University of Kerbala
College of Computer Science & Information Technology
Computer Science Department

Recommendation System Based On Opinion Mining using Machine Learning Techniques

A Thesis

Submitted to the Council of the College of Computer Science & Information Technology / University of Kerbala in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Written by

Amenah Nahedh AbdulAmeer

Supervised by

Assist. Prof. Dr. Mohsin Hasan Hussein

2024 A.D. 1445 A.H.

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

﴿ اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِن شَجَرَةٍ مُّبَارَكَةٍ زَيْتُونَةٍ لَّا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نُّورٌ عَلَىٰ نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَن يَشَاءُ وَيَضْرِبُ اللَّهُ الْأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ ﴾

[النور: 35]

Supervisor Certification

I certify that the thesis entitled (**Recommendation System Based On Opinion Mining using Machine Learning Techniques**) was prepared under my supervision at the department of Computer Science/College of Computer Science & Information Technology/ University of Kerbala as partial fulfillment of the requirements of the degree of Master in Computer Science.

Signature:

Supervisor Name: Assist. Prof. Dr. Mohsin Hasan Hussein

Date: / /2024

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "Recommendation System Based On Opinion Mining using Machine Learning Techniques" for debate by the examination committee.

Signature:

Assist. Prof. Dr. Muhannad Kamil Abdulhameed

Head of Computer Science Department

Date: / /2024

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Recommendation System Based On Opinion Mining using Machine Learning Techniques**) presented by the student (**Amenah Nahedh AbdulAmeer**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with (**Excellent**) standing as a thesis for the Master degree in Computer Science.

Signature:
Name: Firas Sabar Miften
Title: Prof. Dr.
Date: / / 2024
(Chairman)

Signature:
Name: Noor Dhia Al-Shakarechy
Title: Assist. prof. Dr.
Date: / / 2024
(Member)

Signature:
Name: Akeel Abdulkarim Farhan
Title: Assist.prof .Dr.
Date: / / 2024
(Member)

Signature: Mohsin Hasan Hussein
Name: Assist. prof .Dr.
Title:
Date: / / 2024
(Member and Supervisor)

Approved by the Dean of the College of Computer Science & Information Technology, University of Kerbala.

Signature: Mowafak
Assist. Prof. Dr. Mowafak khadom Mohsen
Date: / / 2024
(Dean of College of Computer Science & Information Technology)

Dedication

Peace be upon the designer of the human civilization that is established on the Unity of Almighty Allah, Peace be upon the redeemer of human volition and thought, Peace be upon the Seal of the Prophets and the Master of all beings; the Holy Prophet Muhammad (peace be upon him and his Household.

I dedicate my success in completing this dissertation and obtaining this academic degree to the Seal of his Successors, the Reviver of his Faith, and the resuscitator of his Mission: Imam al-Mahdi Al-Montazar, peace be upon him who shall fill the earth with justice and righteousness after it will be filled with injustice and prejudice. I intend for my study to be a preparation for his blessed appearance.

Acknowledgement

From the depths of my being, I thank Allah, the source of all strength, for guiding me through hardship and accepting my humble pleas. With His grace, this accomplishment soars beyond a mere dream.

My deepest gratitude goes to my esteemed supervisor, Assist. Prof. Dr. Mohsin Hasan Hussein. His invaluable guidance and unwavering encouragement were the compass that steered me through the intricacies of this work.

I would like to thank my parents for their duaa and thank my sister, brother, relatives, and my friends for supporting me.

Finally, I would like to say that all the words are unable to express the extent of my thanks to my husband who supported and encouraged me all the time.

Abstract

The Internet and the Web have made it possible for a vast amount of information to be shared and accessed by large numbers of people. This has led to a problem called information overload (the challenge of making decisions when faced with too much information). This problem necessitated the creation of recommendation systems, which address the information overload challenge by suggesting products or services that may be useful to users and their interests.

Recommendation systems may face several problems, including cold-start and sparsity. These problems lead to a decline in the performance of the recommender system.

In this work, a Recommender System based on Textual Reviews and using the Deep Learning method (RS-TRDL) was proposed to perform two tasks namely alleviate the user cold-start and alleviate sparsity problems, this leads to improving the performance of the proposed method. Textual reviews were used as additional information alongside the users' numerical ratings. Important aspects were extracted from these reviews, in addition to the polarity of sentiment by using one of the deep learning algorithms, which is Long Short-Term Memory (LSTM) algorithm, to then benefit from these aspects in the recommendation process.

The RS-TRDL model employed a comprehensive pre-processing stage for the dataset. This stage encompassed various steps, including Handling Missing Values and Data Labeling. Additionally, it incorporated text-specific pre-processing operations such as Text Cleaning and general Text Preprocessing, it then proceeded with aspects extraction. This step employed spaCy for Noun Extraction via Part-of-Speech (POS) tagging. Additionally, Topic Modeling was performed using the BERTopic algorithm. Finally, Sentiment Analysis was conducted utilizing the Long Short-Term Memory (LSTM) algorithm.

After extracting the aspects, the cold-start users and non-cold-start users were treated separately. For cold-start users, the Rating prediction process was

done using K-Nearest Neighbours (KNN) algorithm based on the ratings of the non-cold-start users who share the same aspects of the same items and have high helpfulness value.

For non-cold start users, firstly grouping process was done based on aspects extracted from users' reviews, then a similarity matrix was created for each group using Cosine similarity measure. Finally, the rating prediction process was performed using KNN based on the ratings of the nearest users belonging to the same group and having a high helpfulness value.

Extensive Experiments were conducted by the proposed system on two Amazon datasets: Amazon Electronics and Amazon Fine Food. The experimental results show that the proposed RS-TRDL model exceeded all literature-reviewed comparison methods in the rating prediction process for both tasks it was built to perform. It is worth highlighting the model's consistent performance across both tasks, as evidenced by the improvement range of 0.24% to 34.32% for alleviating the user cold start problem task and 3.21% to 58.7% for alleviating the sparsity problem and enhancing the recommender model task.

Declaration Associated with this Thesis

Some of the works presented in this thesis have been published or accepted as listed below.

1) The research paper entitled: “Recommendation System Based on Opinion Mining: A Survey” was presented at The Fourth International - Seventh National Scientific Conference (ISC-SNC-2023) held on 24-26 September 2023 and accepted to be published in IEEE Xplore digital library.

2) The research paper entitled: “Alleviating the User Cold-Start Problem in Recommendation Systems Based on Textual Reviews using Deep Learning” was presented at The 4th International Intelligent Systems and Applications Conference (ISCAU:ISA23) held on 16-17 December 2023 and accepted to publish in Iraqi Science Journal (IS), a Scopus-indexed with a CiteScore of 1.1.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
ABSA	Aspect-Based Sentiment Analysis
Adam	Adaptive Moment Estimation
AE	Amazon Electronics
AFF	Amazon Fine Food
BERT	Bidirectional Encoder Representations from Transformers
BPR	Bayesian Personalized Ranking
CAML	Co-Attentive Multi-task Learning
CB	Content-based Filtering
CF	Collaborative-Filtering
CNN	Convolution Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Tree
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
JSD	Jensen Shannon divergence
KNN	K-Nearest Neighbours
LDA	Latent Dirichlet Allocation

LR	Logistic regression
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCNN	Multichannel Convolutional Neural Network
ME	Maximum Entropy
MF	Matrix Factorization
ML	Machine Learning
MLP	Multilayer Perceptron
MRR	Mean Reciprocal Rank
NB	Naive Bayes
NLP	Natural Language Processing
POS	Part Of Speech
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RS	Recommendation System
RS-TRDL	Recommendation System based on Textual Reviews using Deep Learning
SA	Sentiment Analysis
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TF-IDF	Term Frequency – Inverse Document Frequency
TM	Text Mining
TN	True Negative
TP	True Positive

CHAPTER ONE

INTRODUCTION

1.1 Overview

This chapter delves into the critical area of integrating recommender systems with opinion-mining techniques. It explores the context, objectives, and the specific problem this research addresses. Additionally, it provides a comprehensive review of the most relevant and closely related prior work in the field and presents further details of the chosen methodology, utilized datasets, and the evaluation methods employed for each research.

1.2 Introduction

Nowadays, the Internet offers a wide variety of services, products, and information as a result of the development of the digital world, e-commerce, and social media [1]. The Internet and the Web have allowed a vast amount of information to be shared and accessed by large numbers of people. This has led to a problem called information overload, which is the challenge of making decisions when faced with too much information [2]. This is especially common in e-commerce and social media, where users are constantly bombarded with choices [3]. Users find a variety of news, products, movies, and people when surfing social networks. Finding what is relevant and satisfactory to the users is quite difficult given the multiplicity of options [4].

One of the key solutions presented for the problem of information overload is recommender systems, which can offer suggestions for items that the users might be interested in [5]. The suggestions made by a recommender system are intended to assist users in different processes of decision-making, such as what products to buy, what articles to read, what songs to listen to, or what films to watch by analyzing their past behavior [6]. This can save the time consumed by users as well as their efforts, and help them to deal with the problem of information overload.

RSs have become increasingly popular among researchers in some disciplines, such as Machine Learning, Information Retrieval, and Human-Computer Interaction. This interest has led to the adoption of RSs in industrial domains such as e-commerce marketing and the movie industry [7]. Many popular companies, such as Amazon, Netflix, and eBay, use recommender systems to suggest products to customers based on their past purchases, browsing history, and the products they are currently viewing [8].

In general, there are three main types of recommendation approaches: Collaborative Filtering (CF), Content-Based filtering (CB), and Hybrid strategies [9]. CF is the most popular and versatile approach to recommender systems as it is simple to implement and can be used with a wide range of items, including non-textual data. However, it has two main drawbacks: sparsity of ratings and the cold-start problem [10]. CF approach needs a sufficient number of ratings from a user on an item to make an effective prediction and get accurate results [11].

After all, these techniques are still insufficient, especially when the degree of rating sparsity is high or the target user does not have many prior ratings. Accordingly, researchers have proposed several solutions to the problems of the sparsity of ratings and cold-start, as these problems can have a negative impact on the accuracy of RSs [7].

One common solution is to use a hybrid RS, which combines different techniques to generate more accurate and precise recommendations. Another solution is to use additional information, such as users' demographic information and/or social information [12]. It is encouraging to note that due to the Web's current state, users are becoming more and more easily expressing themselves and sharing their opinions about products on e-platforms through reviews [13]. Therefore, a more promising solution was to use users' textual reviews of items they are interested in [14]. In addition to ratings, users' text reviews provide a wealth of information that can be used to better understand their preferences. These reviews are usually textual comments that describe why the user likes or dislikes a product according to their actual usage experiences [15].

This motivated researchers to provide review-based recommendation systems, which make use of user textual feedback and analyze it to estimate their preferences. In recent years, various types of review-based recommender systems have been provided to integrate the helpful information found in user-generated textual reviews into the user modeling and recommendation process [14]. Multiple types of review elements can be extracted using advanced text analysis and opinion-mining techniques.

Opinion mining is a fundamental task in Natural Language Processing (NLP), it uses computational linguistics and NLP techniques to analyze user reviews and then identify and extract subjective information from textual data. This data can then be used to make recommendations based on the sentiment expressed in the user's feedback [16].

In this thesis, an attempt has been made to alleviate the problem of new users or the so-called Cold Start problem, in addition to the sparsity problem by proposing a novel recommendation method. This method is based on opinion-mining techniques, specifically on aspect-based opinion mining. Also, the Long Short-Term Memory (LSTM) algorithm which has certain contributions in the field of review-based recommender systems was used to obtain superior results.

1.3 Problem Statement

Cold start and sparsity problems are persistent obstacles for many recommender systems. Their detrimental influence on accuracy has spurred extensive research, with numerous literature reviews dedicated to investigating potential solutions throughout the past years. These two obstacles are addressed in this thesis.

The cold-start problem arises when new users or items enter the system, lacking enough rating data for effective recommendations. CF relies on sufficient user-item interactions to generate accurate predictions, rendering it inoperable for users with sparse rating histories. It is like meeting someone new and trying to guess their taste in movies: if you don't know anything about them, it's a shot in the dark. The same goes for recommender systems with new users or items - not enough data, no good recommendations.

The sparsity in user-item rating matrices, where users rate only a small fraction of available items, poses a significant challenge. This problem will cause a decline in the accuracy of recommendation models when Inferring user relationships based on limited data which is difficult, leading to inaccurate recommendations.

1.4 Aim of thesis

This thesis presents RS-TRDL model which incorporates opinion-mining techniques with the CF approach for RS and aims to perform the following two tasks:

1) Alleviating the user cold start problem.
2) Address the sparsity problem to enhance the prediction accuracy of the proposed RS model.

To perform these two tasks, numerous research efforts have explored the potential of textual reviews. By leveraging the rich implicit information contained within these reviews, beyond explicit numerical ratings, some studies have achieved significant improvements. It can be said that the success of these approaches in harnessing the insights found within textual reviews has provided the impetus for thinking about selecting the best functions (i.e., Aspect extraction, Sentiment analysis, and Topic modeling).

1.5 Contributions

The main contribution of this thesis is to propose an effective model using a popular deep learning algorithm in the NLP field which is LSTM with a crucial feature within the textual dataset namely "Helpfulness" which quantifies the value or utility of a user review for other users or a specific target audience

1.6 Thesis Organization

The thesis is divided into five chapters. Each chapter begins with a short background that underlines the key contributions and offers an impression of the chapter. The summaries of the chapters are as follows:

Chapter 2 offers the definition, concepts, and types of RS. It also highlights the concepts, levels, and techniques of opinion mining. In addition to LSTM algorithms and the evaluation metrics used in this thesis.

Chapter 3 discusses the methodology of the proposed system.

Chapter 4 demonstrates the evaluation results and the discussion of the proposed system.

Chapter 5 articulates conclusion statements and future trends.

1.7 Related Work

The growth and expansion of the World Wide Web and electronic commerce have encouraged users to create and share product reviews to express their opinions. These reviews are typically written by users in free text format and represent a variety of aspects or viewpoints about the experience a user had with a particular product [17]. Hence, they provide a very significant information source on user preferences and can be utilized to develop user profiles and enhance personalized recommendations. Therefore, various recent attempts have been made to incorporate valuable information found in user reviews into the recommendation process [7]. An overview of recently published research on review-based recommender systems is summarized in this section.

In the work of Zhou et al. (2016) [18], LSTM was used to solve the problem of classifying review usefulness and combined its outputs with the built recommender system using the matrix factorization model for the rating prediction process.

Lin et al. (2018) [19] proposed MulAttRec model which is a recommendation model, that leverages a multi-level attention mechanism to explore the most valuable insights from user reviews. By pinpointing crucial words and reviews, MulAttRec distills the essence of user preferences. Furthermore, a hybrid prediction layer blends the strengths of a Factorization Machine (FM) and a Deep Neural Network (DNN), capturing both low-order and high-order feature interactions to model the intricate relationships between users and items.

As for Wu et al. (2019) [20], they merged the ratings of the users and the textual reviews information into a unified model. The model makes use of CNNs and an attention mechanism to learn the pertinent latent features by taking into account their related reviews. The model creates latent rating embeddings for users and items from the interaction matrix by using a rating-based component. The learned content features and latent rating embeddings are combined in a Factorization Machine (FM) to find the final rating score.

Chen et al. (2019) [21] proposed a Co-Attentive Multi-task Learning (CAML) approach. This innovative approach leverages the inherent correlations between recommendation and explanation tasks by tightly coupling them within a single model, allowing for a comprehensive understanding of the recommendation choices and thus achieving significant advancements in both the accuracy and explainability of recommendations.

Liu et al. (2020) [22] propose a Hybrid neural recommendation model called HRDR that extracts user and item embeddings from reviews and ratings. First, a

Multilayer Perceptron (MLP) network is used to obtain the rating representations from rating data. The next step is to create review-based representations using CNNs with an attention mechanism, where each review has a corresponding informativeness score.

Da’u et al. (2020) [23] proposed a recommendation model based on MCNN model relying on weighted aspect-based opinion mining. They first described how the MCNN model could be used to extract aspect terms using different model channels. Next, they explained how the extracted aspects from the user text review could be utilized to generate aspect ratings using a lexicon-based approach. They also illustrated how aspect ratings are used to generate weighted opinions and how the TF approach is employed to infer rating prediction. Later, they used datasets from the actual world to evaluate the model's performance in terms of aspect extraction and item recommendation.

Hung (2020) [24] proposed a recommender system based on sentiment analysis. They concentrated on the document level of opinion mining, which involves detecting if an expressed opinion is positive, negative, or neutral. First, they used hybrid deep learning models CNN-LSTM to classify the usefulness of each review. In this step, they generated a vector representation for each user product review, and then they trained their classification model using those vectors. Next, they evaluated the efficiency of their classification model. After obtaining the result in the first step, they incorporate it into the recommender system based on user-item rating data. They had experiences with Amazon food reviews and used the RMSE score to evaluate the performance of their model.

Harrag et al. (2020) [25] proposed a hybrid strategy by integrating sentiment analysis and recommender systems to address the issue of data sparsity by predicting the rating of items from users’ reviews using NLP techniques and text mining. They focused on Arabic reviews and used the Opinion Corpus for Arabic (OCA) dataset (an Arabic dataset that includes 500 reviews of various movies from various websites). Before feeding the texts into the sentiment analysis phase, they preprocessed the texts first. Later, they used the SVM algorithm as supervised machine learning to extract the polarity of each review and train the model. This phase produced the review polarity values (+1, -1) as output. Then the values were utilized again as a recommendation system phase input.

In [26] Dual learning-based framework was proposed by Sun et al. (2020), which enhances the power of dual learning to capture the probabilistic connection between two crucial tasks in review-based recommender systems: predicting user preferences and content generation. This framework harnesses the duality correlation between these tasks to enhance the performance of both preference prediction and review generation.

Han et al. (2020) [27] introduced a latent factor model called adaptive deep latent factor model (ADLFM), this model is capable of learning user preference factors for the specific items being considered. This flexibility is achieved by employing a user representation approach that enhances descriptions of the items users have rated, rather than relying solely on traditional user-item rating data.

In the work of Systems et al. (2021) [28], a recommendation approach was presented, integrating sentiment analysis with collaborative filtering methods to enhance recommendation accuracy. This approach rests upon a dynamic, adaptive architecture that incorporates feature extraction techniques and deep learning models specifically designed to leverage sentiment information extracted from user reviews. The experiment was conducted with the recommender with sentiment analysis on different values of the β to obtain different results. Their findings indicated that a β value of 0.3 yielded the best results.

Oudah et al. (2022) [29] developed a recommendation model based on text reviews as additional information. The proposed model has been shaped by five fundamental steps. The First step is data preprocessing. Next, two concurrent processes will be performed, namely, text classification (by using the DistilBERT model) and topic modeling (by using the LDA model). The fourth step is a combination process between the text polarity and the topic probabilities of the text. Subsequently, a process of text similarity is implemented using the JSD metric. In the last step, after integrating the inferred adjustment weights in the classification score equation, the Naive Bayes model is used to produce a recommendation of items that satisfy the target user. The researchers made use of three datasets from Amazon. They demonstrated that their proposed model exceeded all comparison methods in the Top-N recommendation task.

Baizal et al. (2022) [30] employed Improved Collaborative Filtering (ICF), a method that exploits user similarities to predict missing values in the user-item rating matrix. ICF operates by first identifying users with similar characteristics to the active user and then calculating the average distance between them. This similarity-based approach guides the subsequent prediction of missing ratings for items the active user has interacted with but hasn't been rated by any of the identified similar users.

Ho et al. (2023) [31] designed a Transformer-based recommender system to exploit the richness of both utility matrices and textual data sources. This model meticulously integrates feature extraction techniques, drawing insights from multiple information sources. It employs a conversion algorithm to segment (user-item) feature vectors into sequences of token vectors, tailor-made for subsequent classification modeling. By incorporating transformer models into the recommender systems, the recommender system gains the ability to discern and utilize the most relevant features, mitigating the impact of noise and conflicting information.

Bayu Samudra Siddik et al. (2023) [32] investigated the effectiveness of various Matrix Factorization (MF) algorithms for predicting user ratings in the context of a food item recommendation system. By comparing Singular Value Decomposition (SVD), SVD with Implicit Ratings (SVD++), and Non-Negative Matrix Factorization. NMF achieved the lowest average prediction error, measured by Mean Absolute Error (MAE).

In addition to these researches, many studies have specialized in treating or alleviating the problem of cold start. Over many years, several works have been introduced to address the cold start problem for providing a positive experience to new users and ensuring that the recommendation system remains efficient as new items are encountered. For instance, by Xu et al. (2017) [33], a strategy known as RAPARE (Rating Comparison for Alleviating Cold-Start) has been proposed to address the challenge of cold-start scenarios. This strategy, formulated as an optimization problem, leverages insights gleaned from existing user and item data to effectively calibrate the latent profiles of new users or items that lack substantial interaction history. Their proposed generic RAPARE strategy was instantiated on both matrix factorization-based (RAPARE-MF) and neighborhood-based (RAPARE-KNN) collaborative filtering.

Fu et al. (2019) [34] proposed a Review and Content-based Deep Fusion Model named RC-DFM for a cross-domain recommendation. First, they extended Stacked Denoising Autoencoders (SDAE) to successfully fuse review texts and item contents with the rating matrix in both the auxiliary and the target domains. In this manner, the learned latent factors of users and items across both domains preserve more semantic information for recommendation. Once user latent factors had been transferred between the two domains, they used a multi-layer perceptron to produce predictions in order to address the data sparsity and cold start problems. They used Amazon to evaluate the efficiency of their model. The experimental results showed that their model was superior.

Herce-Zelaya et al. (2020) [35] presented an approach for using social media data to generate a behavioral profile to classify users. Based on this classification, predictions will be created via machine learning techniques such as decision trees and random decision forests. The system would use this data to generate user profiles, which will be the input for the engine of recommender systems. Hence, the user would not have to actively provide any kind of explicit data other than their social media source. This eventually alleviated the cold start problem. The researchers evaluated prediction accuracy using precision, RMSE, and F-measure.

