

Bidirectional Transformers driven Contextual Sequential Recommendation with Contrastive Learning



Author

Asima Bashir

Registration No: 00000431940

Supervisor:

Prof Dr. Naima Iltaf

A thesis submitted to the faculty of Computer Software Engineering Department,
Military College of Signals, National University of Sciences and Technology, Islamabad
Pakistan in partial fulfillment of the requirements for the degree of MS in
Software Engineering

(July 2024)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Ms **Asima Bashir**, Registration No. **0000431940**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: Naima

Name of Supervisor: Prof Dr. Naima Iltaf

Date: 10 July 2024

Signature (HoD): Asif Masood

Date: 30/7/24

Signature (Dean/Principal): Asif Masood

Date: 31/7/24

Brig
Dean, MCS (NUST)
(Asif Masood, PhD)

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY
MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by Asima Bashir, Regn No 00000431940 Titled: "Bidirectional Transformers driven Contextual Sequential Recommendation with Contrastive Learning" be accepted in partial fulfillment of the requirements for the award of MS Software Engineering degree.

Examination Committee Members

1. Name: Assoc Prof Dr. Ihtesham UI Islam

Signature: _____

2. Name: Asst Prof Mobeena Shahzad

Signature: _____

Co Supervisor: Dr. Usman Zia

Signature: _____

Supervisor's Name: Prof Dr. Naima Iltaf

Signature: _____

Date: _____

 Head of Department

30/7/24
 Date

COUNTERSIGNED

Date: 31/7/24

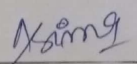
 Brig
 Dean, MCS (NUST)
 (Asif Masood, Phd)
 Dean

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled "**Bidirectional Transformers driven Contextual Sequential Recommendation with Contrastive Learning**" was conducted by Ms Asima Bashir under the supervision of Prof Dr. Naima Iltaf.

No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Computer Software Engineering of Military College of Signals** in partial fulfillment of the requirements for the degree of Master of Science in Field of **Software Engineering** Department of **Computer Software Engineering**, National University of Sciences and Technology, Islamabad.

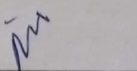
Student Name: Asima Bashir

Signature: 

Examination Committee:

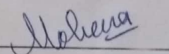
a) External Examiner 1: Dr. Ihtesham Ul Islam

Assoc Prof, Dept of CSE, MCS

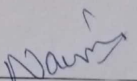
Signature: 

b) External Examiner 2: Mobeena Shahzad

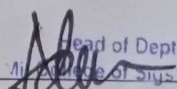
Asst Prof, Dept of CSE, MCS

Signature: 

Name of Supervisor: Prof Dr. Naima Iltaf

Signature: 

Name of Dean/HOD: Brig Adnan Ahmad Khan, PhD

Signature: 
Brig
Head of Dept of CSE
Military College of Signals (NUST)

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled “**Bidirectional Transformers driven Sequential Recommendation with Contrastive Learning**” is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature:



Name: **Asima Bashir**

Date: 10 July 2024

AUTHOR'S DECLARATION

I Asima Bashir hereby state that my MS thesis titled “**Bidirectional Transformers driven Contextual Sequential Recommendation with Contrastive Learning**” is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Student Signature:



Name: **Asima Bashir**

Date: 10 July 2024

DEDICATION

"In the name of Allah, the most Beneficent, the most Merciful"

Glory be to Allah Almighty, the Creator and Sustainer of the Universe, the Omnipotent and the Omnipresent. There is nothing I could have accomplished without His guidance and blessings. I dedicate this thesis to my family, friends, and teachers, particularly my parents, who supported me each step of the way.

ACKNOWLEDGEMENTS

All praises to Allah for the strengths and His blessing in completing this thesis.

I would like to convey my gratitude to my supervisor, Prof Dr.Naima Iltaf, PhD, for her supervision and constant support. Her invaluable help of constructive comments and suggestions throughout the experimental and thesis works are major contributions to the success of this research.

