (Basel, Switzerland)*, *21*.

Dang, E., Hu, Z., & Li, T. (2022). Enhancing collaborative filtering recommender with prompt-based sentiment analysis. *ArXiv*, *abs/2207.12883*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, *abs/1810.04805*.

du Boucher-Ryan, P., & Bridge, D. G. (2005). Collaborative recommending using formal concept analysis. In *Knowledge-based systems.*

Feng, J., Xia, Z., Feng, X., & Peng, J. (2021). Rbpr: A hybrid model for the new user cold start problem in recommender systems. *Knowl. Based Syst.*, *214*, 106732.

Gabrilovich, E., & Markovitch, S. (2014). Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Intell. Res.*, *34*, 443-498.

Géron, A. (2022). *Hands-on machine learning with scikit-learn, keras, and tensorflow.* " O'Reilly Media, Inc.".

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, *4*, 133–151.

Gope, J., & Jain, S. K. (2017). A survey on solving cold start problem in recommender systems. In *2017 international conference on computing, communication and automation (iccca)* (pp. 133–138).

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*.

Gupta, D. G., Shin, H., & Jain, V. (2023). Luxury experience and consumer behavior: a literature review. *Marketing Intelligence & Planning*, *41*(2), 199–213.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. *Proceedings of the 26th International Conference on World Wide Web*.

Hinz, O., & Eckert, J. (2010). The impact of search and recommendation systems on sales in electronic commerce. *Business & Information Systems Engineering*, *2*, 67–77.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 89–115.

Hu, N., Zhang, J., & Pavlou, P. A. (2009). Overcoming the j-shaped distribution of product reviews. *Communications of the ACM*, *52*(10), 144–147.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, *16*(3), 261–273.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems - an introduction..

Kardan, A. A., & Ebrahimi, M. (2013). A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Information Sciences*, *219*, 93–110.

Krestel, R., Fankhauser, P., & Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In *Proceedings of the third acm conference on recommender systems* (pp. 61–68).

Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. *arXiv preprint arXiv:1507.08439*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lee, D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, *13*.

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Syst. Appl.*, *41*, 2065-2073.

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, *7*(1), 76–80.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lops, P. (2014). Semantics-aware content-based recommender systems. In *Recommender systems handbook.*

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Manthiou, A., et al. (2020). Applying the eee customer mindset in luxury: reevaluating customer experience research and practice during and after corona. *Journal of Service Management*, *31*(6), 1175–1183.

Melville, P., Mooney, R. J., Nagarajan, R., et al. (2002). Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, *23*, 187–192.

Muchnik, L., Aral, S., & Taylor, S. J. (2013). Social influence bias: A randomized experiment. *Science*, *341*(6146), 647–651.

Nguyen, L. V., Nguyen, T.-H., & Jung, J. J. (2020). Content-based collaborative filtering using word embedding: a case study on movie recommendation. In *Proceedings of the international conference on research in adaptive and convergent systems* (pp. 96–100).

Osman, N. A., Noah, S. A. M., Darwich, M., & Mohd, M. (2021). Integrating contextual sentiment analysis in collaborative recommender systems. *PLoS ONE*, *16*.

Penha, G., & Hauff, C. (2020). What does bert know about books, movies and music? probing bert for conversational recommendation. In *Proceedings of the 14th acm conference on recommender systems* (pp. 388–397).

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., & Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on intelligent user interfaces* (pp. 127–134).

Rashid, A. M., Karypis, G., & Riedl, J. (2008). Learning preferences of new users in recommender systems: an information theoretic approach. *Acm Sigkdd Explorations Newsletter*, *10*(2), 90–100.

Recht, B., Re, C., Wright, S., & Niu, F. (2011). Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *Advances in neural information processing systems*, *24*.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rennie, J. D., & Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on machine learning* (pp. 713–719).

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 acm conference on computer supported cooperative work* (pp. 175–186).

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1), 59.

Rubtsov, V., Kamenshchikov, M., Valyaev, I., Leksin, V., & Ignatov, D. I. (2018). A hybrid two-stage recommender system for automatic playlist continuation. In *Proceedings of the acm recommender systems challenge 2018* (pp. 1–4).

Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on machine learning* (pp. 880–887).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd acm conference on electronic commerce* (pp. 158–167).

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Singh, A. P., & Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining* (pp. 650–658).

Srebro, N., Rennie, J., & Jaakkola, T. (2004). Maximum-margin matrix factorization. *Advances in neural information processing systems*, *17*.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*.

Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2008). Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd kdd workshop on large-scale recommender systems and the netflix prize competition* (pp. 1–8).

Ungar, L. H., & Foster, D. P. (1998). Clustering methods for collaborative filtering. In *Aaai workshop on recommendation systems* (Vol. 1, pp. 114–129).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., … others (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., & Chen, Z. (2005). Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international acm sigir conference on research and development in information retrieval* (pp. 114–121).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet:

Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Zahálka, J., Rudinac, S., & Worring, M. (2015). Interactive multimodal learning for venue recommendation. *IEEE Transactions on Multimedia*, *17*, 2235-2244.

Zanker, M., Aschinger, M., & Jessenitschnig, M. (2007). Development of a collaborative and constraint-based web configuration system for personalized bundling of products and services. In *Wise.*

Zhang, Y. (2015). Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.*

Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.*

Zhuang, Y., & Kim, J.-K. (2021). A bert-based multi-criteria recommender system for hotel promotion management. *Sustainability.*