Feng et al. (2021) [36], presented a novel ranking model RBPR, which integrated explicit ratings and implicit feedback into a single model. First, the proposed method employed the Singular Value Decomposition (SVD) model for pre-processing to increase the density of explicit ratings. After that, Probabilistic Matrix Factorization (PMF) and Bayesian Personalized Ranking (BPR) were unified jointly. BPR was used to reconnoiter the implicit features of users and items from implicit feedback data. On the other hand, PMF was employed to reconnoiter the explicit features of users and items from explicit ratings. Lastly, the final features of users and items were determined by taking the shared latent features of users and items that were extracted from both models. Four original datasets were used to test this model. They are Movielens 100k, Movielens 1M, FilmTrust, and Ciao artificially. According to experimental results, the proposed approach RBPR performs well in terms of different evaluation metrics.

By Wang et al. (2021) [37], a framework MetaTL was proposed to improve sequential recommendations for cold-start users. In a meta-learning manner, MetaTL learns a model that can be adapted for new users with a few interactions. The proposed MetaTL can deliver significant improvement based on the experiments on three real-world datasets and using the evaluation metric Mean Reciprocal Rank (MRR). They also evaluate the Hit Rate (Hit) for the top-1 prediction.

Finally, Mondal et al. (2022) [38] proposed DeCS model that uses a deep neural network (DNN) framework and addresses the cold start problem in recommendation systems and works primarily in stages that involve creating embeddings and vectors followed by training and prediction of three fundamental metrics MSE, MAE, RMSE by the framework. Table 1.1 presents a summary of related works.

Table 1.1: Summary of Related Works

Reference	**Year**	**Technology**	**Dataset**	**Results**
[18]	2016	LSTM- matrix factorization	Amazon Fine Food Review	RMSE = 1.1198
[19]	2018	Multi-level attention-based CNN	Yelp Books Electronics	RMSE metric ranges from 0.812 to 1.206
[20]	2019	Convolution Operations and Attention Mechanism	Beer, Musical Instruments, Office Products, Digital Music, Video Games, Tools Improvement, Yelp 16-17.	MSE metric ranges from 0.553 to 1.446

[21]	2019	Co-attentive multi-task learning using encoder-selector-decoder	Amazon Electronics, Amazon Movies&TV, Yelp	RMSE metric ranges from 0.987 to 1.180
[22]	2020	CNN	Yelp 2013, Yelp 2014, Amazon Gourmet Food, Amazon Video Games	RMSE metric ranges from 0.933 to 1.011
[23]	2020	MCNN	Amazon Musical Instruments, Amazon Automotive, Amazon Instant Video, Yelp.	RMSE metric ranges from 0.7990 to 1.2501 MAE metric ranges from 0.5497 to 0.9785
[24]	2020	MF integrated Sentiment Analysis using CNN-LSTM	Amazon food reviews	RMSE = 1.1536
[25]	2020	SVM	Opinion Corpus for Arabic dataset (OCA)	Precision = 0.93 , MAE = 0.422
[26]	2020	Dual learning-based framework with both Preference prediction and review Content generation	Amazon Books and Amazon Electronics	RMSE metric ranges from 0.8376 to 0.9672
[27]	2020	combination of deep learning technique with the latent factor model	15 Amazon dataset	MSE metric ranges from 0.915 to 1.483 MAE metric ranges from 0.712 to 0.961
[28]	2021	CNN - LSTM sentiment models with SVD, NMF, and SVD++ algorithms.	Amazon Fine Food Reviews and Amazon Movie Reviews	MAE metric ranges from 0.5770 to 0.9706 RMSE metric ranges from 0.8577 to 1.2312 NMAE metric ranges from 0.1443 to 0.2427
[29]	2022	Naïve Bayes Classifier (NBC)	Amazon Musical Instruments, Amazon Automotive, Amazon Instant Video	Precision metric ranges from 0.81 to 0.89 Recall metric ranges from 0.83 to 0.91 F1 metric ranges from 0.82 to 0.90
[30]	2022	improved collaborative filtering method	Amazon electronic	MAE = 0.80 RMSE = 1.10
[31]	2023	Transformer Model using utility matrix and textual sources	MovieLens, Amazon-Toys and Games, Amazon-Electronic, Amazon-Video and Games	MAE metric ranges from 0.445 to 1.572 RMSE metric ranges from 0.743 to 2.293 Precision metric ranges from 47.75% to 92.07%

[32]	2023	Singular Value Decomposition (SVD), SVD with Implicit Ratings (SVD++), and Non-Negative Matrix Factorization (NMF)	Amazon Fine Food Reviews dataset.	MAE metric ranges from 0.7311 to 1.1752 RMSE metric ranges from 1.0607 to 1.4019 Hit Ratio metric ranges from 0.0003 to 0.0025
[33]	2017	rating comparison strategy based on MF and KNN	MovieLens, EachMovie, Yelp, Amazon Automotive and Amazon Electronic	RMSE metric ranges from 0.94 to 1.49
[34]	2019	Reviews and contents based deep fusion model for cross-domain recommendation	Amazon Movies & Music CDs, Amazon Books	RMSE metric ranges from 0.9468 to 1.0706 MAE metric ranges from 0.7590 to 0.8589
[35]	2020	Decision Trees and Random Decision Forests	Twitter dataset	Accuracy error ranges from 0.298 to 0.338 F1 error = 0.634 RMSE metric ranges from 0.569 to 0.532
[36]	2021	Probabilistic Matrix Factorization (PMF) and Bayesian Personalized Ranking (BPR)	FilmTrust, Ciao, Movielens 1M, Movielens 100k	Precision@N metric ranges from 0.0113 to 0.1243 Recall@N metric ranges from 0.0203 to 0.3057 MAP metric ranges from 0.024 to 0.26 MRR metric ranges from 0.038 to 0.37
[37]	2021	Meta-learning framework	Amazon Electronics, Amazon Movie, Goodreads book	Hit@1 metric ranges from 0.224 to 0.420 MRR metric ranges from 0.352 to 0.555
[38]	2022	Deep learning-based recommendation system	MovieLens-100K, MovieLens-1M, MovieLens-10M, MovieLens-20M, Douban Book, Douban Movie, Douban Music, Amazon Movie, Amazon Electronics, and Amazon Book.	MSE metric ranges from 0.4338 to 1.2911 RMSE metric ranges from 0.6883 to 1.1362 MAE metric ranges from 0.4691 to 0.8745

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Overview

This chapter explains the Recommendation System (RS) with its definition, basic concepts, and types. In addition, it sheds light on opinion mining and advanced analysis techniques of textual data. Finally, an overview is made of deep learning and Long Short Term Memory (LSTM) algorithm used in this thesis.

2.2 Recommendation Systems

Most people now have access to an enormous amount of data due to the Internet and the growth of the Web. Each user will therefore have the problem of finding their best requirements because extracting information from this large volume of data could be a challenging and complex process [29].

RS is a type of data filtering system that provides suggestions to users about products or services based on their preferences, interests, or previous behavior [39]. The main objective of recommendation systems is to help people find easily and quickly relevant and new items that they would find interesting or useful, including products, movies, music, articles, and other types of material [40]. They can do this owing to their ability to speed up and simplify the users' searching processes. Therefore, recommendation systems are considered an important tool for solving problems by reducing search time and predicting users' preferences and the products they are interested in [41].

RSs have grown to be an essential part of our digital lives, assisting users in navigating the enormous number of options available and personalizing their online experiences. They are commonly used in several kinds of online services and platforms, including social media platforms, e-commerce websites (such as

Amazon, E-bay, and E-shops Taobao), content platforms, and streaming services (such as Netflix and Spotify) [15].

RSs analyze user data, item characteristics, and user-item interactions using several kinds of techniques and algorithms, including machine learning, data mining, and statistical modeling. These techniques learn and enhance their recommendations frequently over time by integrating updated data and feedback from users.

2.2.1 Recommender Systems' Feedback Information

Personalized recommendations hinge on the assumption that the system possesses an intimate knowledge of each user. This necessitates the construction of user profiles encompassing both user details and preferences. While a user-based model acts as the backbone of any recommender system, the specific method for gleaning and utilizing individual user information varies based on the employed recommendation technique. For instance, the system can implicitly capture user preferences by monitoring their behavior. However, users also regularly encounter explicit prompts to express their preferences directly [42].

The limitations of traditional recommender systems in offering effective recommendations necessitate the exploration of additional information sources. This information can be acquired directly (explicit) or derived through hidden patterns within existing data (implicit) [34]. However, the crucial question remains: what specific types of information can augment traditional systems and enhance their recommendation accuracy?

RSs collect data about user preferences for diverse products like books, movies, music, and travel destinations. This information can be explicitly gathered through user ratings and feedback, or implicitly inferred from browsing behavior and downloaded content [40].

A form of explicit information is for users to express their opinions, where users not only rate items through numerical ratings but also provide personal text reviews that support their preferences and shed light on the underlying reasons for their choices. These reviews are valuable as they are commonly presented in the form of textual comments detailing users' preferences or critiques regarding the evaluated items. They serve as a valuable resource, offering insights into users' preferences, enabling the development of intricate user profiles, and improving personalized recommendation systems [43].

Additionally, RSs may utilize user demographics (age, race, gender) and social network data (friends, followers) alongside user-generated content and social media interactions to personalize recommendations further.

While the apparent simplicity of explicit feedback through rating scales is an advantage, employing complex scales (1-5) or binary (like/dislike) options can burden users with cognitive load and potentially deter them from engaging with the rating process. Conversely, implicit feedback, extracted from natural user actions like following or commenting, captures user preferences indirectly without overt prompting. This passive monitoring allows for unobtrusive data collection [39].

2.2.2 User-Item Rating Matrix

The core of any RS is a rating matrix, which contains user opinions on various items within the system's domain. This sparse matrix, with its gaps and inconsistencies, serves as a key performance indicator for the system itself. Users express their preferences through "ratings," recorded in diverse formats like numerical scales (1-5 stars), binary choices (like/dislike), or even simpler binary indicators like "purchased" on e-commerce platforms [6].

The rating matrix is comprised of three elements: users, items, and ratings (representing user preferences). While this matrix aims to capture all user opinions, it often remains sparse due to the common reluctance to provide explicit feedback. Consequently, a significant portion of entries remain unrated and unknown, typically represented by "NaN" values.

Users	**Items**					
	$\boldsymbol{i_1}$	$\boldsymbol{i_2}$	$\boldsymbol{i_3}$	$\boldsymbol{i_4}$	$\boldsymbol{i_5}$	$\boldsymbol{i_6}$
$\boldsymbol{u_1}$	3			2		5
$\boldsymbol{u_2}$	4		3		2	
$\boldsymbol{u_3}$	5	4	3		3	2
$\boldsymbol{u_4}$	1			5		4

Figure 2.1: User-Item Rating Matrix

Figure 2.1 shows the rating matrix for four users and six items, where $U = \{\boldsymbol{u_1}, \boldsymbol{u_2}, \boldsymbol{u_3}, \boldsymbol{u_4}\}$ to signify the users, $I = \{\boldsymbol{i_1}, \boldsymbol{i_2}, \boldsymbol{i_3}, \boldsymbol{i_4}, \boldsymbol{i_5}, \boldsymbol{i_6}\}$ for the products (items), and $\boldsymbol{R_{m \times n}}$ as the matrix of ratings $\boldsymbol{r_{u,i}}$ recorded in the system, with $u \in U, i \in I$.

User engagement with items typically expressed through rating systems like Amazon's five-star scale, provides valuable insights into their preferences. 4 or 5 stars signify positive opinions across diverse product categories, while 1 or 2 stars indicate the opposite. Neutral sentiment is often represented by 3 stars, though some studies might categorize it depending on specific research objectives. To address missing ratings, recommender system algorithms predict these "blank" entries in the matrix and recommend items with positively estimated ratings to targeted users.

2.3 Challenges of Recommendation Systems

Recommender systems face two major challenges: rating sparsity and cold-start [34]. Given the significant impacts of these issues on recommendation accuracy, research efforts are actively focused on attempting to develop effective mitigation strategies.

2.3.1 Rating Sparsity

Real-world application of recommender systems often reveals sparse rating matrices. This sparsity arises from users typically rating a limited number of items relative to the vast system catalog, resulting in numerous empty or unknown entries within the user-item matrix. This phenomenon is known as the sparsity problem in recommender systems [10].

The challenge in making accurate predictions within a sparse data environment lies in supplementing limited user ratings. To address this, leveraging users' demographic data, textual reviews ,personal interests, and educational backgrounds proves a well-defined approach [12].

2.3.2 Cold Start

Recommender systems (RSs) grapple with a major challenge known as the cold-start problem. This arises when either a new user joins the system, lacking sufficient data to predict their preferences, or when a new item is added, lacking rating history to assess its appeal. Consequently, two variations of the cold-start problem exist user-based cold start for new users and item-based cold start for new items [44].

To address the challenge, researchers have primarily focused on leveraging content-based features specific to new users or items. These features are then integrated with existing rating data in a hybrid system, mitigating the issue's impact. Additionally, explorations have extended to utilizing alternative data sources like demographic information and social connections [45].

This research attempts to alleviate the cold-start and sparsity problems in recommender systems by leveraging the rich data within user reviews. Each user's review was delved into and extracted key aspects of their preferences to enhance recommendation accuracy, particularly for new users.

2.4 Types of Recommendation Systems

Generally, as shown in Figure 2.2 RSs are classified into three main types: content-based (CB), collaborative filtering (CF), and hybrid systems.

CF builds its recommendations on the assumption that users with similar preferences for past items will likely exhibit similar preferences for future items. In contrast, CB recommender systems focus on the inherent characteristics or content of items or users to suggest items with matching features [46].

Lastly, to leverage the strengths of both approaches while mitigating their limitations, hybrid recommender systems integrate CB and CF techniques in diverse ways. This synergistic combination capitalizes on the personalized insights of CB and the collaborative intelligence of CF to deliver more robust and accurate recommendations [25].

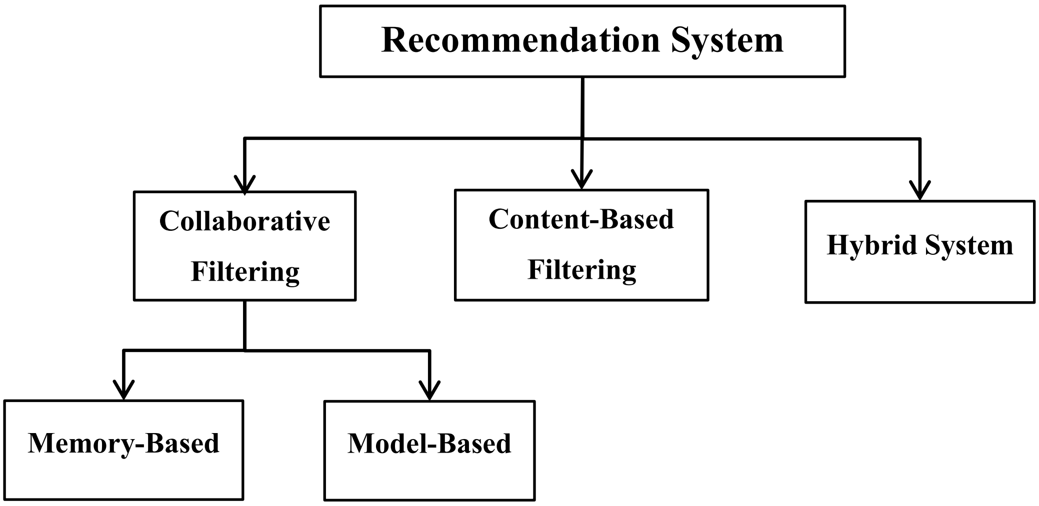


Figure 2.2: Types of Recommendation Systems. [47]

Collaborative Filtering dominates the recommendation system landscape due to its ease of implementation and broad applicability [3]. It works by measuring the similarity between users or items (or both) to predict which items a target user might prefer. This flexibility allows it to handle diverse item types. However, CF faces challenges like sparse rating matrices and cold-start issues [10].

CF within recommender systems draws its charm from several factors, primarily the elegant simplicity of its core data structure which is the user-item matrix. Though this approach conceals vital item information beyond user ratings, the abundance of CF datasets has spurred the development of diverse techniques.

CF techniques are divided into memory-based and model-based method categories [48].

- **Memory-Based Collaborative Filtering**

Memory-based Collaborative Filtering methods are a type of CF that deals with the user-item rating matrix directly. This method analyzes user and/or item similarities based on existing rating patterns, predicting missing values and recommending items likely preferred by target users [48].

Among memory-based collaborative filtering algorithms, the K-Nearest Neighbors (KNN) method stands out for its popularity. KNN comes in two flavors: user-based filtering, which predicts user preferences based on similar users' behavior or ratings, and item-based filtering, which recommends items based on their similarity to the user's past favorites [12].

To make recommendations for user A, the following steps should be performed:

i. The KNN algorithm first identifies its k most similar neighbors based on a chosen similarity measure.

ii. It leverages the ratings of these neighbors to predict A's preference for a specific item I.

iii. Finally, the algorithm recommends the top n items with the highest predicted scores to A.

While user-based collaborative filtering enjoys significant success, its application to large datasets is hampered by demanding memory and processing time requirements [49].

- **Model-Based Collaborative Filtering**

Model-Based Collaborative Filtering is a type of collaborative filtering that uses machine learning techniques to learn the patterns and relationships in user-item interaction data and generate recommendations [48]. These approaches delve into user-item interaction data using algorithms like Bayesian networks, neural networks, or matrix factorization, extracting patterns and building predictive models [12]. This allows them to recommend items even for users with sparse interactions, setting them apart as the top contenders in the RS scene [9].

The heavy lifting for model-based CF occurs during the training phase, where intricate learning algorithms construct a robust predictive model capable of personalized recommendations [50].

2.5 Similarity Measures

Within the domain of memory-based recommender systems, the computation of similarity weights occupies a position of importance, wielding a profound effect on both accuracy and performance [39]. This essential calculation underpins the fundamental premise of memory-based collaborative filtering, which hinges upon the extraction of similarity values between the system's constituent entities, namely users and items.

In user-based collaborative filtering, a similarity metric quantifies the degree of similarity between user pairs. This matching, based on shared ratings for all items, forms the foundation for recommending new items to a target user [12].

In high-dimensional recommendation systems, where numerous features or factors are considered, relying on comparisons of similarity presents substantial challenges in terms of both time and memory usage. Nevertheless, the proven effectiveness of this approach, particularly in offline research settings, can often compensate for these costs.

There are many measures used to calculate similarity between users. Cosine similarity is one of the most common measures in this field used to estimate the relative similarity between users or items within such systems.

Cosine similarity is a mathematical measure used in recommendation systems to determine the similarity between two users or vectors in a multi-dimensional space [15]. It is frequently employed in RSs to determine how closely aligned users or items are in terms of their preferences or characteristics. Cosine similarity measures the angle between these vectors. A smaller angle indicates greater similarity and a larger angle implies dissimilarity, as in the following equation:

$$\boldsymbol{Sim_{AB} = \cos\theta = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ \sqrt{\sum_{i=1}^{n} B_i^2}}} \qquad (2.1)$$

Where $\boldsymbol{Sim_{AB}}$ represents the degree of similarity value between user A and user B, $\boldsymbol{A_i}$ is the actual rating of user A on an item i, and $\boldsymbol{B_i}$ is the actual rating of user B on an item i. The result will be a number between 0 and 1. The number 1 indicates that the two users are identical, and the zero number represents that they are opposite of each other.

2.6 Review-Based Recommendations Systems

Traditional recommender systems often struggle to grasp the hidden preferences of users, leading to suboptimal performance. because they depend only on users' overall ratings of items; they do not take into account users' opinions about different aspects of an item [17]. Their reliance on simple rating data overlooks the nuanced motivations behind user interactions. As a result, the rating may not accurately reflect the user's opinions by omitting essential information [14].

As a result of the growth and expansion of e-commerce, users have been encouraged to create and share reviews expressing their opinions about products. Therefore, several attempts have been made recently to incorporate the valuable information found in user reviews into the recommendation task [17]. These reviews are typically written by users in free text format and represent a variety of aspects or viewpoints about the experience a user had with a particular product. As a result, they provide a very significant information source on user preferences and can be utilized to develop user profiles and enhance personalized recommendations [7].

A review-based recommendation system builds upon traditional approaches by enriching them with knowledge gleaned from textual reviews [17]. These review-derived features, explicitly extracted, act as hidden clues for the recommendation algorithm. This enables them to harness the rich insights found in textual reviews, revealing the "why" behind user engagement with items beyond the binary "like/dislike" of ratings. This deeper understanding empowers the system to identify and exploit user preferences with increased accuracy thus ultimately boosting its performance [51].

Before integrating textual reviews in CF, meticulous preprocessing is essential, acting as the cornerstone of effective text mining and NLP. Review data often suffers from noise like HTML tags, hyperlinks, and irrelevant words. Removing these impediments not only saves computational resources but also ensures accurate processing downstream, paving the way for robust analysis techniques and ultimately enhancing the performance and outcome of CF models.

The value of textual reviews is reflected in the fact that they are the second most important element after the numerical ratings that the consumer will look at when inquiring about a specific item.

While other research explores more elaborate options, most focus on these essential text preprocessing operations. This thesis focuses on these common pre-processing steps:

1) Tokenization: this is the vital first step in NLP, and involves dissecting textual data into smaller, manageable pieces known as tokens. This study employed a simple yet effective strategy: splitting the text documents into words based on whitespace delimiters. While Python libraries offer diverse toolkits for tokenization, this approach served as a foundational step in the analysis [52].

2) Stop Words Removal: this is a crucial step in text pre-processing to eliminate frequently occurring words devoid of significant semantic meaning, like

articles, pronouns, and conjunctions. While their absence minimally impacts textual understanding, caution is necessary. Inspecting texts post-removal is crucial, as certain documents composed primarily of stop words might become empty, hindering further analysis [1].

3) Stemming or Lemmatization: is a crucial preprocessing step for many text mining applications. Two procedures are utilized interchangeably to convert words to their fundamental base by eliminating their suffixes. Stemming differs in that it simply removes suffixes regardless of meaning, potentially creating grammatically nonsensical forms. lemmatization aims to return words to their base form while preserving their semantic integrity, which reduces lexical ambiguity and supports effective analysis within the vast domain under study. In this thesis, the latter method is favored for usage over the former to uphold the semantic significance of words and curtail the expansive range of vocabulary [53].

4) Lowercasing: means converting all words in a text to lowercase. This eliminates the ambiguity of capitalization, ensuring that identical words with different cases are treated as the same entity during subsequent processing. This is crucial for tasks like text mining and information retrieval, where case variations can lead to inconsistencies and missed matches [52].