Moreover, I am highly thankful to my family, friends and teachers, especially my parents. They have always stood by my dreams and aspirations and have been a great source of inspiration for me. I would like to thank them for all their care, love and support through my times of stress and excitement.

ABSTRACT

Contrastive learning (CL) with Transformer-based sequence encoders offers a robust framework for sequential recommendation by effectively addressing data noise and sparsity issues. By utilizing the advantages of CL, these models are able to learn rich representations from sequential user interactions, leading to improved recommendation and user satisfaction. However, recent CL methods are affected by two limitations. Firstly, CL approaches are mainly process input sequences in single direction i.e left to-right which is sub-optimal for sequential prediction tasks because user historical interactions might not be in a fixed single direction. Secondly, these models focus on designing CL objectives based solely on input sequence, overlooking the valuable self-supervision signals available as contextual information of descriptive text. To address these limitations, this research proposes a novel framework called **Bidirectional Transformers driven Contextual sequential Recommendation with Contrastive Learning (CCLRec)**. Specifically, bidirectional Transformers are extended to incorporate auxiliary information by using sentence embedding formulated from item’s textual description. Next, we introduce the rolling glass step technique for handling lengthy user sequence and descriptive features of respective item, which enables more refined partitioning of user sequences. Next the cloze task mask, random occlusion and the dropout mask are fused for producing high standard of positive samples to demonstrate better performance for contrastive learning objective. Comprehensive experiments upon three benchmark datasets show remarkable improvements when correlating with other similar contemporary models.

Keywords: Contextual Sequential Recommendation, Bidirectional Transformers, Contrastive Learning, Auxiliary contextual Information

Contents

ABSTRACT	IX
LIST OF TABLES	XII
LIST OF FIGURES	XIII
LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS	XIV
1 Introduction	1
1.1 Research Objectives	5
1.1.1 Research Contribution	6
1.2 Outline Report	6
2 Literature Review	8
2.1 Sequential Recommendation System	8
2.2 Context driven Recommendation	12
2.3 Contrastive Learning based Recommendation	14
3 Proposed Framework	19
3.1 Framework Architecture	19
3.1.1 Mathematical Formulation of Proposed Model	21
3.2 Embedding Layer	21
3.2.1 Sequence Embedding	22
3.2.2 Descriptive Text Embedding	23
3.3 Context aware Bidirectional Transformers	24
3.3.1 Context aware Self attention	25
3.3.2 Feed Forward Network	26
3.4 Output Layer	28
3.5 Context driven Contrastive Learning	29
3.6 Training	30

4	Experimental Results and Analysis	31
4.1	Experiments	31
4.1.1	Datasets Pre-Processing	31
4.1.2	Evaluation Metrics	32
4.1.3	Baselines	33
4.1.3.1	Sequential Models	33
4.1.3.2	Context driven sequential methods	34
4.1.3.3	Sequential models with contrastive learning	34
4.1.4	Implementation Details	35
4.2	Comprehensive Performance Analysis	36
4.3	Hyperparameter Study	37
4.3.1	Number of Positive Samples	38
4.3.2	Dropout Ratio	39
4.3.3	Hidden Dimensionality d	39
4.3.4	Rolling Glass Step	40
4.4	Computational Complexity	40
4.5	Ablation Study	41
4.5.1	Impact of Proposed Modules	41
4.5.2	Impact of Auxiliary Contextual Task μ	42
5	Summary of Research Work	43
6	Conclusion	45

List of Tables

2.1	Comparison and Analysis of Present Studies	16
4.1	Datasets statistics after Pre-Processing	32
4.2	Comprehensive performance analysis of proposed model with refer- enced models for next item recommendations. The highest scores are shown in bold, while the second place scores are underlined.	37
4.3	Ablation Study on our proposed modules. The outcomes depict that proposed techniques enhanced the overall performance.	42