5) Additional text preprocessing operations: This stage involves multiple operations. Such as removing URLs (which have no contextual meaning and therefore can be confusing to the NLP model), in addition to Unicode normalization (which involves the process of handling the issue of equivalence and handling the emoji). Lastly, the Spelling Correction process.

After implementing text pre-processing as the first step for proposed review-based recommendation models, another important step is how to extract important aspects from text reviews. The extracted aspects will then be classified according to their polarity (positive or negative).

2.7 Opinion mining

The tidal wave of user-generated text flooding the internet demands powerful tools to unlock its hidden insights. Text mining techniques rise to the challenge of transforming messy, unstructured data into structured knowledge. This extracted treasure trove fuels diverse analyses, from exploring hidden patterns to predicting future trends [54].

Text mining (TM), powered by Machine Learning (ML), Deep Learning (DL), and other tools, unlocks valuable information from documents through a staged process. The initial and crucial step involves text analysis, where various techniques are iteratively applied to refine and extract relevant data [55]. While this approach focuses on structured and semi-structured sources like e-mails, textual reviews, social media posts, and HTML files, machine learning tools remain the gold standard for organizing and managing vast quantities of online data.

Opinion mining, often referred to as Sentiment Analysis (SA), is a rising star in research and leverages the power of advanced TM, ML, DL, and NLP to navigate the ocean of user-generated text [56]. This captivating field focuses on automatically deciphering the emotional undercurrents (positive or negative) within the subjective text, unlocking a wealth of insights from user opinions and expressions [57].

Opinion mining has grown in importance in business and marketing as companies seek to access customers' feedback to enhance their products and services. It can assist companies in determining customer satisfaction and improving brand reputation [25]. In addition, it can be used for customer service, market research, and political analysis [24].

2.7.1 Opinion Mining levels

In general, opinion mining methods are categorized according to the level of analysis into three levels [58], as illustrated in Figure 2.3 :

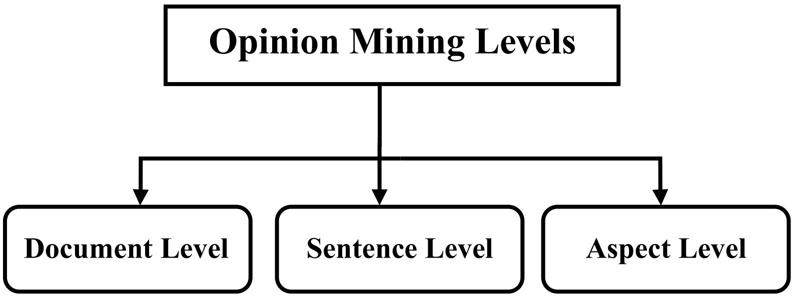


Figure 2.3: Opinion mining levels. [59]

1. Document level: At the document level, the goal is to identify the general sentiment of a text passage, such as a review or a social media post [16]. This can be done by analyzing the words and phrases in a text and then assigning the entirety of the document a sentiment score (either positive, negative, or neutral). Because this level of opinion mining does not go into specifics and the process of review is conducted from an abstract and general view, the mining process can be completed considerably more quickly [59].

2. Sentence level: At the sentence level, the objective is to determine the sentiment expressed in each sentence contained inside a document. This can offer a more thorough and nuanced perspective of the opinions stated in the text and assist in locating particular areas of positive or negative sentiment. Since the documents are divided into sentences, the classification of comments at the sentence level involves more difficulties than that at the document level as it gives more exact information on the polarity of the viewpoints [60].

3. Aspect level: At the aspect level of sentiment analysis (ABSA), the objective is to determine the sentiment expressed concerning particular features or aspects of an item, service, or experience. This includes extracting and obtaining sentiments and opinions expressed about particular aspects [61]. It provides a more granular and precise understanding of opinions and sentiments. Instead of just determining if a text is positive or negative, ABSA provides information on the specific aspects the user is interested in. For instance, in the process of buying a phone, one user may be most interested in the camera quality, while another may only care about the battery life. Depending on a 4-star rating for a specific phone, none of them may decide to buy it without reading additional reviews. Aspects, which are features or elements of an item, are typically mentioned in reviews to describe the item's quality [17]. For instance, "service", "food", and "staff" are some of the restaurants' aspects. Reviewers describe the quality of each aspect using sentiments which are mostly adjectives, such as "good service", "delicious food", and "uncooperative staff". These sentiments reflect the level of satisfaction customers have with the quality of each aspect [43].

In this thesis, aspect level has been relied upon in the sentiment analysis process, Therefore, the thesis delves beyond mere sentiment polarity in user reviews. salient features were extracted that enrich the information feeding into collaborative filtering algorithms. This fusion approach unlocks a deeper understanding of user preferences and enhances recommendation accuracy.

2.7.2 Opinion mining techniques

Opinion mining is another term for sentiment analysis, which is the process of identifying and extracting subjective information from textual data, such as social media posts, reviews, and news articles. It involves using NLP techniques to analyze the sentiment, emotion, and opinion expressed in the text [62].

Automatic sentiment recognition may be useful in various contexts and Applications. it has grown in importance in business and marketing as companies seek to access the customers' feedback to enhance their products and services. It can assist companies in determining customer satisfaction and improving brand reputation. In addition, it can be used for customer service, market research, and political analysis [63].

There are various techniques used for performing sentiment analysis, each with advantages and disadvantages. Below is an overview of the most common sentiment classification techniques.

- Lexicon-based techniques: (also known as knowledge-based techniques) are a common strategy in opinion mining. Lexicons are sets of tokens, each of which has a predefined score. In the lexicon-based approach, the document is initially divided into tokens of single words. The polarity of each token is then calculated, and the scores of each token are aggregated with positive, negative, and neutral scores being added independently. In the last stage, the overall polarity of the text is assigned according to the highest value of each score [60]. Lexicon-based techniques primarily involve two methods: the Dictionary-Based Method and Corpus-Based Method [41].

- Machine Learning techniques: These techniques make use of the power of algorithms to discover relationships and patterns in data automatically, enabling precise sentiment classification and opinion extraction. Machine learning algorithms used in sentiment analysis include Naive Bayes (NB), Support vector machine (SVM), Logistic regression (LR), Decision tree (DT), Maximum entropy (ME), K-nearest neighbors (KNN), and Semi-supervised learning [56]. In addition to Deep Learning algorithms which provide state-of-the-art performance in various opinion mining tasks. Deep learning methods, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown significant improvements in opinion mining. RNNs, like Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU), can detect contextual dependencies and sequential information in textual data [64].

- Hybrid techniques: Hybrid opinion mining refers to the combination of multiple techniques or approaches to improve the accuracy and effectiveness of sentiment analysis. Hybrid techniques combine the strengths of multiple algorithms or approaches, compensating the limitations of each other and producing more reliable and accurate results. For example, integrating Lexicon-based methods with Machine Learning, Rule-based approaches with Deep Learning, or Linguistic-based models with Topic Modeling. These techniques are crucial for dealing with the difficulties and complexity of sentiment analysis tasks, especially when handling noisy real-world text data [65].

2.7.3 Aspect Extraction

In tasks like sentiment analysis, opinion mining, and product review analysis, extracting aspects (the specific features or components) plays a crucial role. This process, known as aspect extraction, involves identifying and isolating these key elements from within the text, making it a fundamental step for further analysis and interpretation. There are several techniques used to extract aspects, which are classified in Figure 2.4.

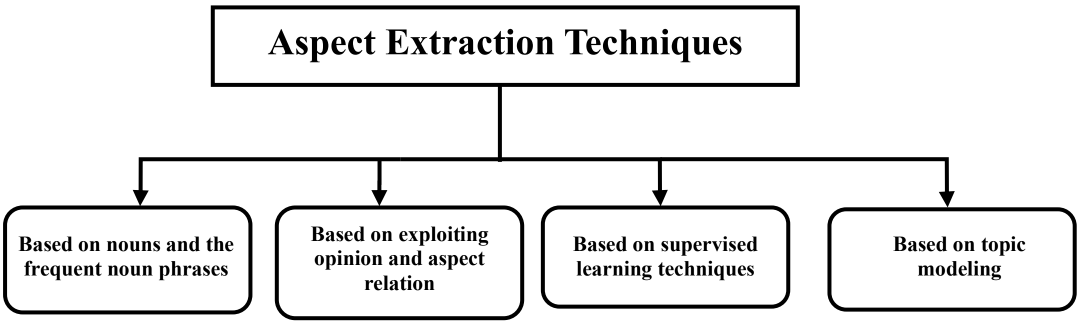


Figure 2.4: Aspect Extraction Techniques Classification. [60]

2.7.3.1 Extraction based on the frequency of noun phrases and nouns

This method extracts key topics based on the frequency of noun phrases and nouns is a well-established and efficient approach. This method leverages the tendency of people to repeat specific words and phrases when expressing their sentiments about various aspects of a product. Part-of-speech (POS) taggers identify nouns and noun phrases within the text, while the selection of prominent aspects relies on frequently occurring entities.

POS tagging constitutes a vital initial step in natural language processing (NLP). This fundamental task entails assigning grammatical categories (tags) to each word within a sentence, such as noun, verb, adjective, and adverb. These tags reflect the word's role and function within the sentence structure, thereby providing crucial information for subsequent NLP tasks. By comprehending the syntactic relationships between words, POS tagging facilitates downstream applications like parsing, sentiment analysis, and information extraction [66].

Consider the sentence “This food is delicious.” Applying POS tagging would attribute "The" as (DT) indicating its determiner function, "food" as (NN) signifying its noun status, "is" as (VBZ) denoting its present tense third-person singular verb form, and "delicious" as (ADJ) highlighting its role as an adjective describing the noun "food" and adding information about its quality. These tags elucidate the syntactic structure and the grammatical roles that each word plays in conveying the sentence's intended meaning.

There are several excellent tools and libraries for POS tagging, each with its strengths and weaknesses. spaCy is one of the most powerful and efficient NLP libraries in Python and excels in POS tagging due to its exceptional accuracy, ease of use, and pre-trained models for diverse languages. This open-source library transcends a mere tagger, serving as a versatile gateway to deeper text comprehension and manipulation within NLP tasks [67].

- **Extraction based on topic modeling**

Within the field of natural language processing, statistical topic models have emerged as a robust and systematic approach for uncovering latent thematic structures within text document collections. Operating in an unsupervised manner, these models posit that each document harbors k underlying topical clusters. For example in the case of restaurant investigation, where reviews routinely discuss standard features like location, cleanliness, and service. In such situations, the ability to automatically extract relevant aspects without human annotation becomes paramount. Statistical topic models, with their capacity to identify hidden thematic groupings, offer a compelling solution for this challenge.

Since this approach, uses statistical methods like latent semantic analysis (LSA), latent Dirichlet allocation (LDA), and Bidirectional Encoder Representations from Transformers (BERT) it is called a statistical model too. In addition, these models use the bag of words represented in documents, so they can be used only in document-level opinion mining.

BERT is one of the aspect extraction techniques based on topic modeling that has revolutionized various natural language processing tasks, including aspect extraction because of its ability to analyze words in their surrounding context, thanks to its bidirectional architecture, allows it to capture subtle nuances and relationships between words related to different aspects. BERT can be combined with other NLP techniques like POS tagging and dependency parsing to refine aspect extraction. This comprehensive approach allows for better identification of specific opinionated phrases and their association with particular aspects.

BERTopic is a topic modeling technique that utilizes BERT embeddings for clustering and topic modeling. While BERTopic is primarily used for topic modeling, it can also be employed for aspect extraction by interpreting the identified topics as aspects within the text data [68].

Overall, BERTopic offers a powerful toolset for enriching and supporting aspect extraction. By leveraging its strengths in topic identification, feature engineering, and interpretability, it can be valuable insights into the key aspects discussed in the text data and inform further analysis or model development.

2.8 Deep Learning

Deep learning leverages the power of layered artificial neural networks, often called neural networks for short, to tackle complex learning tasks [69]. This approach unlocks the vast potential of these networks, once limited to simpler tasks with few layers and minimal data.

Taking inspiration from the intricate architecture of the biological brain, neural networks are built upon a web of information processing units, known as neurons, intricately arranged in layers [70]. These interconnected neurons work in harmonious concert, mimicking the brain's learning process by constantly adjusting their connection strengths, ultimately mastering diverse tasks like classification.

Neural networks, akin to intricate webs of interconnected neurons, harness the power of numerous layers to process and categorize data [70]. An input layer, a hidden layer(s), and an output layer(s) are all present. Each layer consists of nodes, and the weight of each node is taken into account as data is processed and passed on to the next layer. Input data feeds into this layered maze, navigating through nodes at each layer, whose weighted connections dictate the flow and transformation of information. Hidden layers, the network's engine, perform complex computations, ultimately culminating in predictive outputs generated by the final layer.

The past decade has witnessed a breathtaking rise in deep learning, generating cutting-edge results across diverse applications. It began with computer vision, then conquered speech recognition, and most recently, stormed the gates of NLP [71]. This renaissance finds its roots in a potent confluence of factors: unparalleled computing power fueled by Graphics Processing Unit (GPU) advancements, oceans of training data available for hungry algorithms, and the inherent strengths of deep networks in extracting and leveraging rich intermediate representations.

Despite its remarkable achievements, deep learning's progress hinges on a weak point which is the data. The quantity and quality of training data pose significant limitations, as the amount needed for sufficient and reliable training remains a complex puzzle. This intricate calculus depends not only on the sheer volume of input data but also on its internal quality and the inherent complexity of the task at hand. Typically, forging an accurate and generalizable model necessitates thousands of training examples, though this figure can fluctuate drastically.

While thousands of training examples typically pave the way for accurate and generalizable models, bigger isn't always better in the world of deep learning. Low-quality, mislabeled, or biased data, even in vast quantities, can lead models astray, resulting in poor performance in real-world applications.

Deep learning models, while potent, encounter the "black box" conundrum: they produce outputs from inputs, yet their internal decision-making processes remain opaque. Simpler linear algorithms, though sometimes less powerful, offer lucid interpretability. Upon training completion, they reveal feature weights, enabling transparent model comprehension and potential identification of crucial predictors, a boon for addressing black box concerns [72].

To illuminate the inner workings of deep learning models and address the "black box" conundrum, activation maps, or heatmaps, emerge as a valuable tool. These visual aids unveil the image regions that most strongly contribute to the model's output classification.

2.8.1 Activation functions

Activation functions are the most important components of Artificial Neural Networks, used to perform nonlinear mappings of the input data, and are typically applied element-wise to all neurons in a hidden layer. The activation functions are commonly used in the intermediate layers of Neural Networks [73].

Without the magic touch of an activation function, a neural network stumbles, stuck in a one-dimensional world of straight lines and acting as a linear Regression Model with limited performance and power. While linear equations hold their charm, their simplicity confines them to predictable patterns, unable to unearth the hidden gems of complexity within data [74].

While neural networks excel at learning and computing linear relationships, the real world throws us far more intricate curveballs. From feature extraction related to images and videos to deciphering the intricacies of written text and spoken language and text analysis, success demands venturing beyond the confines of linearity. To truly unlock the potential of these powerful models, we must equip them with the tools to navigate the boundless complexities of diverse data forms.

For this purpose, activation functions and artificial neural network methodologies were used such as Deep Learning, in which the model has many hidden layers and sophisticated architecture. This enables the model to make sense of complex, high-dimensional, nonlinear datasets. There are several activation functions, but only Sigmoid function is explained briefly.

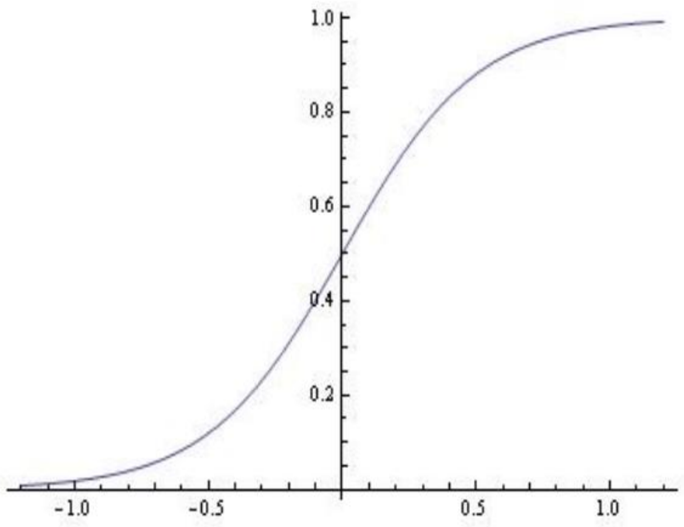


Figure 2.5: Sigmoid Activation Function. [75]

The ubiquitous Sigmoid Activation Function, which is also known as the "logistic function" or "squashing function," reigns supreme among non-linear activation functions [73]. As shown in Figure 2.5, Sigmoid Activation Function takes a real number as input and outputs a value between 0 and 1. The Sigmoid function is used to map input values (represented by real numbers) onto the curve that spans from 0 to 1 [76], It is possible to describe it as:

$$\boldsymbol{F(x) = 1/1 + e^{-x}} \qquad (2.2)$$

Where $\boldsymbol{x}$ represents the input value.

2.8.2 Optimization algorithm

In the world of deep learning, optimization algorithms are presented as the crucial tools feeding a machine's ability to extract knowledge from its interactions with the world. These algorithms calculate gradients that guide them toward minimizing the distance between predicted and actual outcomes thus reducing the loss function to its smallest value. This principle paves the way for a multitude of learning strategies, each wielding use optimization tools to navigate the path to knowledge acquisition [77]. Many optimization algorithms are used in the field of deep learning, such as Adam Optimizer.

Adam optimizer, an abbreviation for the Adaptive Moment Estimation optimizer, represents a commonly employed optimization algorithm within deep learning. Serving as an extension of the stochastic gradient descent (SGD) method, it focuses on adjusting the neural network's weights throughout the training process.

Adam optimizer holds numerous advantages, leading to its widespread utilization. It stands established as a benchmark in deep learning research papers and is advocated as the default optimization algorithm. Furthermore, its uncomplicated implementation, renders a faster execution, demanding minimal memory, and necessitating less fine-tuning compared to alternative optimization algorithms [78].

Algorithm 2.1: Adam Algorithm.

Require: Step size ϵ (Suggested default: 0.001)
Require: Exponential decay rates for moment estimates, ρ_1 and ρ_2 in $[0, 1)$. (Suggested defaults: 0.9 and 0.999 respectively)
Require: Small constant δ used for numerical stabilization (Suggested default: 10^{-8})
Require: Initial parameters $\boldsymbol{\theta}$
Initialize 1st and 2nd moment variables $\boldsymbol{s} = \mathbf{0}$, $\boldsymbol{r} = \mathbf{0}$
Initialize time step $t = 0$
while stopping criterion not met **do**
Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
Compute gradient: $\boldsymbol{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
$t \leftarrow t + 1$
Update biased first moment estimate: $\boldsymbol{s} \leftarrow \rho_1 \boldsymbol{s} + (1 - \rho_1)\boldsymbol{g}$
Update biased second moment estimate: $\boldsymbol{r} \leftarrow \rho_2 \boldsymbol{r} + (1 - \rho_2)\boldsymbol{g} \odot \boldsymbol{g}$
Correct bias in first moment: $\hat{\boldsymbol{s}} \leftarrow \frac{\boldsymbol{s}}{1 - \rho_1^t}$
Correct bias in second moment: $\hat{\boldsymbol{r}} \leftarrow \frac{\boldsymbol{r}}{1 - \rho_2^t}$
Compute update: $\Delta \boldsymbol{\theta} = -\epsilon \frac{\hat{\boldsymbol{s}}}{\sqrt{\hat{\boldsymbol{r}}} + \delta}$ (operations applied element-wise)
Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta \boldsymbol{\theta}$
end while

2.8.3 Long Short-Term Memory (LSTM)

In domains involving sequential data like text, audio, and video, RNNs reign supreme. However, for larger input gaps, standard RNNs with sigmoid or tanh cells stumble in capturing relevant information. Enter the Long Short-Term Memory (LSTM) network, an innovator characterized by incorporating Gate functions into the cell structure. LSTM has become a state-of-the-art tool for processing various sequential and temporal data [79].

LSTM was proposed by Hoch Reiter and Schmid Huber in 1997 and was refined and popularized by many people [80]. The key strength of LSTM lies in its ability to manage and remember information over extended sequences, mitigating the vanishing gradient problem often encountered in traditional RNNs. This is accomplished through a sophisticated gating mechanism that includes input, forget, and output gates [81]. These gates regulate the flow of information within the network, allowing it to selectively retain or forget information as needed.

LSTMs reign supreme in the deep learning realm, eclipsing most other RNN approaches with their powerful learning capabilities. Their impact stretches across diverse domains, from speech recognition and acoustic modeling to trajectory prediction, language translation, text generation, correlation analysis, and sentiment analysis. This prowess stems from their intricate structure, where recurrent layers harbor cells whose states, intricately influenced by past information and current input, form the backbone of their success [79].

LSTM units come in various architectures, but a typical design centers around a memory cell and three regulatory gates: input, output, and forget. These gates control the flow of information within the unit, allowing it to selectively remember, process, and transmit data. Some variations deviate from this standard, omitting gates or adding new ones, tailoring their memory management for specific tasks.

Based on the connections shown in Figure 2.6, The mathematical expressions of the LSTM can be written as follows

$$f_t = \sigma\left(w_{fh}h_{t-1} + w_{fx}x_t + b_f\right) \qquad (2.3)$$

$$i_t = \sigma(w_{ih}h_{t-1} + w_{ix}x_t + b_i) \tag{2.4}$$

$$\widetilde{C}_t = tanh(w_{\tilde{c}h}h_{t-1} + w_{\tilde{c}x}x_t + b_{\tilde{c}}) \tag{2.5}$$

$$c_t = f_t.c_{t-1} + i_t.\widetilde{c}_t \tag{2.6}$$

$$o_t = \sigma(w_{oh}h_{t-1} + w_{ox}x_t + b_o) \tag{2.7}$$

$$h_t = o_t.tanh(c_t) \tag{2.8}$$

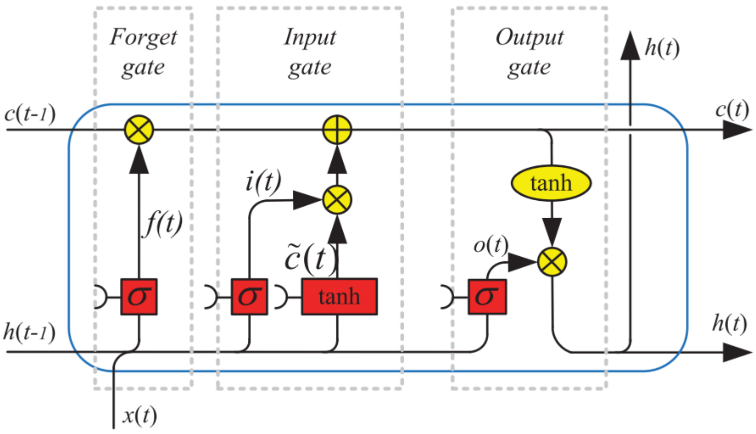


Figure 2.6: Architecture of LSTM. [82]

Where f_t represents the value of the forget gate which can decide what information will be thrown away from the cell state. When f_t is 1, it keeps this information. In contrast, a value of 0 means it gets rid of all the information. c_t denotes the cell state of LSTM. W_i, $W_{\tilde{c}}$, and W_o are the weights, and the operator $(\cdot)$ denotes the pointwise multiplication of two vectors. b is the bias. b_f is the bias of the forget gate, b_i is the bias of the input gate, b_o is the bias of the output gate, and $b_{\tilde{c}}$ is cell state bias.

x_t , h_t denote the input, the recurrent information, and the output of the cell at time t, respectively.

2.9 Evaluation Metrics

There are several evaluation metrics for different kinds of applications. In this chapter, the evaluation metrics (that were used in this thesis) are explained for each application separately. Section 2.9.1 describes the Confusion Matrix. Section 2.9.2 illustrates Accuracy. Section 2.9.3 highlights the Cross-Validation method. Finally, sections 2.9.4 and 2.9.5 present two metrics used to evaluate the accuracy of rating prediction models, which are MAE and RMSE.

2.9.1 Confusion Matrix

The evaluation process is an integral part of any model. Categorization models (or classifiers) are one of these models that require a performance evaluation process to make informed decisions about model selection and optimization [25]. In this context, the confusion matrix emerges as a pivotal tool for illuminating model performance and directing subsequent refinement efforts [66].

This matrix, a tabular representation, juxtaposes predicted classifications against actual outcomes, thereby facilitating a comprehensive appraisal of the model's efficacy.

Figure 2.7 illustrates the binary classification in the confusion matrix which serves as a structured framework for evaluation. This matrix encompasses four distinct sets, each representing a unique intersection of actual and predicted values: True Positive, True Negative, False Positive, and False Negative [83]

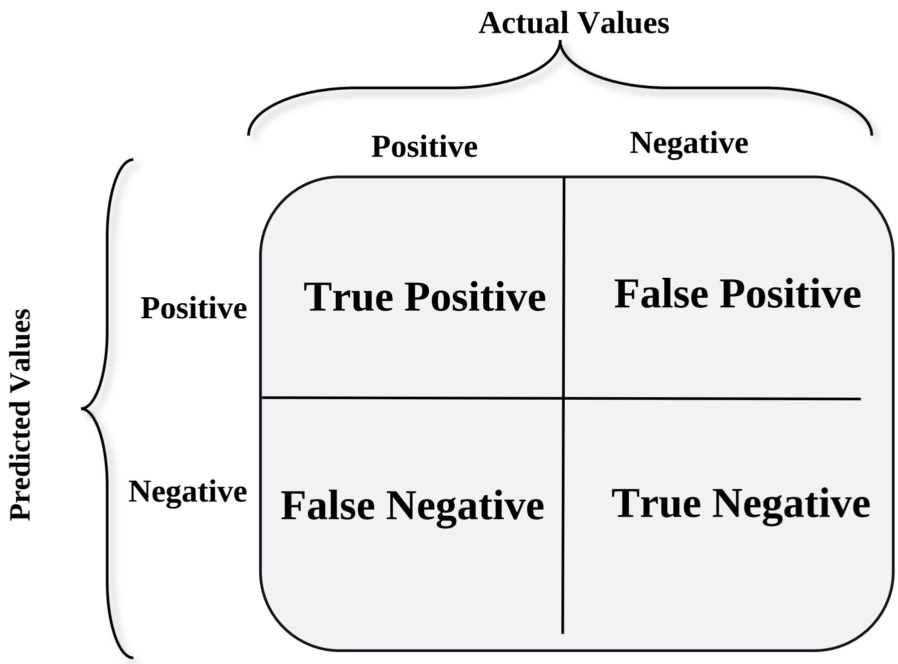


Figure 2.7: Confusion matrix

In the following, we embark on a meticulous dissection of each term within the confusion matrix, elucidating their meanings and offering brief explanations:

True Positive (TP): In the confusion matrix, a data point is deemed True Positive when the expected positive outcome finds matching in the actual outcome.

False Positive (FP): Within the framework of the confusion matrix, a data point is designated as a false positive when the model erroneously forecasts a positive outcome, yet the actual outcome proves to be negative. This is classified as a (Type 1 Error).

False Negative (FN): Within the domain of the confusion matrix, a data point is assigned as a false negative when a negative outcome is predicted, whereas a positive outcome happens. This misalignment is formally categorized as a (Type 2 Error).

True Negative (TN): a data point is classified as True Negative (TN) in a confusion matrix when the model's prediction of a negative outcome aligns with the observed reality of an actual negative outcome.

2.9.2 Accuracy

The performance of the initial sentiment analysis stage within the proposed model was assessed utilizing the accuracy metric, the most used metric in all the classification tasks. Accuracy, defined as the degree to which predicted values coincide with standard values [66], was calculated as the sum of both True Positives and True Negatives (TP + TN) divided by the total dataset size (P + N). A value of 1 signifies perfect accuracy, while 0 represents the worst possible outcome, as captured by the following equation:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\mathbf{TP+TN+FP+FN}} \qquad (2.9)$$

Where $\boldsymbol{TP}$ is the true positive, $\boldsymbol{TN}$ is the true negative, **FP** is the false positive, and **FN** is the false negative.

2.9.3 Cross-Validation

To ensure the accuracy of the sentiment analysis model, many researchers suggest the use of the cross-validation method. Cross-validation is a model evaluation parameter that demonstrates the ability of the system to make new predictions accurately. It promotes a cyclical approach to evaluate model performance on unseen data. The available dataset is split into several subsets, known as folds. Each fold, in turn, serves as the validation set while the remaining folds constitute the training set. This procedure is repeated iteratively, with each fold taking on the validation role once, and the averaged performance metrics across all folds provide a robust assessment of the model's generalizability.

There are many types of cross-validation, and k-fold is one of the most popular types used to conduct the cross-validation process to assess the performance of the model. In K-fold cross-validation, the dataset is divided into k subsets, namely folds which are repeated k times. For every iteration, The model is trained on K-1 folds and validated on the remaining fold, so that each fold serves as the validation set exactly once. The final performance metric is the average of the metrics from all K folds.

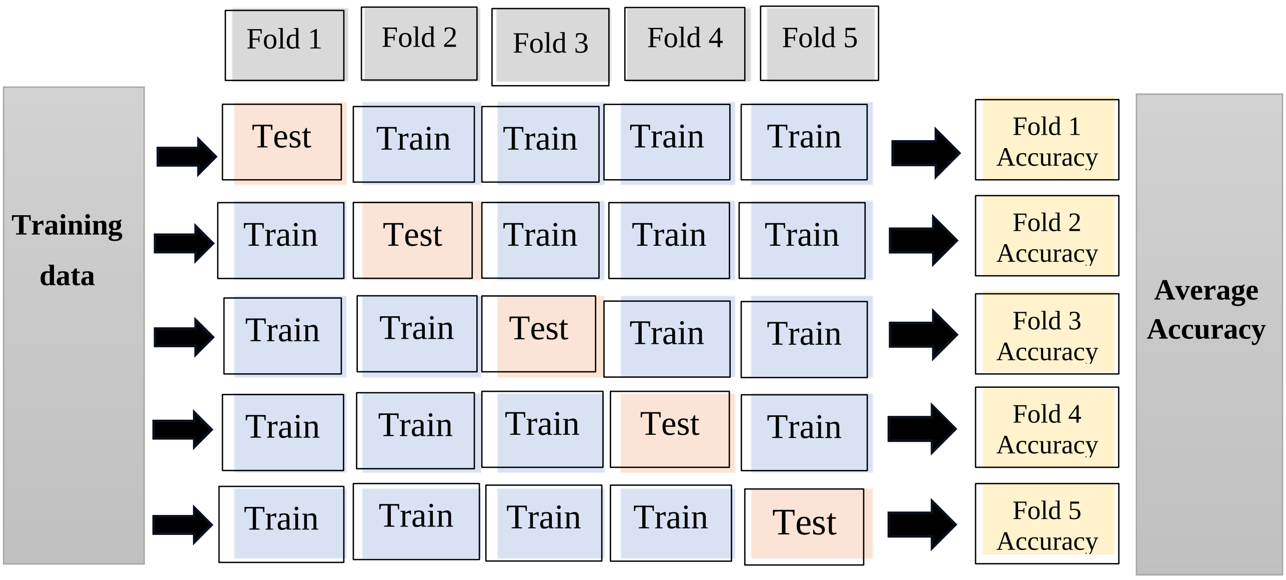


Figure 2.8: K-fold cross-validation procedure

2.9.4 Mean Absolute Error (MAE)

The ultimate aspiration of any RS is to deliver top-notch recommendations. To verify the success, a critical step is performance evaluation [6]. This allows the proposed system to be compared with established benchmarks and to measure its effectiveness in providing accurate and satisfying recommendations.

The vast landscape of RS research encompasses two main tasks: predicting missing ratings and recommending top-N items for users [84]. While the former focuses on minimizing error through accurate rating estimation of empty cells, the latter seeks to identify and suggest the most appealing items for the active user. Each task demands distinct evaluation metrics to scrutinize the performance of proposed systems.

In the process of rating prediction, two metrics reign supreme: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The smaller the value, the better the performance. MAE acts as a steadfast gauge of a system's predictability, quantifying the extent of its error a stark reflection of its accuracy's inverse. In essence, this metric meticulously calculates the average absolute deviation between predicted ratings and their corresponding authentic counterparts within the test set [85], as shown in the equation below:

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n} \left| p_{u,i} - r_{u,i} \right|} \qquad (2.10)$$

Where, $\boldsymbol{p_{u,i}}$ is the predicted rating of the user $\boldsymbol{u}$ to item $\boldsymbol{i}$, $\boldsymbol{r_{u,i}}$ is the real rating, and $\boldsymbol{n}$ is the number of all ratings in the test set.

2.9.5 Root Mean Square Error (RMSE)

RMSE differs from MAE in that it represents the sample standard deviation of the differences between predicted and actual ratings. It calculates the square of the error between the expected value and its actual counterpart and then computes the square root of the output [86]. The RMSE is calculated as shown in equation 2.11.

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (p_{u,i} - r_{u,i})^2}{n}} \qquad (2.11)$$

Where, $\boldsymbol{p_{u,i}}$ is the predicted rating of the user $\boldsymbol{u}$ to item $\boldsymbol{i}$, $\boldsymbol{r_{u,i}}$ is the real rating, and $\boldsymbol{n}$ is the number of all ratings in the test set.

CHAPTER THREE

PROPOSED METHODOLOGY

3.1 Overview

This chapter presents the architecture of the proposed system, which consists of two tasks: alleviating the user cold start problem and addressing the sparsity problem to enhance the prediction accuracy of the proposed RS model.

3.2 The Main Architecture of the Proposed RS-TRDL Model

In general, the objective of our proposed RS-TRDL model is to develop a hybrid system by fusing recommender systems and sentiment analysis ones. It predicts the ratings by employing the polarity of the aspects extracted from user textual reviews as additional information to alleviate the cold start and sparsity problem and to improve the performance of the recommender system. This section will present the system's major phases and components. Figure 3.1 illustrates the entire RS-TRDL architecture.

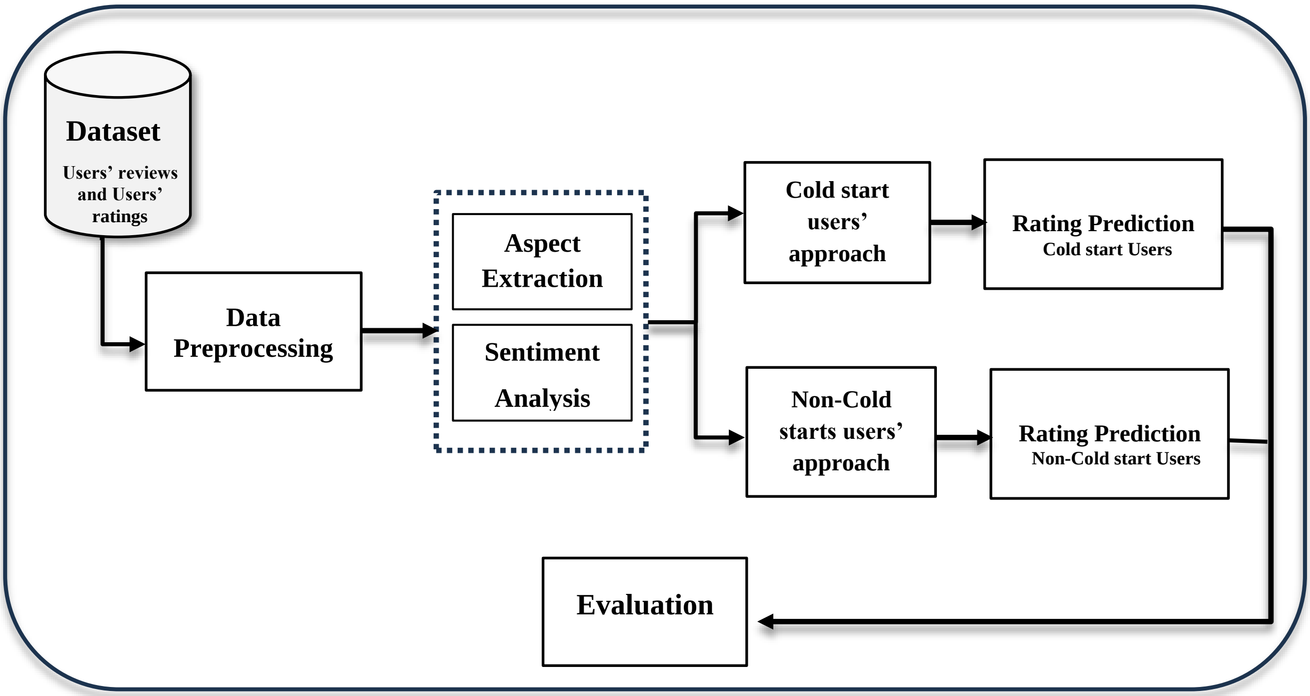


Figure 3.1: RS-TRDL architecture.

The proposed model is depicted in detail with the following steps:

3.2.1 Data Preprocessing

In general, NLP projects consist of a modeling phase in which the model is trained so that it may be used with real-world data. This model requires numerical data, but the data that was used is textual, therefore it must go through several steps of text preprocessing. In this work, two stages of pre-processing were applied, the first related to Dataset in general and the other related to texts represented by user reviews.

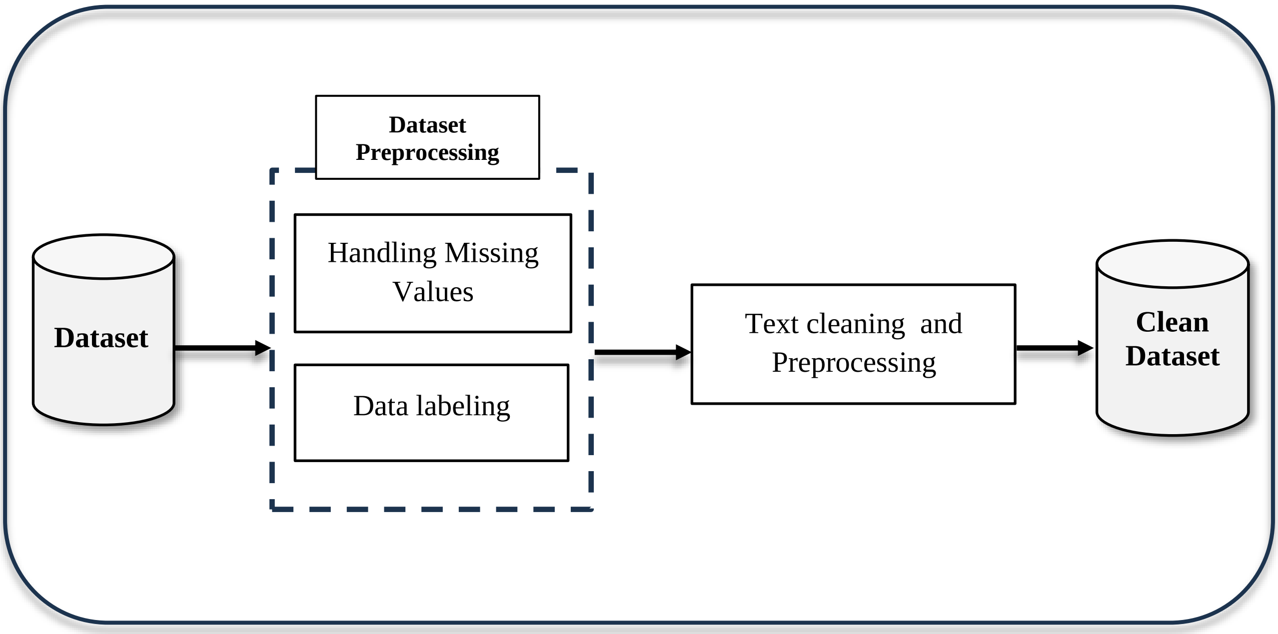


Figure 3.2: Data Preprocessing

Figure 3.2 shows the pre-processing mechanism that was relied upon in this work.

- The preprocessing stage is related to the dataset presented by two steps which are Handling Missing Values and the data labeling step.

1. **Handling Missing Values step** targets users who do not have any information (rating, text review, etc.) as they cannot be helped or benefited from. These can be handled using lots of techniques. In this work, the rows having missing values in the ratings or reviews columns are dropped.

1	marketplace	customer_id	review_id	product_id	product_title	product_categ	star_rating	helpful_vote	review_body	review_date
2	US	41409413	R2MTG1GCZLR2DK	B00428R89M	yoomall 5M Antenna WIFI RP-SMA Female to M	Electronics	5	0	As described.	8/31/2015
3	US	49668221	R2HBOEM8LE9928	B000068O48	Hosa GPM-103 3.5mm TRS to 1/4" TRS Adaptor	Electronics	5	0	It works as advertising.	8/31/2015
4	US	12338275		B000GGKOG8	Channel Master Titan 2 Antenna Preamplifier	Electronics		1		
5	US	38487968	R1EBPM82ENI67M	B000NU4OTA	LIMTECH Wall charger + USB Hotsync & Chargin	Electronics	1	0	Did not work at all.	8/31/2015
6	US	23732619	R372S58V6D11AT	B00JOQIO6S	Skullcandy Air Raid Portable Bluetooth Speake	Electronics	5	1	Works well. Bass is somewhat lacking but is present. Overall pleased with the i	8/31/2015
7	US	21257820	R1A4514XOYI1PD	B008NCD2LG	Pioneer SP-BS22-LR Andrew Jones Designed Bo	Electronics	5	1	The quality on these speakers is insanely good and doesn't sound muddy when	8/31/2015
8	US	3084991		B00007FGUF	C2G/Cables to Go 03170 3.5mm F/F Stereo Cou	Electronics	1	0		
9	US	8153674	R1WUTD8MVSROJU	B00M9V2RMM	COOLEAD-HDMI Switcher BOX	Electronics	5	0	works great	8/31/2015
10	US	52246189	R1QCYLT25812DM	B00J3O9DYI	Philips Wireless Portable Speaker	Electronics	4	0	Great sound and compact. Battery life seems good. Happy with this product.	8/31/2015
11	US	41463864	R904DQPBCEM7A	B00NS1A0E4	PlayStation 3 3D Glasses (Super Value 4 Pack)	Electronics	4	0	It works well~~~	8/31/2015
12	US	2781942	R1DGA6UQIVLKZ7	B007B5V092	JVC HAFR201A Xtreme Xplosive Deep Bass Eart	Electronics	5	0	Alll good	8/31/2015
13	US	707292	RLQT3V8SMNIBH	B00IODHGVG	Sylvania Alarm Clock Radio with CD Player and	Electronics	5	0	Love clock radio & CD player. Easy to operate.	8/31/2015
14	US	31463514	R3T9GZS2TMXZGM	B0035PBHX6	Coby 8 GB 1.8-Inch Video MP3 Player with FM F	Electronics	1	0	Breaks very easily, and takes a while to load music	8/31/2015
15	US	33475055	R24HVAEYP5PLDN	B00K1JJWFO	Diamond (Original) SRH77CA 144/440 MHz. Dua	Electronics	5	0	Excellent gain in radio frequency reception over the stock antenna that came w	8/31/2015
16	US	16543871	R32KMAPNV5NJPJ	B00S3LJ5EA	Kingvom 8gb 50 Hours Continuous Playback Mp	Electronics	5	0	everything I expected for a great price	8/31/2015
17	US	38472651	RC7VLPHUT6UAF	B00B5QNGN6	JBL Ultra-Portable Speaker with Built-In Bass P	Electronics	5	0	Love this small speaker with loud volume, great for the beach	8/31/2015

Figure 3.3: Sample of dataset

For example, in Figure 3.3, two users have no textual review information (Costumer_id = 12338275 and Costumer_id = 3084991). Those two users had been dropped in the Handling Missing Values step.

2. **Data labeling** is the process of assigning relevant and informative tags or labels to raw data (such as text, images, audio, or video) to make it accessible and useable by machine learning algorithms. In this work, the data was labeled manually based on textual reviews and numerical ratings without using any algorithm. The user's data with a rating greater than 3 was assigned as positive and the remainder which had a rating less than 3 as negative reviews. As for the rating value of 3, we relied on the classification

of this user's text review to be labeled based on it. The TextBlob Python library was used to extract the sentiment of the user's textual reviews (positive or negative reviews). If the sentiment of the textual review is positive, then the data point is labeled as positive and vice versa. This stage is necessary to perform binary classification of text reviews.

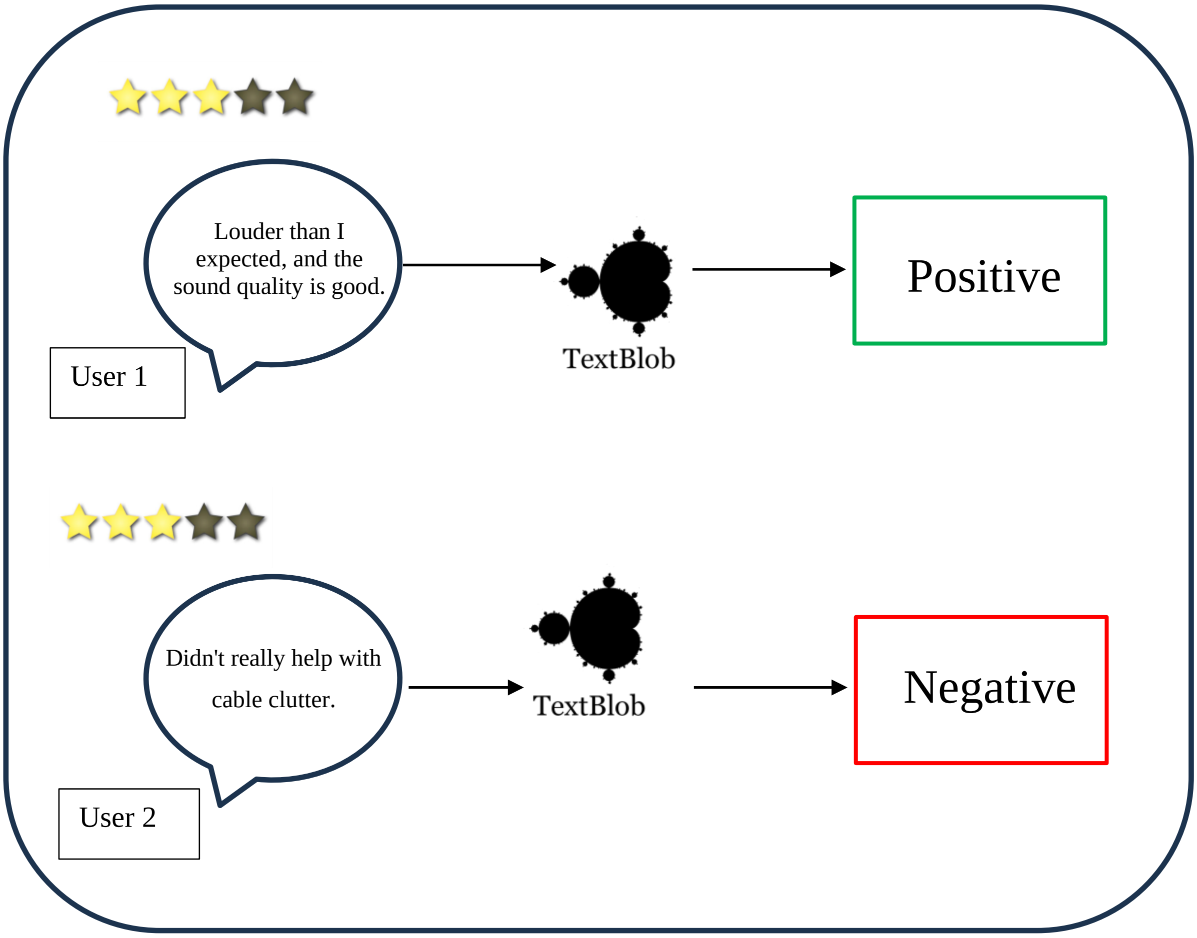


Figure 3.4: Data Labeling step

For example, in Figure 3.4, Following a rating of 3 assigned by the first user, their accompanying textual review was analyzed to determine its sentiment. This analysis was conducted using the Python library a TextBlob, which was employed to extract the polarity (positive or negative sentiment) from the review content. Based on the extracted polarity, the review was subsequently labeled as “positive”.

Conversely, the second user also assigned a rating of 3 to the product. However, the sentiment analysis of their written review revealed negative feelings, leading to the labeling of "negative".

- The next preprocessing stage is related to the textual data. This stage includes several steps:

  1. **Text cleaning**, that is, the process of extracting the raw text from the input data and converting it to the desired encoding format by eliminating all non-textual elements like markups and metadata. This step involves three operations:

     - First, removing URLs, which have no contextual meaning and therefore can be confusing to the NLP model.
     - Second, Unicode Normalization (which involves the process of handling the issue of equivalence and handling the emoji).
     - Third, Spelling Correction (using pyspellchecker a Python library).

Figure 3.5 presents a clear illustration of the aforementioned text-cleaning procedures through a simplified example. This figure employs a single sample review to demonstrate the application of each text-cleaning step.

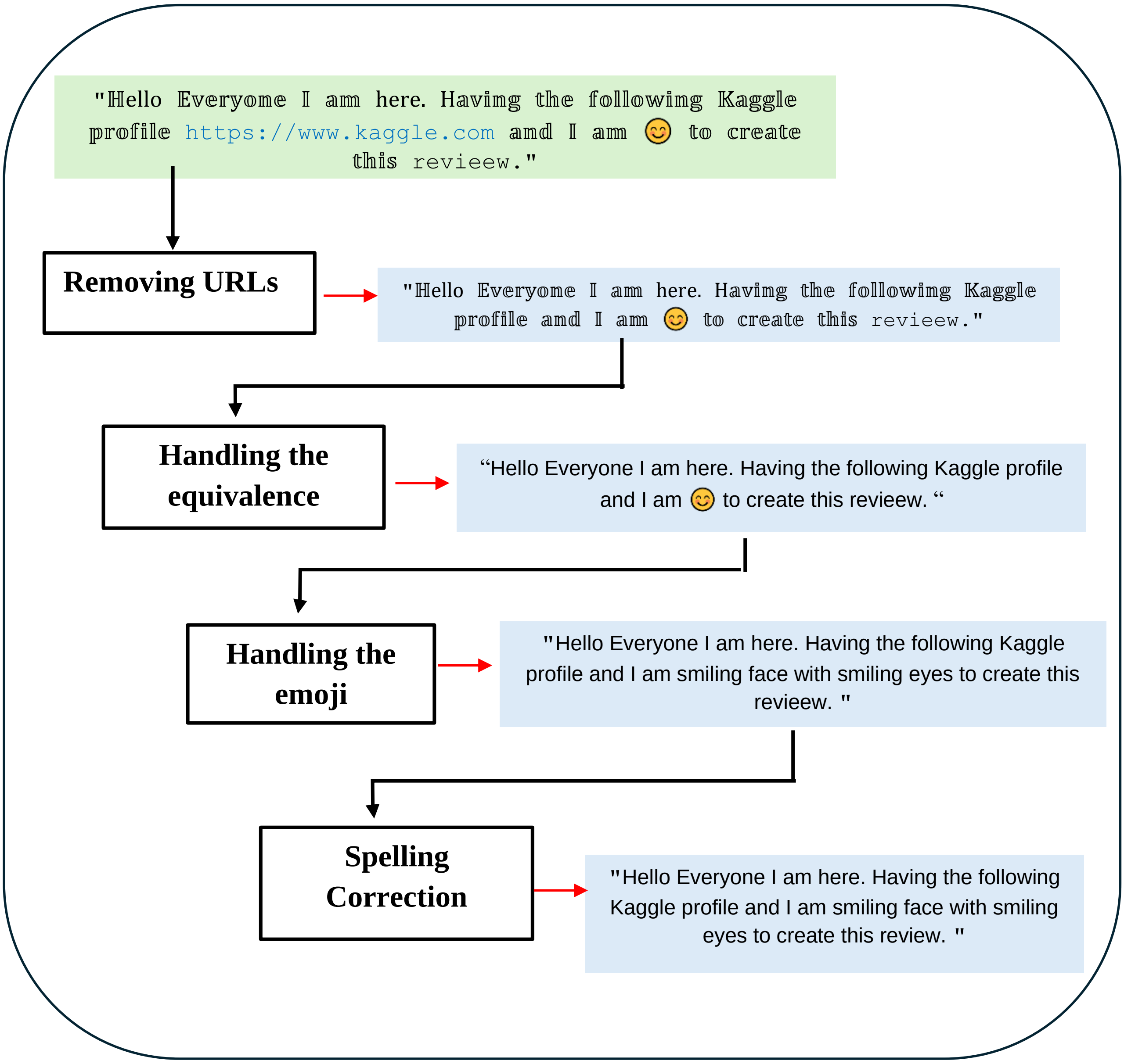


Figure 3.5 Text cleaning

2. Later, there is the Text Pre-Processing step which includes several tasks (Tokenization, Removing Stop Words, Lowercasing, Stemming, and Lemmatization).

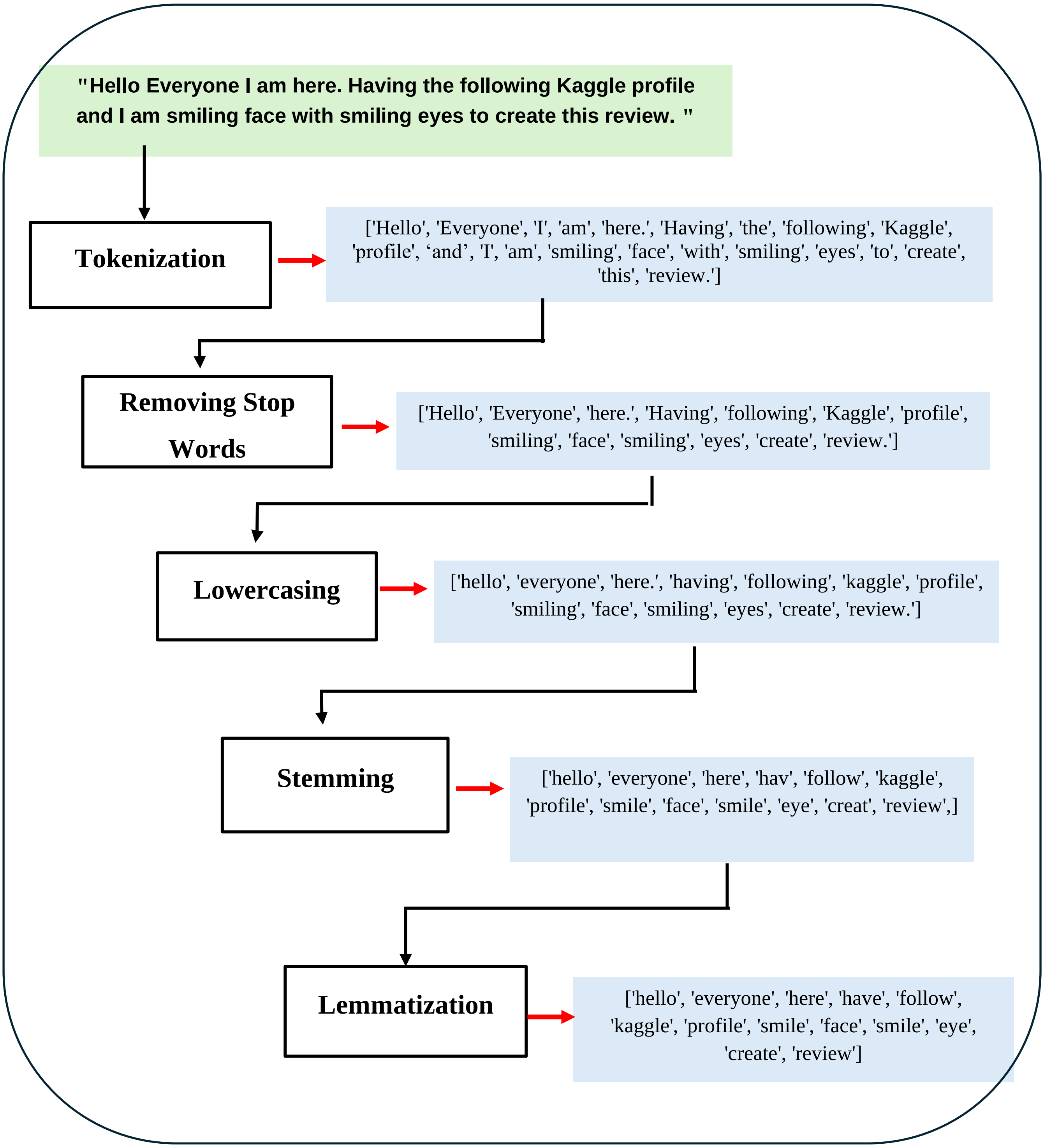


Figure 3.6: Text Preprocessing

Figure 3.6 illustrates a practical application of the text pre-processing steps. The output from the example presented in Figure 3.5 serves as the input for these pre-processing steps. The figure explicitly demonstrates the output generated at each step of the process.

3.2.2 Aspect Extraction

This phase incorporates three operations, namely, Segmentation, Noun Extraction, and Topic Modeling. Figure 3.7 illustrates those operations.

- **Segmentation** is the process of dividing a continuous stream of text into meaningful sentences or segments. The textual review is divided into sentences, and each sentence proceeded separately.
- **Noun Extraction**, given the search for the user's significant aspects (usually expressed as nouns in textual user reviews), the words that had the noun tag were extracted using POS tagging task which extracts nouns from texts. (As explained in 2.7.3).
- **Topic Modeling**, because the total number of these nouns may be quite large, unrelated nouns are filtered using Bidirectional Encoder Representations from Transformers Topic (BER Topic). BERTopic is an open-source library that uses a BERT model to do Topic Detection with a class-based TF-IDF procedure.

3.2.3 Sentiment Analysis

The sentiment Analysis phase works parallel with the aspect Extraction process. In this phase, we need to extract and identify the polarity of the extracted aspects based on the sentiment of the sentence in which they appear. To do this process accurately, we utilized the LSTM algorithm, which is explained in section 2.8.3 to build the sentiment classifier. Initially, the dataset was divided into subsets for training and validation.

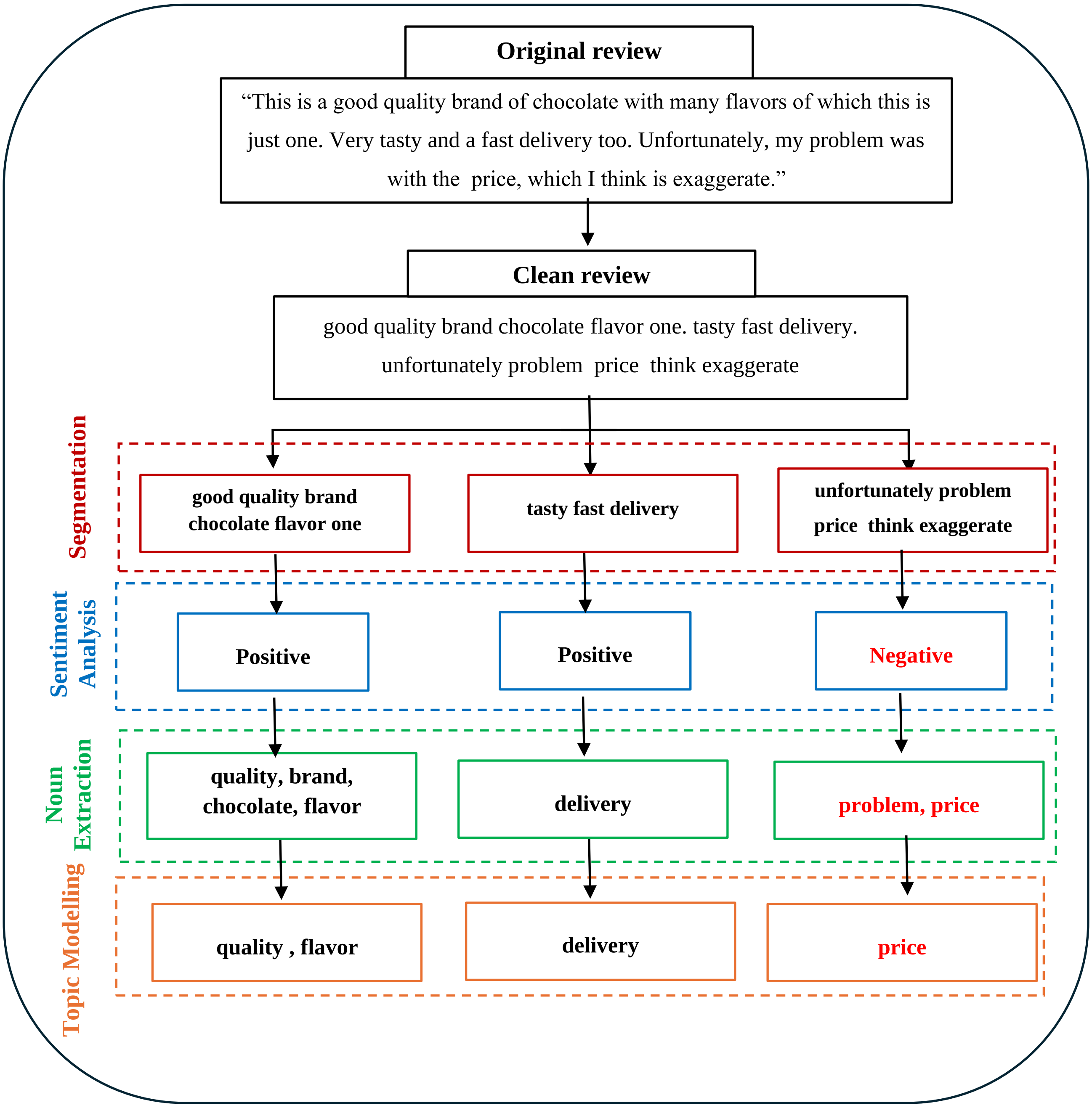


Figure 3.7: Aspect Extraction and Sentiment Analysis Processes.

As well known in the basics of deep learning, the model was trained based on the labeled dataset to learn and train the classifier. Then, this model was used in classifying the extracted aspects by classifying the segmented sentences from the original review. When the classifier determined the polarity of the sentence toward positivity, all aspects that were extracted from this sentence were classified as positive aspects. Conversely, if a sentence is classified as negative, then all aspects that were extracted from this sentence will be classified as negative aspects.

Figure 3.7 comprehensively illustrates the parallel execution of aspect extraction and sentiment analysis processes. These processes were implemented on the provided user review, resulting in the identification of three positive aspects (quality, flavor, delivery) and one negative aspect (price).

3.2.4 Cold start users' approach

To deal with the cold start problem, the cold start users should be selected. Therefore, the users have been divided into two groups based on their rating history. The first group comprises users who have provided ratings exceeding a predefined threshold (namely Non-Cold Start group). Conversely, the second group, often referred to as the (Cold Start group), consists of users with a rating history equal to or less than the established threshold. Extensive experimentation and data analysis led to the selection of a threshold value of 5 for identifying cold-start users. So, each group was split with its positive and negative extracted aspects (cold start group with its numerical ratings and extracted aspects, non-cold start group with its numerical ratings and extracted aspects). Figure 3.8 shows the approach of cold start users.

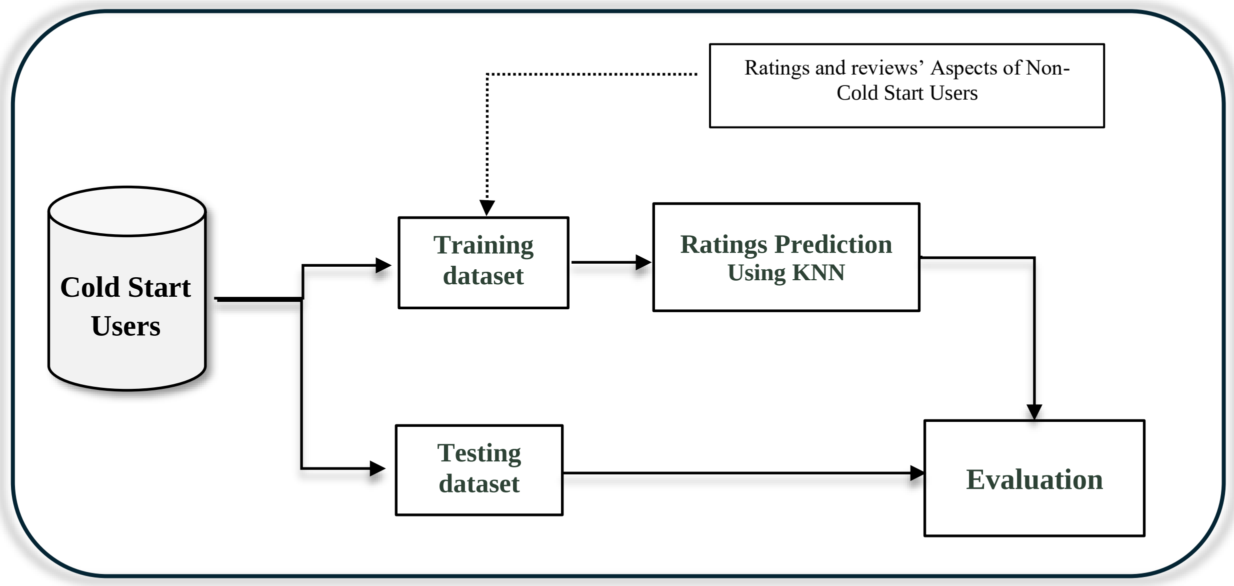


Figure 3.8: Cold start users' approach.

3.2.4.1 Train-Test Splitting

After selecting the cold start user group, the process of train-test data splitting was done to assess the performance of the trained model. In our work, the cold start group dataset is split into 2 sub-datasets. 80 percent of the entire data is used for training and 20 percent of it for testing.

3.2.4.2 Rating prediction For Cold start Users

For each new user of the cold start group in the training dataset, the ratings have been predicted depending on the similarity in the aspects preferred by the user. These aspects are extracted from the text reviews they publish about a specific item. Therefore, rather than utilizing only such default criteria as the item and user information in the prediction, we attempt to make use of two further important features. These are the aspects that the user cared about in his text reviews, in addition to the Helpfulness of the review. The latter refers to how much other users find a particular review helpful in making decisions. This could be indicated through ratings, upvotes, or other mechanisms.

For each new user of the cold-start group in the training dataset, all users who belong to the non-cold-start group and share the same aspects of the same item as this user are determined. Then, these users are ranked in descending order based on their helpfulness value. Next, ten users with the highest values are chosen. Next, ten users with the highest values are chosen to proceed with the rating prediction process according to this formula:

$$\rho = \left(\sum_{u=1}^{n} R_u\right)/n \qquad (3.1)$$

Where ρ is the predicted rating value, R_u is the ratings of the selected users, and n is the number of selected users. Having applied this formula, we get the final value of the predicted rating.

3.2.5 Non-cold start users' approach

The non-cold start group contains users with a number of ratings greater than the threshold, which is set at 5. Each user in this group has a set of aspects that were extracted from his text reviews during the previous stages. These aspects were later used in the process of rating prediction.

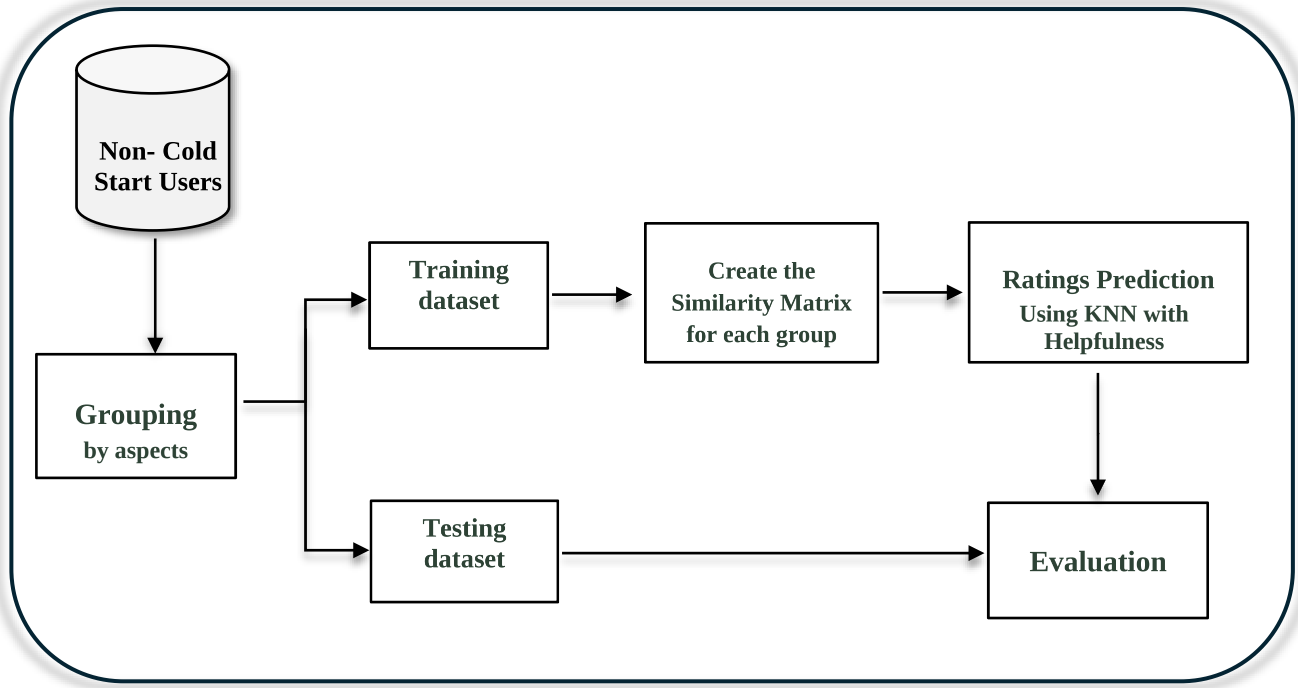
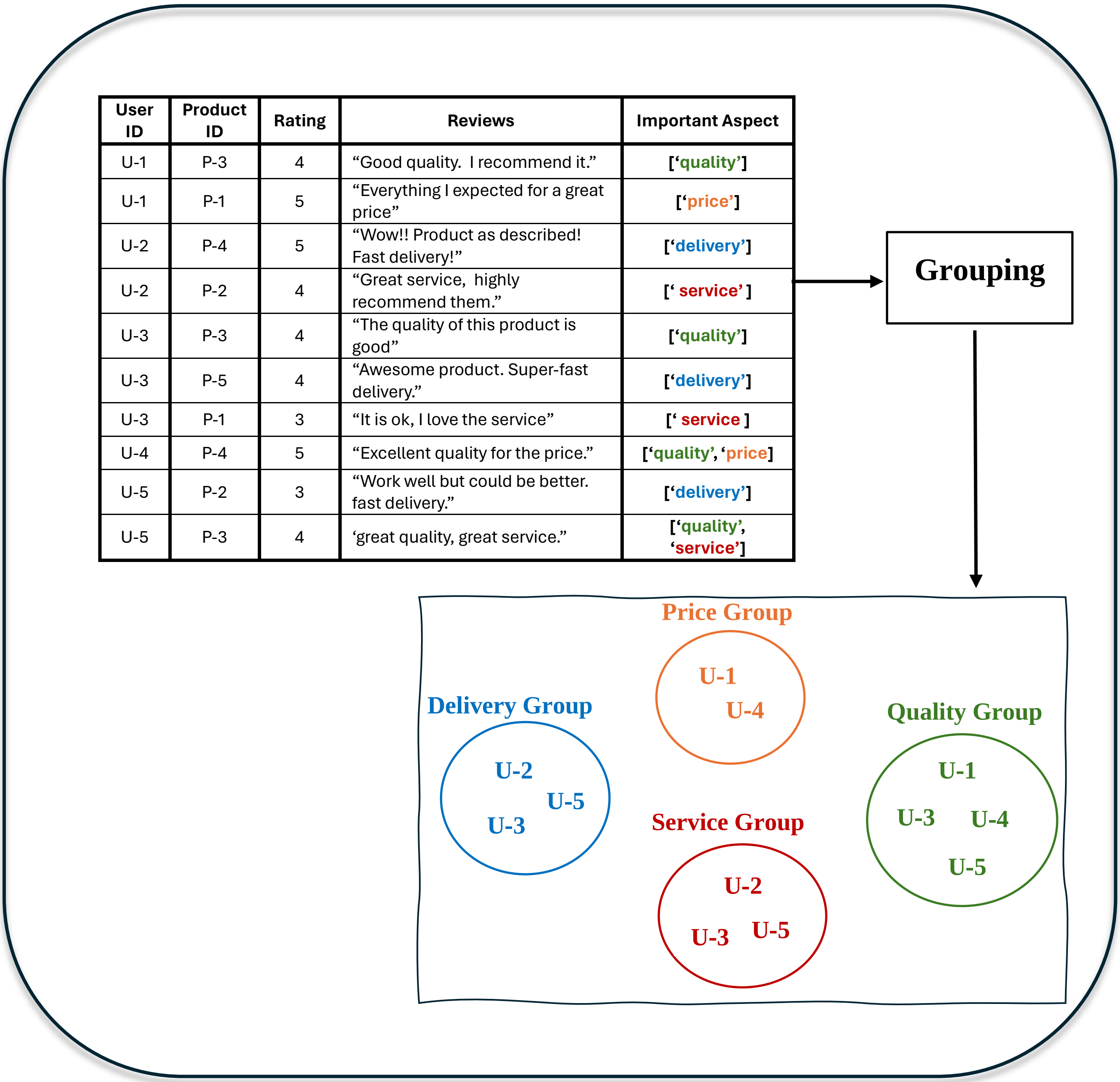


Figure 3.9: Non-cold start users' approach.

3.2.5.1 Grouping

This subsection describes how the users and their associated opinion information can be summarized in the form of groups which can be used for estimating the similarity between users belonging to the same cluster. In practice, several aspect terms are mentioned in user text reviews; Each aspect was represented as a group, and all users who mentioned this aspect in their textual review were gathered into one group.

Figure 3.10 illustrates the user grouping process, where users are grouped based on significant aspects extracted from their text reviews.



User ID	Product ID	Rating	Reviews	Important Aspect
U-1	P-3	4	“Good quality. I recommend it.”	[‘quality’]
U-1	P-1	5	“Everything I expected for a great price”	[‘price’]
U-2	P-4	5	“Wow!! Product as described! Fast delivery!”	[‘delivery’]
U-2	P-2	4	“Great service, highly recommend them.”	[‘ service’]
U-3	P-3	4	“The quality of this product is good”	[‘quality’]
U-3	P-5	4	“Awesome product. Super-fast delivery.”	[‘delivery’]
U-3	P-1	3	“It is ok, I love the service”	[‘ service]
U-4	P-4	5	“Excellent quality for the price.”	[‘quality’, ‘price]
U-5	P-2	3	“Work well but could be better. fast delivery.”	[‘delivery’]
U-5	P-3	4	‘great quality, great service.”	[‘quality’, ‘service’]

Figure 3.10: Grouping Process.

3.2.5.2 Splitting non-cold start users

After the grouping process was done, the result of the process was a set of groups, each one consisting of a collection of users who mentioned the aspect related to this group in their text reviews. Then, each group was split into 2 sub-datasets, training dataset-testing dataset, 80 percent of the entire data was used for training and 20 percent of it for testing. After this creating the Similarity Matrix for each training data belonging to each cluster was carried out.

3.2.5.3 Create the Similarity Matrix

For each group, it needed to build a similarity matrix among all the users who belong to the training dataset of that group. This can be done by using several similarity measures. In this work, cosine similarity (explained in section 2.4) was used to calculate the percentage of similarity between each user and the other users in the same cluster. Finding similarities between users is one of the tasks that facilitate the rating prediction process.

Algorithm 3.1: Creating the Similarity Matrix Algorithm

Input:
- user-item interaction matrix (rating matrix)

Output:
- The similarity matrix contains pairwise cosine similarities between users.

Begin
- Compute User Profiles:
  - For each user u:
    - Calculate the user's profile vector representing their preferences:
    - Normalize the user's ratings by subtracting the mean rating given by that user.
    - Treat the user's ratings as a vector, where each element corresponds to the rating of an item.

- Compute Cosine Similarity:
  - For each pair of users (u_i, u_j):
    - Compute the cosine similarity between their profile vectors:
      - Compute the dot product of profile vectors of users u_i and u_j.
      - Compute the Euclidean norms of the vectors u_i and u_j.
      - Calculate the cosine similarity using the formula:
        cosine_similarity(u_i, u_j) = dot_product(u_i, u_j) / (norm(u_i) * norm(u_j)).
    - Store the computed similarity value in the similarity matrix at position (i, j) and (j,i).

3.2.5.4 Rating Prediction For Non-Cold start Users

After building a similarity matrix for each group between users, the KNN method was used to identify the 20 most similar users to each user belonging to the training dataset. To determine the closest users, each user's 20 most similar users were arranged in descending order based on the helpfulness value of their review. After that 10 users with the highest helpfulness values were selected and identified as the closest user to this user so that their rating scores could be used in the rating prediction process. After selecting these ten users, the rating prediction process continues with the same formula that was used in the rating prediction process for non-cold start users (equation 3.1).

3.2.6 Evaluation

The performance evaluation is the last and most important step to test the effectiveness of the proposed model (RS-TRDL). The evaluation is done on the test set after training the proposed model on the training set. In particular, 80% of the dataset is devoted to the purpose of building and generalizing the model to be able to predict and classify test ratings which represent 20% of the ratings of each trained use. According to the rating prediction task accomplished in this model, measures such as (MAE, and RMSE) as mentioned in equations (2.10) and (2.11) that are explained in sections 2.9.4 and 2.9.5, are used to estimate the recommendation accuracy of RS-TRDL.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Overview

This chapter presents a description of the two Amazon datasets used in this thesis. Also, the experimental results will be discussed in terms of Environmental Experiments, Sentiment analysis Experiments, and Recommendation System Experiments

4.2 Datasets Description

In this thesis, two datasets from Amazon are used to train and test our RS-TRDL model. They are Amazon Electronics (AE) dataset and Amazon Fine Food (AFF) dataset. Table 3.1 describes the datasets which are used in this thesis.

These datasets encompass textual reviews of products along with numerical ratings sourced from a repository of recommendation system datasets. Specifically, the columns included in the dataset contain important information such as the identifier of the users and products, the rating value of the corresponding product by the corresponding user, the textual review written by the user, the date the review was written, and more information. This file contains all the information for each user of the dataset.

4.2.1 Amazon Electronics Dataset

Amazon Electronics dataset stands as one of Amazon's iconic datasets. It is commonly used for tasks related to product reviews and customer sentiment analysis. Commencing in 1995, this ever-growing collection features over a hundred million reviews, meticulously documenting user opinions and experiences with electronic products on Amazon.com. The original AE dataset includes 256,059 users and 74,258 items. Table 4.1 presents a description of this dataset.

Table 4.1: Amazon Electronics Dataset Description

Datasets	***#Users***	***#Items***	***#Ratings***	***#Reviews***
Amazon Electronics	802,412	96,799	1,048,575	1,048,575

This rich data source empowers academic researchers to delve into a variety of fields. AE dataset includes ten columns as explained in Table 4.2.

Table 4.2: Features included in the AE Dataset.

Feature	**Description**
Marketplace	2-letter country code of the marketplace where the review was written.
customer_id	A unique identifier is assigned to each user in the dataset
review_id	The unique ID of the review
product_id	A unique identifier is assigned to each product in the dataset
product_title	Title of the product
product_category	Broad product category that can be used to group reviews
star_rating	The 1-5 star rating value of the corresponding product by the corresponding user
helpful_votes	Number of users who found the review helpful
review_body	The review text
review_date	The date the review was written.

4.2.2 Amazon Fine Food Dataset

The Amazon Fine Food Reviews dataset is a popular subset of the larger Amazon Reviews dataset. It is a collection of reviews for food items on Amazon.com, focusing specifically on reviews for fine food products. It covers a timeframe of nearly two decades (October 1999 - October 2012). Table 4.3 describes the details of the dataset.

Table 4.3: Amazon Fine Food Dataset Description

Datasets	***#Users***	***#Items***	***#Ratings***	***#Reviews***
Amazon Fine Food	256,059	74,258	568,454	568,454

The AFF dataset includes nine columns or features as explained in Table 4.4.

Table 4.4: Features included in the AFF Dataset.

Feature	**Description**
ProductId	A unique identifier is assigned to each product in the dataset
UserId	A unique identifier is assigned to each user in the dataset
ProfileName	Contain information about your users such as name and contact information.
HelpfulnessNumerator	Number of users who found the review helpful
HelpfulnessDenominator	Number of users who indicated whether they found the review helpful or not
Score	The 1-5 star rating of the corresponding product by the corresponding user
Time	Time of the rating
Summary	Present a summarization of the textual review
Text	The review text

Given the memory constraints of the local machine environment, a representative subset was extracted from both extensive Amazon datasets. This involved randomly selecting 9,621 users who contributed 37,527 ratings and textual reviews across 16,330 items regarding AE dataset, ensuring a balanced and informative sample for subsequent analysis. Likewise, 8,489 users and their 67,814 ratings and reviews across 20,678 items were extracted from AFF dataset, thereby guaranteeing a statistically sound and diverse sample for analysis. Table 4.5 presents the datasets' descriptions.

Table 4.5: Datasets Description

Datasets	***#Users***	***#Cold start users***	***#Items***	***#Ratings***	***#Reviews***
Amazon Electronics	9621	8173	16330	37527	37527
Amazon Fine Food	8489	5490	20678	67814	67814

In the textual datasets above, there is an important feature that was used in this work, which is "Helpfulness". This feature typically refers to a numerical value or rating that indicates how helpful or valuable other users have found from a particular review. Within our work, the helpfulness feature was employed as a key metric for identifying influential users within the dataset. Analyzing user-generated text reviews and their associated helpfulness ratings, we successfully pinpointed individuals whose contributions consistently resonated with a large number of users, thus highlighting the importance of their perspectives.

4.3 Environmental Experiments

The environment that was used to build the proposed model, as well as the libraries that were utilized, are outlined in full below in Table 4.6.

Table 4.6: The hyperparameter of the environmental experiment.

Hyperparameter	***Value***
Platform	Google Colab
Programming Language	Python 3.10.12
NumPy	1.22.4
Scikit-learn	1.2.2
Keras	2.12.0
TensorFlow	2.12.0
nltk	3.8.1
Pandas	1.5.3
spaCY	3.7.4
BERTopic	0.16.0

4.4 Sentiment Analysis Experiments

In the process of sentiment analysis, where the goal is to classify textual data represented by user reviews, LSTM networks are a preferred choice due to their ability to retain information from long sequences and model relationships between distant elements within the text (LSTM was touched upon in section 2.8.3). This effectiveness in handling sequential data makes LSTMs a strong candidate for processing and classifying user reviews compared to other techniques.

First, the hyperparameters for the LSTM model were declared which is illustrated in Table 4.7.

Table 4.7: Hyperparameters of LSTM model.

Hyperparameter	***Value***
Batch size	32
Epochs	30
Loss Function	Binary cross entropy
Metrics	Accuracy
optimizers	Adam
Activation Function	Sigmoid
Learning Rate	0.01

Based on experimental observations during the text classification task with an LSTM algorithm, Adam optimizer emerged as the preferred choice due to its superior performance compared to other investigated algorithms. Also after conducting several experiments, the values 32 and 30 were adopted for the Hyperparameters Batch size and Epochs.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
embedding (Embedding)           (None, 10, 128)          128000
lstm (LSTM)                     (None, 10, 128)          65664
fc1 (Linear)                    (None, 128)             32896
dropout (Dropout)               (None, 128)             0
fc2 (Linear)                    (None, 1)               129
sigmoid (Sigmoid)               (None, 1)               0
================================================================
Total params: 226,689
Trainable params: 226,689
Non-trainable params: 0
----------------------------------------------------------------
```

Figure 4.1: Layers of LSTM Network.

As shown in Figure 4.1, the LSTM network consists of several layers, the first of which is the embedding layer which converts words to vectors. Then, two fully connected layers (fc1, fc2) are used after the LSTM layer to process the extracted features and generate the final output. Followed by a Dropout layer (with its value set to 0.3) to prevent overfitting. The final layer is the output layer, and its job is to carry out the classification process. The sigmoid function serves as the activation function for this layer.

The evaluation of the sentiment analysis model using LSTM algorithm was conducted using 5-fold cross-validation. Hence the training dataset is divided into 5 folds, one of the folds is used for validating the model, and the remaining 4 folds are used for training. The cross-validation process is then repeated 5 times with each of the 5 subsamples used exactly once as validation data. Finally, the average of the five experiments was calculated to determine the accuracy-specific value. Table 4.8 shows the results obtained from 5-fold cross-validation when experimenting with the model on the AFF dataset.

Table 4.8: The results of 5-fold cross-validation on AFF dataset.

Fold	***Training Accuracy***	***Validation Accuracy***
Fold 0	99.918	90.141
Fold 1	99.927	90.242
Fold 2	99.922	89.937
Fold 3	99.925	90.568
Fold 4	99.928	90.615
Average	**99.924**	**90.301**

To determine the final value of the model accuracy, the value of the average accuracy obtained through all 5-folds is calculated. By calculating the average accuracy values shown in Table 4.3. By averaging the accuracy values shown in the table, we get the final value of training accuracy = 99.924 and validation accuracy = 90.301.

The fifth fold was taken as an example of the results obtained in implementing the sentiment analysis model on AFF dataset. As shown in Figure 4.2 which visualizes the learning curve over 30 epochs.

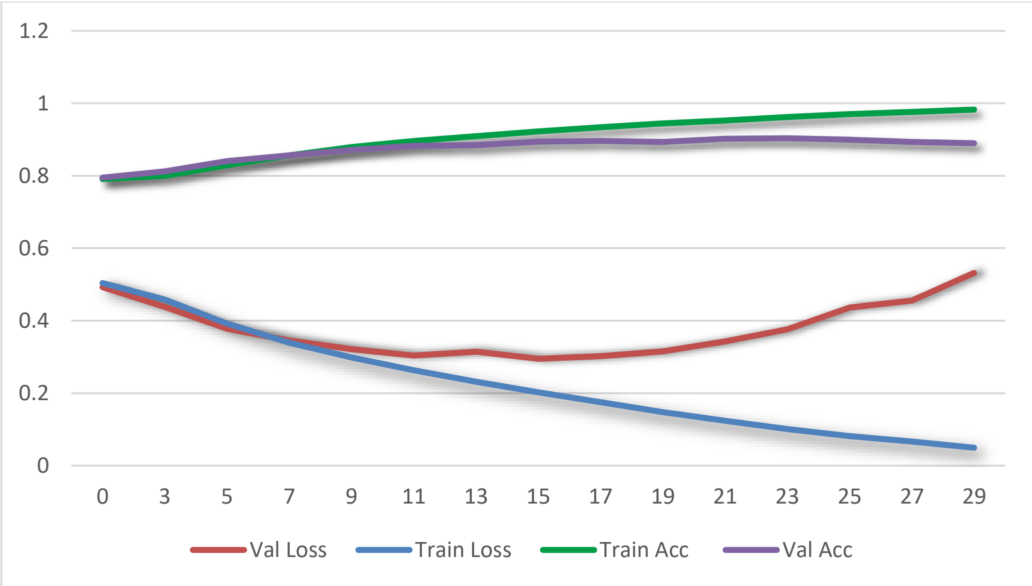


Figure 4.2: Learning curve of the fifth fold on AFF dataset.

Table 4.9 shows the results obtained from 5-fold cross-validation when experimenting with the model on the AE dataset.

Table 4.9: The results of 5-fold cross-validation on AE dataset.

Fold	***Training Accuracy***	***Validation Accuracy***
Fold 0	97.913	88.664
Fold 1	97.845	88.607
Fold 2	98.013	88.54
Fold 3	98.086	88.737
Fold 4	98.075	89.52
Average	**97.986**	**88.814**

To find the overall accuracy of the sentiment analysis model when experimenting on the AE dataset, the value of the average accuracies obtained through all 5-folds is calculated. By averaging the accuracy values shown in Table 4.9, we get the final value of training accuracy = 97.986 and validation accuracy = 88.814. Figure 4.3 visualizes the learning curve over 30 epochs on AE dataset.

Figure 4.3: Learning curve of the fifth fold on AE dataset.

4.5 Recommendation System Experiments

This study aims to use sentiment analysis in a recommendation system which is responsible for implementing two tasks:

1) Alleviating the user cold start problem.
2) Alleviating the sparsity problem and enhancing the recommender model.

The experiment of each task is presented in this section separately.

4.5.1 Alleviating the user cold start problem

Alleviating the user cold start problem is the first task in the proposed system. Extensive experiments were conducted in this task to verify the outputs of the proposed model on Amazon Electronics (AE) and Amazon Fine Food Reviews (AFF).

a) Experiments on Amazon Electronics dataset

For the AE dataset, we selected two previous studies that used the same dataset to compare them with our model RS-TRDL. The first one proposed RAPARE-MF model [33] which relied on a strategy known as RAPARE (Rating Comparison for Alleviating Cold-Start) and was instantiated on matrix factorization-based (RAPARE-MF).

The second study proposed DeCS model [38] which used a deep neural network (DNN) framework. Both studies aim to address the cold start problem in recommendation systems. Table 4.10 and Figure 4.4 below display the results.

Table 4.10: The results of RS-TRDL model against the comparison methods on cold start users' AE dataset.

NO.	***Methods***	***MAE***	***RMSE***
1	RAPARE-MF (2017) [33]	N/A	1.21
2	DeCS (2022) [38]	0.874	1.13
3	**RS-TRDL**	**0.574**	**1.02**

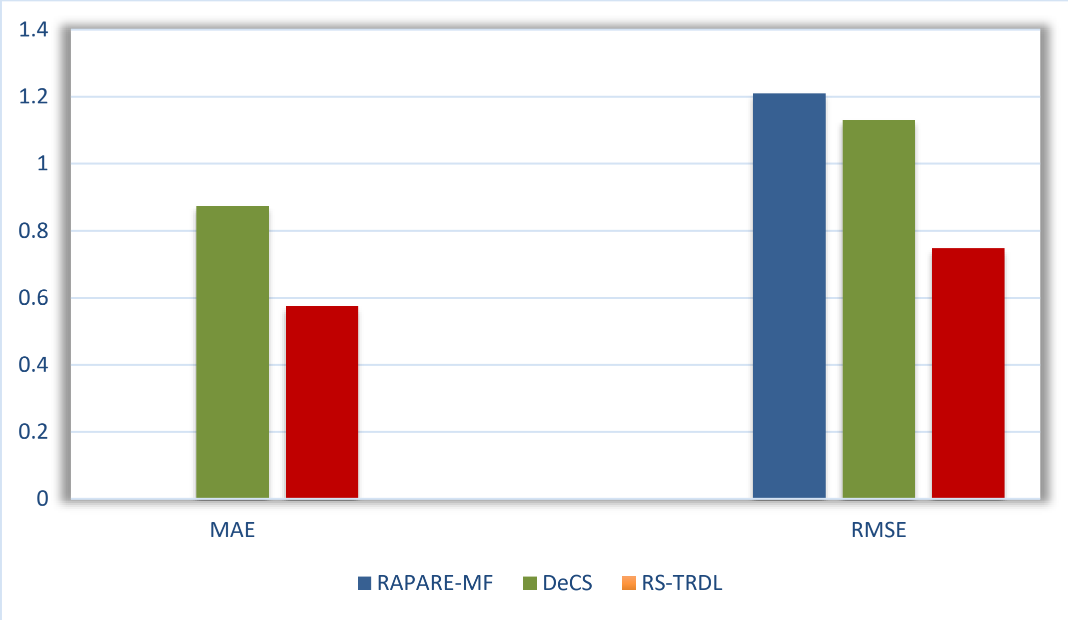


Figure 4.4: The results of RS-TRDL model against the comparison methods on cold start users of AE dataset.

The baseline models can be seen in Table 4.10 and Figure 4.4 illustrating that the RAPARE-MF method which is mainly based on user ratings for predictive performance achieves relatively the lowest performances on the RMSE metric compared to other baseline models. On the other hand, compared to the DeCS model, the proposed RS-TRDL achieves huge gains with a significant margin in terms of both RMSE and MAE. This appears to indicate the impact of incorporating the user textual feedback into the CF approach for RS.

The proposed method demonstrably surpasses previous models in terms of RMSE, achieving a value of 1.02 compared to the value of 1.13 achieved by DeCS model. This translates to a 9.73% improvement in metric performance. In addition to a 15.7% improvement in metric performance compared to RAPARE-MF model which achieved 1.21. Further, in terms of MAE, as shown in Table 4.10 and Figure 4.4 , the proposed model outperforms existing methods, attaining a value of 0.574 compared to the value of 0.874 observed in DeCS model. Thus, the proposed RS-TRDL model demonstrates a 34.32% improvement in predicting the target metric compared to the previous method.

The proposed approach has a significant benefit over baselines as it considers the user's opinions on several aspects of the item. This is in addition to the effective aspect extraction technique utilized to generate quality aspect terms required to improve RS performance. This clearly illustrates that a better aspect extraction technique can improve recommendation system performance.

b) Experiments on Amazon Fine Food dataset

In the literature reviews, No research has been found in the field of cold start problems that use AFF dataset in their work. Consequently, we compared the proposed RS-TRDL model with the baseline method. We considered that the baseline is an RS based on sentiment analysis using the TextBlob library without resorting to deep learning methods or utilizing the Helpfulness feature in the recommendation process.

Table 4.11: The results of RS-TRDL model against the baseline method on cold start users of AFF dataset.

Methods	***MAE***	***RMSE***
Baseline	0.882	1.261
RS-TRDL	**0.855**	**1.258**

As illustrated in Table 4.11, the obtained results showed that the accuracy of the recommender system improved because the RS-TRDL model relied on deep learning methods in the sentiment analysis process.

As previously proved, deep learning models have achieved state-of-the-art performance in various sentiment analysis tasks. In addition to relying on the helpfulness feature (which is an important feature in the dataset that identifies trust users) in the recommendation process, the focus was on the opinions of the users who have the greatest value of helpfulness. This means that many users benefited from their text reviews.

Compared to the baseline method with MAE achieving the value of 0.882, our model demonstrably improves this measure by reaching up to 0.855. This means reducing the error rate and improving performance by 3.06%. As for the RMSE value, the baseline method attained a value of 1.261 compared to the value of 1.258 observed in our proposed model. This indicates a 0.24% improvement in the performance. We can confidently state that the RS-TRDL model has great accuracy that confirms the use of deep learning algorithms in applying sentiment analysis to recommendation systems. It also demonstrates that using the helpfulness feature significantly improves the quality of the recommendations.

4.5.2 Alleviating the sparsity problem and enhancing the recommender model

The second task of the proposed model is alleviating the sparsity problem and enhancing the recommender model.

To obtain the overall accuracy result of the proposed system, two results were considered carefully:

1) evaluation results for cold-start users.
2) evaluation results for non-cold-start users.

By calculating the average of these two results, one value that acts as a representation of the system's overall performance was obtained.

a) Experiments on Amazon Electronics dataset

To evaluate the overall recommendation accuracy of our model which is designed to address the sparsity problem and enhance system performance (the second task of our proposed system), it was compared against six previous studies that used the same dataset (Amazon Electronics dataset).

The first study is MulAttRec [19] which combines attention-based analysis of user reviews with a hybrid prediction layer to personalize recommendations based on both explicit and implicit preferences. The second study proposed Co-Attentive Multi-task Learning (CAML) [21]. The third study proposed DualPC [26] which enhances the performance of review-based recommender systems by capturing the probabilistic connection between user preference prediction and content generation.

The fourth study introduces a latent factor model called adaptive deep latent factor model (ADLFM) [27], this model is capable of learning user preference factors for the specific items being considered. This flexibility is achieved by employing a user representation approach that enhances descriptions of the items users have rated, rather than relying solely on traditional user-item rating data.

The fifth study employed Improved Collaborative Filtering (ICF) [30], a method that exploits user similarities to predict missing values in user-item ratings. it identifies similar users and calculates the average distance between them to predict unrated items.

In the final study, a Transformer-based recommender system [31] was designed to exploit the richness of both utility matrices and textual data sources. Table 4.12 illustrates the MAE and RMSE values for the comparison methods against the result values of the proposed models and Figure 4.5 visualizes these results.

Table 4.12: The results of RS-TRDL model against the comparison methods on non-cold start users of AE dataset.

NO.	***Methods***	***MAE***	***RMSE***
1	MulAttRec(2018) [19]	*N/A*	*0.915*
2	CAML (2019) [21]	*N/A*	*1.085*
3	DualPC(2020) [26]	*N/A*	*0.967*
4	ADLFM (2020) [27]	*0.942*	*N/A*
5	ICF (2022) [30]	*0.80*	*1.10*
6	Transformer-based RS [five views] (2023) [31]	*0.554*	*1.195*
7	**RS-TRDL**	***0.389***	***0.747***

As shown in Table 4.12, the proposed RS-TRDL method demonstrably surpasses previous models in terms of RMSE, achieving a value of 0.747 compared to the range of (0.915-1.195). This translates into a noticeable improvement in performance estimated at a range (18.36% - 37.49%). This is a good percentage from which it can be concluded that the proposed model was able to reduce the percentage of error calculated by RMSE in the rating prediction process compared to other research.

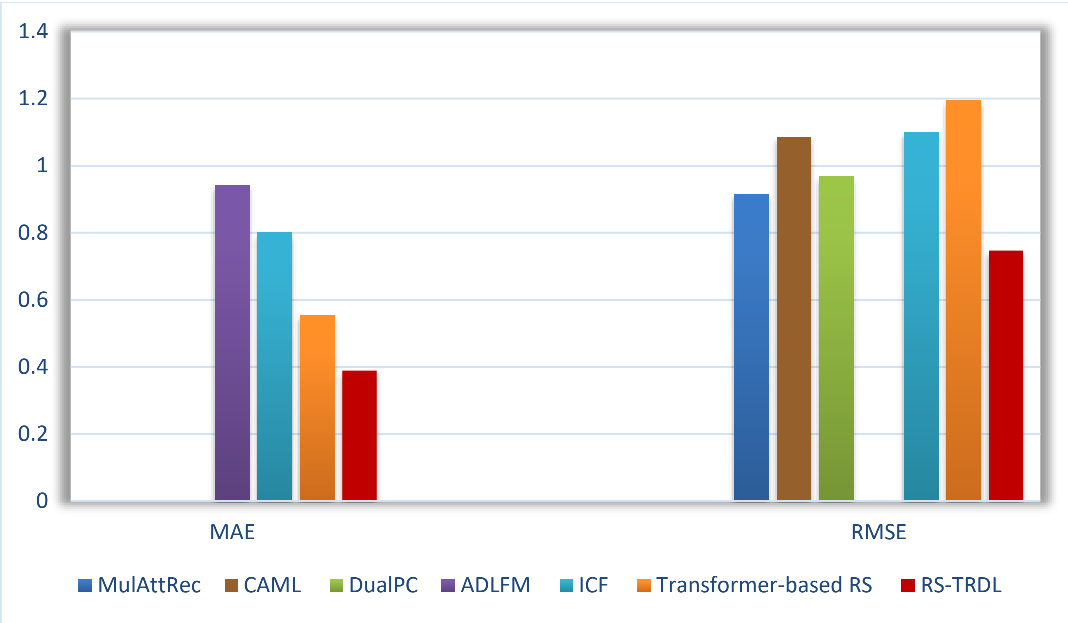


Figure 4.5: The results of RS-TRDL model against the comparison methods on non-cold start users of AE dataset.

Further, in terms of MAE, the proposed model outperforms existing methods, attaining a value of 0.389 compared to the range of (0.554-0.942). Thus, the proposed RS-TRDL model demonstrates a range (29.78%-58.70%) improvement in the rating prediction process compared to the previous methods. Table 4.13 shows the improvement of our model in percentages compared with previous research.

Table 4.13: Percentages of results improvement relative to RMSE and MAE metrics on AE dataset.

NO.	Methods	MAE	RMSE
1	MulAttRec [19]	*N/A*	*18.36%*
2	CAML [21]	*N/A*	*31.15%*
3	DualPC [26]	*N/A*	*22.75%*
4	ADLFM [27]	*58.70%*	*N/A*
5	ICF [30]	*51.38%*	*32.09%*
6	Transformer-based RS [five views] [31]	*29.78%*	*37.49%*

b) Experiments on Amazon Fine Food dataset

RS-TRDL model was evaluated using another dataset (AFF dataset) to gain the overall recommendation accuracy of the model which is compared against previous studies that used the same dataset. Three previous studies were selected to compare the results they obtained with the results obtained by the proposed model when implemented on the same dataset used in the selected studies.

The first study [32] investigated the effectiveness of various Matrix Factorization (MF) algorithms for predicting user ratings in the context of a food item recommendation system. By comparing Singular Value Decomposition (SVD), SVD with Implicit Ratings (SVD++), and Non-Negative Matrix Factorization. NMF achieved the lowest average prediction error, measured by Mean Absolute Error (MAE).

In the second study, a recommendation approach was presented [28], integrating sentiment analysis with collaborative filtering methods to enhance recommendation accuracy. This approach rests upon a dynamic, adaptive architecture that incorporates feature extraction techniques and deep learning models specifically designed to leverage sentiment information extracted from user reviews. The experiment was conducted with the recommender with sentiment analysis on different values of the β to obtain different results. Their findings indicated that a β value of 0.3 yielded the best results. Therefore, the results of RS-TRDL model were compared with the results obtained when using SVD++ algorithm with β=0.3.

The final study [18] used LSTM to solve the problem of classifying review usefulness and combined its outputs with the built recommender system using the matrix factorization model for the rating prediction process. Table 4.14 illustrates the MAE and RMSE values for the comparison methods against the result values of the proposed models and Figure 4.6 visualizes these results.

Table 4.14: The results of RS-TRDL model against the comparison methods on non-cold start users of AFF dataset.

No.	***Methods***	***MAE***	***RMSE***
1	NMF (2023) [32]	*0.7311*	*1.1205*
2	SVD++ With sentiment (β = 0.3) (2021) [28]	*0.8263*	*1.1292*
3	LSTM-Matrix Factorization (2016) [18]	*N/A*	*1.1198*
4	**RS-TRDL**	***0.7069***	***1.0836***

As shown in Table 4.14, the proposed RS-TRDL method demonstrably surpasses previous models in terms of MAE, achieving a value of 0.7069 compared to the range of (0.7311-0.8263). This translates into a noticeable improvement in performance estimated at a range (3.31%- 14.45%). This demonstrates that the proposed model was able to reduce the percentage of error calculated by MAE in the rating prediction process compared to other research.

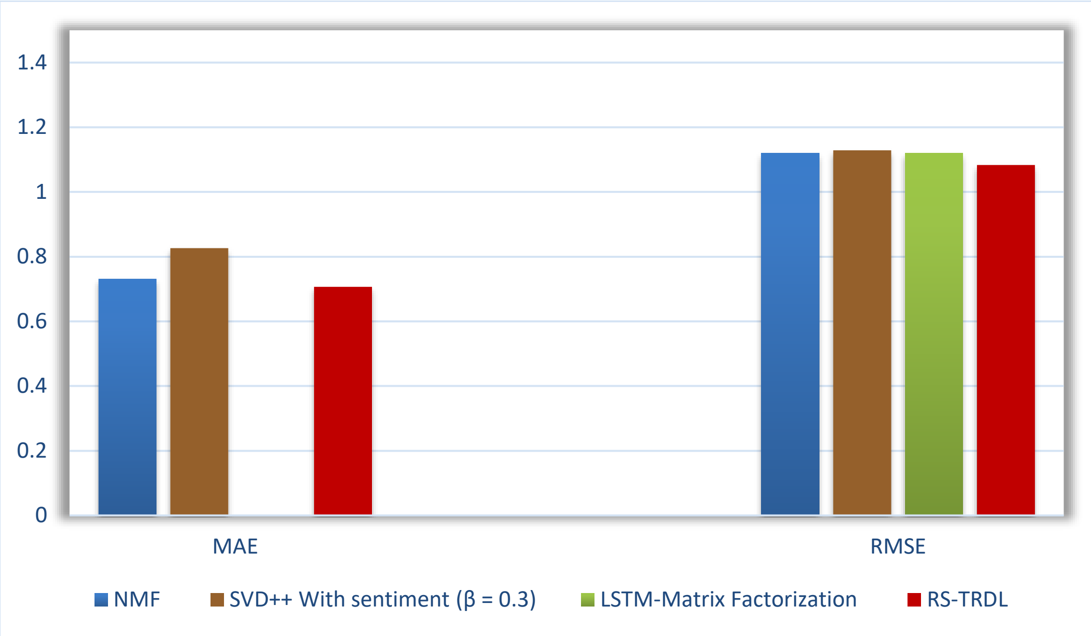


Figure 4.6: The results of RS-TRDL model against the comparison methods on non-cold start users of AFF dataset.

Further, in terms of RMSE, our model outperforms existing methods, attaining a value of 1.0836 compared to the range of (1.1198-1.1292). Therefore, the proposed RS-TRDL model demonstrates a range (3.21%-4.03%) improvement in the rating prediction process compared to the previous methods. Table 4.15 shows the improvement of our model in percentages compared with previous research.

Table 4.15: Percentages of results improvement relative to RMSE and MAE metrics on AFF dataset.

No.	***Methods***	***MAE***	***RMSE***
1	NMF (2023) [32]	*3.31%*	3.29%
2	SVD++ With sentiment (β = 0.3) (2021) [28]	*14.45%*	4.03%
3	LSTM-Matrix Factorization (2016) [18]	*N/A*	3.21%

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 Overview

This chapter presents the thesis's conclusions in addition to the findings and suggestions for future research.

5.2 Conclusions

Based on the results that have been obtained from the performance of the proposed models on different datasets, we have reached the following conclusions:

1) When we use the proposed RS-TRDL model, the obtained results show that the accuracy of the system has been improved because the system can analyze text reviews to provide a greater understanding of user preferences and extract the aspects that each user cares about.

2) On the other hand, the number of cold-start users who get the benefit of the recommendations has been increased by using the proposed method. In other words, the coverage measures have been improved.

3) In the proposed recommender system context, non-Cold Start users achieved demonstrably superior results compared to Cold Start users. This disparity can be attributed to the wealth of information, ratings, and text reviews Available by non-Cold Start users, which the system can leverage to generate more accurate recommendations.

4) The extensive experiments on two Amazon datasets depicted that the proposed RS-TRDL model surpassed all literature-reviewed comparison methods in the rating prediction process for both tasks it was built to

perform. This supported the idea that integrating recommender systems and sentiment analysis would have significant advantages.

5) It can be confidently stated that the proposed RS-TRDL model has great accuracy which supports the use of deep learning algorithms in applying sentiment analysis to recommendation systems. It also demonstrates that using the helpfulness feature significantly improves the quality of the recommendations..

5.3 Future Work

In this section, some ideas and suggestions related to our thesis will be presented. Regarding the part related to sentiment analysis, several proposals can be applied in the future:

1. Experiment with increasing the number of folds and epochs in the K-fold cross-validation process to obtain new results and compare them with the existing results

2. Analyzing emojis within texts to determine the sentiment being expressed therein.

In terms of the part related to the recommendation model, there are several proposals for future work, such as:

3. Using different clustering techniques to community detection of large datasets and utilize it in the recommendation system.

4. Depending on timestamp information, text reviews can be filtered from other fake ones to avoid wrong recommendations to the target users within RS-TRDL model, because the incorrect suggestion of an item is worse than no correct suggestion of this item.

5. Employing other rating similarity metrics to infer the nearest trustworthy neighbors and compare the results with the RS-TRDL results.

6. Extending the RS-TRDL model to include implicit feedback information in addition to the explicit information used, represented by users' textual reviews. Trust relations are one of the implicit feedback information that can be utilized to reach the best possible performance. In other words, incorporating the review-based RS with other types of recommenders, such as trust-aware RS, might be a potent combination of a hybrid RS.

REFERENCES

[1] C. Zhang, M. Yang, J. Lv, and W. Yang, “An improved hybrid collaborative filtering algorithm based on tags and time factor,” *Big Data Min. Anal.*, vol. 1, no. 2, pp. 128–136, 2018, doi: 10.26599/BDMA.2018.9020012.

[2] S. Forouzandeh, M. Rostami, and K. Berahmand, “Presentation a Trust Walker for rating prediction in recommender system with Biased Random Walk: Effects of H-index centrality, similarity in items and friends,” *Eng. Appl. Artif. Intell.*, vol. 104, no. May, p. 104325, 2021, doi: 10.1016/j.engappai.2021.104325.

[3] V. K. Singh, M. Mukherjee, and G. K. Mehta, “Combining collaborative filtering and sentiment classification for improved movie recommendations,” *in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 7080 LNAI. pp. 38–50,. doi: 10.1007/978-3-642-25725-4_4.

[4] Y. Li, J. Liu, J. Ren, and Y. Chang, “A Novel Implicit Trust Recommendation Approach for Rating Prediction,” *IEEE Access*, vol. 8, pp. 98305–98315, 2020, doi: 10.1109/ACCESS.2020.2997040.

[5] S. M. Ali, G. K. Nayak, R. K. Lenka, and R. K. Barik, “Movie Recommendation System Using Genome Tags and Content-Based Filtering,” in *In: Kolhe, M., Trivedi, M., Tiwari, S., Singh, V. (eds) Advances in Data and Information Sciences.*, Springer Nature Singapore Pte Ltd., 2018, pp. 85–94. doi: 10.1007/978-981-10-8360-0_8.

[6] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, “Collaborative filtering recommender systems,” *Found. Trends Human-Computer Interact.*, vol. 4, no. 2, pp. 81–173, 2010, doi: 10.1561/1100000009.

[7] M. Srifi, A. Oussous, A. A. Lahcen, and S. Mouline, “Recommender systems based on collaborative filtering using review texts-A survey,” *Information (Switzerland)*, vol. 11, no. 6. MDPI AG, Jun. 01, 2020. doi: 10.3390/INFO11060317.

[8] M. Papagelis, D. Plexousakis, and T. Kutsuras, “Alleviating the sparsity problem of collaborative filtering using trust inferences,” in *Herrmann, P., Issarny, V., Shiu, S. (eds) Trust Management. iTrust 2005. Lecture Notes in Computer Science,* Berlin, Heidelberg: Springer-Verlag, 2005, pp. 224–239. doi: 10.1007/11429760_16.

[9] I. A. Al-Mani, A. M. A. Al-Sabaawi, and M. H. Hussien, “A Review Paper of Model Based Collaborative Filtering Techniques,” *2022 Int. Conf. Data Sci. Intell. Comput. ICDSIC 2022*, no. Icdsic, pp. 52–57, 2022, doi: 10.1109/ICDSIC56987.2022.10076148.

[10] H. Koohi and K. Kiani, “A new method to find neighbor users that improves the performance of Collaborative Filtering,” *Expert Syst. Appl.*, vol. 83, pp. 30–39, 2017, doi: 10.1016/j.eswa.2017.04.027.

[11] X. Chao and C. Guangcai, “Collaborative Filtering and Leaders’ Advice Based Recommendation System for Cold Start Users,” *PervasiveHealth Pervasive Comput. Technol. Heal.*, pp. 158–164,, doi: 10.1145/3404555.3404644.

[12] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013, doi: 10.1016/j.knosys.2013.03.012.

[13] F. G. Contratres, S. N. Alves-Souza, L. V. L. Filgueiras, and L. S. DeSouza, *Sentiment analysis of social network data for cold-start relief in recommender systems*, vol. 746. Springer International Publishing, 2018. doi: 10.1007/978-3-319-77712-2_12.

[14] L. Chen, G. Chen, and F. Wang, “Recommender systems based on user reviews: the state of the art,” *User Model. User-adapt. Interact.*, vol. 25, no. 2, pp. 99–154, 2015, doi: 10.1007/s11257-015-9155-5.

[15] S. Kanwal, S. Nawaz, M. K. Malik, and Z. Nawaz, “A review of text-based recommendation systems,” *IEEE Access*, vol. 9, no. i, pp. 31638–31661, 2021, doi: 10.1109/ACCESS.2021.3059312.

[16] R. Wang, D. Zhou, M. Jiang, J. Si, and Y. Yang, “A survey on opinion mining: From stance to product aspect,” *IEEE Access*, vol. 7, no. c, pp. 41101–41124, 2019, doi: 10.1109/ACCESS.2019.2906754.

[17] M. Angioni, M. L. Clemente, and F. Tuveri, “Combining Opinion Mining with Collaborative Filtering,” in *WEBIST 2015 - 11th International Conference on Web Information Systems and Technologies, Proceedings*, SciTePress, 2015, pp. 370–380. doi: 10.5220/0005412403700380.

[18] Z. Zhou and L. Xu, “Amazon Food Review Classification using Deep Learning and Recommender System,” *Stanford Univ.*, pp. 1–7, 2016, [Online]. Available: https://cs224d.stanford.edu/reports/ZhouXu.pdf

[19] Z. Lin, W. Y. B, Y. Zhang, and H. Wang, *MulAttenRec : A Multi-level Attention-Based Model for Recommendation*, vol. 1, no. 91648204. Springer International Publishing, 2018. doi: 10.1007/978-3-030-04179-3.

[20] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, and X. Luo, “A context-aware user-item representation learning for item recommendation,” *ACM Trans. Inf. Syst.*, vol. 37, no. 2, 2019, doi: 10.1145/3298988.

[21] Z. Chen *et al.*, “Co-attentive multi-task learning for explainable recommendation,” *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2019-Augus, pp. 2137–2143, 2019, doi: 10.24963/ijcai.2019/296.

[22] H. Liu *et al.*, “Hybrid neural recommendation with joint deep representation learning of ratings and reviews,” *Neurocomputing*, vol. 374, no. xxxx, pp. 77–85, 2020, doi:

10.1016/j.neucom.2019.09.052.

[23] A. Da'u, N. Salim, I. Rabiu, and A. Osman, "Weighted aspect-based opinion mining using deep learning for recommender system," *Expert Syst. Appl.*, vol. 140, Feb. 2020, doi: 10.1016/j.eswa.2019.112871.

[24] B. T. Hung, "Integrating Sentiment Analysis in Recommender Systems," *Springer Ser. Reliab. Eng.*, no. May, pp. 127–137, 2020, doi: 10.1007/978-3-030-43412-0_8.

[25] F. Harrag, A. S. Al-Salman, and A. Alquahtani, "Arabic opinion mining using a hybrid recommender system approach," *arXiv Prepr. arXiv2009.07397*, 2020, [Online]. Available: https://arxiv.org/abs/2009.07397

[26] P. Sun, L. Wu, K. Zhang, Y. Fu, R. Hong, and M. Wang, "Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation," *Web Conf. 2020 - Proc. World Wide Web Conf. WWW 2020*, pp. 837–847, 2020, doi: 10.1145/3366423.3380164.

[27] J. Han *et al.*, "Adaptive Deep Modeling of Users and Items Using Side Information for Recommendation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 3, pp. 737–748, 2020, doi: 10.1109/TNNLS.2019.2909432.

[28] R. Systems, C. N. Dang, and N. Moreno-garc, "An Approach to Integrating Sentiment Analysis into," *Sensors (Switzerland)*, vol. 21, 2021.

[29] H. J. Oudah and M. H. Hussein, "Exploiting Textual Reviews for Recommendation Systems Improvement," *2022 Int. Conf. Data Sci. Intell. Comput. ICDSIC 2022*, no. Icdsic, pp. 70–74, 2022, doi: 10.1109/ICDSIC56987.2022.10076074.

[30] K. A. S. A. V P and Z. K. A. Baizal, "Improved Collaborative Filtering Recommender System Based on Missing Values Imputation on E-Commerce," *Build. Informatics, Technol. Sci.*, vol. 3, no. 4, pp. 453–459, 2022, doi: 10.47065/bits.v3i4.1214.

[31] T. L. Ho, A. C. Le, and D. H. Vu, "Multiview Fusion Using Transformer Model for Recommender Systems: Integrating the Utility Matrix and Textual Sources," *Appl. Sci.*, vol. 13, no. 10, 2023, doi: 10.3390/app13106324.

[32] M. Bayu Samudra Siddik and A. Toto Wibowo, "Collaborative Filtering Based Food Recommendation System Using Matrix Factorization," vol. 7, no. 3, pp. 1041–1049, 2023, doi: 10.30865/mib.v7i3.6049.

[33] J. Xu, Y. Yao, H. Tong, X. Tao, and J. Lu, "RaPare: A generic strategy for cold-start rating prediction problem," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1296–1309, 2017, doi: 10.1109/TKDE.2016.2615039.

[34] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li, "Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp.*

Educ. Adv. Artif. Intell. EAAI 2019, pp. 94–101, 2019, doi: 10.1609/aaai.v33i01.330194.

[35] J. Herce-Zelaya, C. Porcel, J. Bernabé-Moreno, A. Tejeda-Lorente, and E. Herrera-Viedma, “New technique to alleviate the cold start problem in recommender systems using information from social media and random decision forests,” *Inf. Sci. (Ny).*, vol. 536, pp. 156–170, 2020, doi: 10.1016/j.ins.2020.05.071.

[36] J. Feng, Z. Xia, X. Feng, and J. Peng, “RBPR: A hybrid model for the new user cold start problem in recommender systems,” *Knowledge-Based Syst.*, vol. 214, p. 106732, 2021, doi: 10.1016/j.knosys.2020.106732.

[37] J. Wang, K. Ding, and J. Caverlee, “Sequential Recommendation for Cold-start Users with Meta Transitional Learning,” *SIGIR 2021 - Proc. 44th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 1783–1787, 2021, doi: 10.1145/3404835.3463089.

[38] R. Mondal and B. Bhowmik, “DeCS: A Deep Neural Network Framework for Cold Start Problem in Recommender Systems,” *2022 IEEE Reg. 10 Symp. TENSYMP 2022*, pp. 1–6, 2022, doi: 10.1109/TENSYMP54529.2022.9864409.

[39] F. Ricci, B. Shapira, and L. Rokach, *Recommender systems handbook, Second edition*. 2015. doi: 10.1007/978-1-4899-7637-6.

[40] M. H. Hussein, A. A. Alsakaa, and H. A. Marhoon, “Adopting explicit and implicit social relations by SVD++ for recommendation system improvement,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 2, pp. 471–478, 2021, doi: 10.12928/TELKOMNIKA.v19i2.18149.

[41] R. P. Shen, H. R. Zhang, H. Yu, and F. Min, “Sentiment based matrix factorization with reliability for recommendation,” *Expert Syst. Appl.*, vol. 135, pp. 249–258, 2019, doi: 10.1016/j.eswa.2019.06.001.

[42] Y. Hu, F. Xiong, D. Lu, X. Wang, X. Xiong, and H. Chen, “Movie collaborative filtering with multiplex implicit feedbacks,” *Neurocomputing*, vol. 398, no. xxxx, pp. 485–494,, doi: 10.1016/j.neucom.2019.03.098.

[43] M. Hernández-Rubio, I. Cantador, and A. Bellogín, *A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews*, vol. 29, no. 2. Springer Netherlands, 2019. doi: 10.1007/s11257-018-9214-9.

[44] A. M. A. Al-Sabaawi, H. Karacan, and Y. E. Yenice, “Svd++ and clustering approaches to alleviating the cold-start problem for recommendation systems,” *Int. J. Innov. Comput. Inf. Control*, vol. 17, no. 2, pp. 383–396, 2021, doi: 10.24507/ijicic.17.02.383.

[45] F. Tahmasebi, M. Meghdadi, S. Ahmadian, and K. Valiallahi, “A hybrid recommendation system based on profile expansion technique to alleviate cold start

problem," *Multimed. Tools Appl.*, vol. 80, no. 2, pp. 2339–2354, 2021, doi: 10.1007/s11042-020-09768-8.

[46] C. V. Sundermann, M. A. Domingues, R. A. Sinoara, R. M. Marcacini, and S. O. Rezende, "Using opinion mining in context-aware recommender systems: A systematic review," *Information (Switzerland)*, vol. 10, no. 2. MDPI AG, Jan. 28, 2019. doi: 10.3390/info10020042.

[47] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00592-5.

[48] A. M. Ahmed Al-Sabaawi, H. Karacan, and Y. E. Yenice, "Exploiting implicit social relationships via dimension reduction to improve recommendation system performance," *PLoS One*, vol. 15, no. 4, 2020, doi: 10.1371/journal.pone.0231457.

[49] M. R. Petrusel and S. G. Limboi, "A Restaurants Recommendation System: Improving Rating Predictions Using Sentiment Analysis," *in Proceedings - 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2019,*. pp. 190–197,. doi: 10.1109/SYNASC49474.2019.00034.

[50] R. Chen, Q. Hua, Y. S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: from traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, no. c, pp. 64301–64320, 2018, doi: 10.1109/ACCESS.2018.2877208.

[51] M. Jiang, D. Song, L. Liao, and F. Zhu, "A Bayesian recommender model for user rating and review profiling," *Tsinghua Sci. Technol*, vol. 20, no. 6, pp. 634–643,, doi: 10.1109/TST.2015.7350016.

[52] H. J. Oudah and M. H. Hussein, "Exploiting social trust via weighted voting strategy for recommendation systems improvement Exploiting social trust via weighted voting strategy for recommendation systems improvement".

[53] D. Jasim Mohammed Mustafa and A. Hiba Jabbar Abdul wahid Aleqabie, "TEXT-IMAGE SENTIMENTANALYSIS IN SOCIAL NETWORKS USING DEEP LEARNING TECHNIQUE Written by," 2023.

[54] X. Li, H. Wang, and X. Yan, "Accurate recommendation based on opinion mining," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2015, pp. 399–408. doi: 10.1007/978-3-319-12286-1_41.

[55] Y. Betancourt and S. Ilarri, "Use of text mining techniques for recommender systems," *ICEIS 2020 - Proc. 22nd Int. Conf. Enterp. Inf. Syst.*, vol. 1, no. Iceis, pp. 780–787, 2020, doi: 10.5220/0009576507800787.

[56] J. A. Balazs and J. D. Velásquez, *Opinion Mining and Information Fusion: A survey,* vol. 27. 2016. doi: 10.1016/j.inffus.2015.06.002.

[57] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Inf. Fusion*, vol. 36, pp. 10–25, 2017, doi: 10.1016/j.inffus.2016.10.004.

[58] A. Gupta, L. Kumar, R. Jain, and P. Nagrath, *Heart Disease Prediction Using Classification (Naive Bayes)*, vol. 121, no. Ic4s. 2020. doi: 10.1007/978-981-15-3369-3_42.

[59] K. Sarawgi, "Opinion Mining: Aspect Level Sentiment Analysis using SentiWordNet and Amazon Web Services General Terms Data-mining, Text-mining, Opinion mining," *Int. J. Comput. Appl.*, vol. 158, no. 6, pp. 975–8887, 2017, [Online]. Available: www.hemobileindian.com,

[60] F. Hemmatian and M. K. Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 1495–1545, 2019, doi: 10.1007/s10462-017-9599-6.

[61] S. Moghaddam, "The FLDA Model for Aspect-based Opinion Mining : Addressing the Cold Start Problem Categories and Subject Descriptors," *Proc. 22nd Int. Conf. World Wide Web*, pp. 909–918, 2013.

[62] M. Z. Naf'an, A. A. Bimantara, A. Larasati, E. M. Risondang, and N. A. S. Nugraha, "Sentiment Analysis of Cyberbullying on Instagram User Comments," *J. Data Sci. Its Appl*, vol. 2, no. 1, pp. 88–98,, doi: 10.21108/jdsa.2019.2.20.

[63] Q. Tian, N. Sebe, G. J. Qi, B. Huet, R. Hong, and X. Liu, "MultiMedia modeling: 22nd International conference, MMM 2016 Miami, FL, USA, january 4–6, 2016 proceedings, part II," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, vol. 9517, pp. 15–27,, doi: 10.1007/978-3319-27674-8.

[64] J. Khairnar and M. Kinikar, "Machine Learning Algorithms for Opinion Mining and Sentiment Classification," *Int. J. Sci. Res. Publ.*, vol. 3, no. 6, pp. 1–6, 2013, [Online]. Available: www.ijsrp.org

[65] Y. Kai, Y. Cai, H. Dongping, J. Li, Z. Zhou, and X. Lei, "An effective hybrid model for opinion mining and sentiment analysis," *2017 IEEE Int. Conf. Big Data Smart Comput. BigComp 2017*, pp. 465–466, 2017, doi: 10.1109/BIGCOMP.2017.7881759.

[66] M. Wankhade, A. C. S. Rao, and C. Kulkarni, *A survey on sentiment analysis methods, applications, and challenges*, vol. 55, no. 7. Springer Netherlands, 2022. doi: 10.1007/s10462-022-10144-1.

[67] S. Consoli, L. Barbaglia, and S. Manzan, "Fine-grained, aspect-based sentiment analysis on economic and financial lexicon," *Knowledge-Based Syst.*, vol. 247, p. 108781, 2022, doi: 10.1016/j.knosys.2022.108781.

[68] B. Haq, S. M. Daudpota, A. S. Imran, Z. Kastrati, and W. Noor, "A Semi-Supervised

Approach for Aspect Category Detection and Aspect Term Extraction from Opinionated Text," *Comput. Mater. Contin.*, vol. 77, no. 1, pp. 115–137, 2023, doi: 10.32604/cmc.2023.040638.

[69] D. R. Tobergte and S. Curtis, "Deep Learning in Python," *J. Chem. Inf. Model*, vol. 53, no. 9, pp. 1689–1699,.

[70] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. February, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[71] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Comput. Sci. Rev.*, vol. 40, p. 100379, 2021, doi: 10.1016/j.cosrev.2021.100379.

[72] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to machine learning, neural networks, and deep learning," *Transl. Vis. Sci. Technol*, vol. 9, no. 2, pp. 1–12,, doi: 10.1167/tvst.9.2.14.

[73] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning." pp. 1–20,. [Online]. Available: http://arxiv.org/abs/1811.03378

[74] S. KILIÇARSLAN, K. ADEM, and M. ÇELİK, "An overview of the activation functions used in deep learning algorithms," *J. New Results Sci.*, vol. 10, no. 3, pp. 75–88, 2021, doi: 10.54187/jnrs.1011739.

[75] N. Kyurkchiev and S. Markov, "Sigmoidal Functions: Some Computational and Modelling Aspects," *Biomath Commun.*, vol. 1, no. 2, 2015, doi: 10.11145/j.bmc.2015.03.081.

[76] S. Sharma and S. Sharma, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," vol. 4, no. 12. pp. 310–316,.

[77] M. Shyu, S. Chen, and S. S. Iyengar, "A Survey on Deep Learning : Algorithms , Techniques ," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, 2018.

[78] R. Zaheer and H. Shaziya, "A Study of the Optimization Algorithms in Deep Learning," *Proc. 3rd Int. Conf. Inven. Syst. Control. ICISC*, pp. 536–539,, doi: 10.1109/ICISC44355.2019.9036442.

[79] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Spec. Top.*, vol. 228, no. 10, pp. 2313–2324, 2019, doi: 10.1140/epjst/e2019-900046-x.

[80] G. S. . Murthy, S. R. Allu, B. Andhavarapu, M. Bgadi, and M. Belusonti, "Text based Sentiment Analysis using Long Short Term Memory (LSTM)," *Int. J. Eng. Res. Technol.*, vol. 9, no. 05, pp. 299–303, 2020.

[81] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, pp. 650–655, 2021, doi: 10.1016/j.procir.2021.03.088.

[82] H. Salman, J. Grover, and T. Shankar, "Hierarchical Reinforcement Learning for Sequencing Behaviors," vol. 2733, no. March, pp. 2709–2733, 2018, doi: 10.1162/NECO.

[83] Ž. Vujović, "Classification Model Evaluation Metrics," *Int. J. Adv. Comput. Sci. Appl*, vol. 12, no. 6, pp. 599–606,, doi: 10.14569/IJACSA.2021.0120670.

[84] A. H. Jadidinejad, C. MacDonald, and I. Ounis, "Unifying explicit and implicit feedback for rating prediction and ranking recommendation tasks," *in ICTIR 2019 - Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval,*. pp. 149–156,. doi: 10.1145/3341981.3344225.

[85] H. Zhao, Y. Zhang, and Y. Xiao, "A new collaborative filtering algorithm with combination of explicit trust and implicit trust," in *13th International Conference on Computer Science and Education, ICCSE 2018*, IEEE, 2018, pp. 319–323. doi: 10.1109/ICCSE.2018.8468763.

[86] J. L. Herlocker, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst. (TOIS*, vol. 22, no. 1, pp. 5–53.

الخلاصة

لقد أتاح الإنترنت والويب إمكانية مشاركة كمية هائلة من المعلومات والوصول إليها من قبل أعداد كبيرة من الأشخاص. وقد أدى هذا إلى مشكلة تسمى الحمل الزائد للمعلومات (information overload) و هو التحدي المتمثل في اتخاذ القرارات عند مواجهة الكثير من المعلومات. استلزمت هذه المشكلة إنشاء أنظمة توصية تعالج تحدي الحمل الزائد للمعلومات من خلال اقتراح منتجات أو خدمات قد تكون مفيدة للمستخدمين ومصالحهم.

قد تواجه أنظمة التوصية عدة مشاكل منها التشتت (Sparsity) والبداية الباردة (Cold-start). تؤدي هذه المشكلات إلى انخفاض أداء نظام التوصية.

في هذا العمل، تم اقتراح نظام التوصية القائم على المراجعات النصية وباستخدام طريقة التعلم العميق (RS-TRDL) لأداء مهمتين رئيسيتين: المهمة الأولى هي تخفيف مشكلة (user cold-start) والمهمة الثانية هي تخفيف مشكلة (Sparsity) وتعزيز أداء النموذج المقترح. تم استخدام المراجعات النصية كمعلومات إضافية إلى جانب التقييمات الرقمية للمستخدمين. وتم استخلاص جوانب مهمة من هذه المراجعات، بالإضافة إلى قطبية المشاعر باستخدام إحدى خوارزميات التعلم العميق وهي خوارزمية الذاكرة طويلة المدى (LSTM)، للاستفادة من هذه الجوانب في عملية التوصية.

في نموذج RS-TRDL، تم إجراء المعالجة المسبقة على مجموعة البيانات وتضمنت هذه المرحلة خطوات مختلفة، بما في ذلك التعامل مع القيم المفقودة وتصنيف البيانات. بالإضافة إلى ذلك، قام بدمج عمليات المعالجة المسبقة الخاصة بالنص مثل تنظيف النص والمعالجة المسبقة العامة للنص، ثم انتقل بعد ذلك إلى استخراج الجوانب. استخدمت هذه الخطوة spaCy لاستخراج الأسماء عبر وضع علامات على جزء من الكلام (POS). بالإضافة إلى ذلك، تم إجراء نمذجة الموضوع باستخدام خوارزمية BERTopic. وأخيرًا، تم إجراء تحليل المشاعر باستخدام خوارزمية الذاكرة طويلة المدى (LSTM).

بعد استخراج الجوانب، تمت معاملة مستخدمي البداية الباردة ومستخدمي البداية غير الباردة بشكل منفصل.

بالنسبة لمستخدمي البداية الباردة، تم إجراء عملية التنبؤ بالتقييم باستخدام خوارزمية K-Nearest Neighbors (KNN)استنادًا إلى تقييمات المستخدمين غير المبتدئين الذين يتشاركون نفس الجوانب من نفس العناصر ولديهم قيمة مساعدة (helpfulness) عالية.

بالنسبة للمستخدمين غير الباردين، تمت عملية التجميع أولاً بناءً على الجوانب المستخرجة من مراجعات المستخدمين، ثم تم إنشاء مصفوفة تشابه لكل مجموعة باستخدام مقياس تشابه جيب التمام. وأخيرًا، تم إجراء عملية التنبؤ بالتقييم باستخدام KNN استنادًا إلى تقييمات أقرب المستخدمين المنتمين إلى نفس المجموعة والتي تتمتع بقيمة مساعدة (helpfulness) عالية.

تم إجراء تجارب واسعة النطاق بواسطة النظام المقترح على مجموعتي بيانات أمازون: Amazon Electronics وAmazon Fine Food. تظهر النتائج التجريبية أن نموذج RS-TRDL الخاص بنا قد تجاوز جميع طرق المقارنة مع البحوث التي تمت مراجعتها في عملية التنبؤ بالتقييم لكلا المهمتين الذي تم تصميمه لأداءها. تجدر الإشارة إلى الأداء المتسق للنموذج عبر كلا المهمتين، كما يتضح من نطاق التحسين الذي يتراوح بين 0.24% إلى 34.32% لتخفيف مهمة مشكلة البداية الباردة للمستخدم ومن 3.21% إلى 58.7% لتخفيف مشكلة التناثر وتعزيز مهمة نموذج الموصي. دعمت هذه التجارب فكرة أن دمج أنظمة التوصية وتحليل المشاعر سيكون له مزايا كبيرة.

جامعة كربلاء
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

نظام التوصية المبني على تنقيب الآراء باستخدام تقنيات التعلم الآلي

رسالة ماجستير
مقدمة الى مجلس كلية علوم الحاسوب وتكنولوجيا المعلومات / جامعة كربلاء وهي جزء من متطلبات نيل درجة الماجستير في علوم الحاسوب

كتبت بواسطة
آمنة ناهض عبد الامير محمد

بإشـراف
أ.م.د محسن حسن حسين

1445 هـ

2024 م