List of Figures

1.1	Relevant movie is recommended with the help of auxiliary contextual information available in the form of movie categories.	4
2.1	Next item is recommended with the help of sequence of user interactions.	9
2.2	Framework Architecture of Transformers	10
2.3	Framework Architecture of Bert4Rec	11
2.4	Context driven Recommendation System.	13
2.5	Core concept of Contrastive Learning where similar samples are placed together where as dissimilar samples are pushed away in embedding space.	14
3.1	Model Architecture of Contextual Sequential RS using Contrastive Learning.	20
3.2	Framework Architecture of Sentence-BERT	23
3.3	The proposed Framework Architecture of CSA module.	25
4.1	$NDCG@10$ performance analysis of three benchmark datasets with respect to other hyperparameters.	38
4.2	Ablation study ($NDCG@10$) on the impact of proposed modules and auxiliary contextual task	41

LIST OF ABBREVIATIONS AND ACRONYMS

Recommender Systems	RS
Sequential Recommender System	SRS
Bidirection Encoder Representation for Transformer	BERT
Contrastive Learning	CL
Deep Learning	DL
Recurrent Neural Network	RNN
Gated Recurrent Unit	GRU
Mask Item Prediction	MIP
Convolutional Neural Network	CNN
Long Short-term Memory	LSTM
Markov Chain	MC
Positional Embedding	PE
Learning rate	lr
Context driven Contrastive Learning Objective	CCL
Hit Ratio	HR
Normalized Discounted Cumulative Gain	NDCG

Chapter 1

Introduction

Recommender systems (RS) predict the users' future preferences by characterizing the users' intent that are usually dynamic in nature. These systems are essential tools in the modern data-driven landscape, helping users find relevant items in vast datasets. They enhance user experience, increase engagement, and drive business value across various industries. RS are widely being used in e-commerce platforms and online media streaming websites to mitigate the efforts by the user in this information overload world. Users' interests are usually not stable and keeps on changing with time. This temporal aspect is crucial in acquiring user dynamic preferences. For the purpose of identifying user intents more precisely, numerous sequential recommendation techniques have been introduced in recent past that uses user's previous historical interactions [1].

The intention of sequential recommendation (SR) models works in two phases. First they gather the sequence of past objects from user's history and then projecting the most relevant and accurate interaction for each user. Traditionally, researches exploit Markov Chain model to anticipate future item based on recent items in the user historical behavior. These models are mathematical systems that undergo transitions from one state to another in a state space. They are named after the Russian mathematician Andrey Markov and are characterized by the property that the next state depends only on the current state and not on the sequence of events that preceded it. [2], [3]. The

significant advancement of machine learning has been marked by rapid advancements and innovations. CNNs revolutionized image processing and are primarily designed for spatial data such as images. CNNs are adept at capturing local patterns through convolutional layers, which apply filters to detect features like edges and textures. This hierarchical feature extraction makes CNNs highly effective for image classification, object detection, and segmentation tasks. On the other hand, RNNs are tailored for sequential data, such as text and time series, where the order of the data points matters [4], [5], [6], [7]. However, CNN based models tend to overlook global features and RNNs show difficulty in capturing inter items dependencies. More recently, transformers and attention mechanisms have taken center stage, pushing the boundaries of what is possible in natural language processing and beyond. Self attention mechanisms from transformers encode sequential behaviors in efficient manner [8]. SASRec predominantly outperformed other SR models through unidirectional transformers. SASRec uses self-attention mechanisms inspired by the Transformer architecture to capture the relevance of past interactions in predicting future actions. By focusing on relevant items in a user’s history, it effectively models long-term dependencies without the limitations of recurrent networks [9]. BERT4Rec [10], on the other hand, applies the BERT (Bidirectional Encoder Representations from Transformers) model to sequential recommendation, utilizing a bidirectional transformer to consider both past and future contexts of a user’s interaction sequence. This bidirectional approach allows Bert4Rec to capture complex patterns and dependencies within the data, providing more accurate and context-aware recommendations. KeBERT4Rec [11] integrates keywords along with the item identifier in BERT4Rec model by concatenating the keyword representation with item and its positional representations. However, this model uses one hot encoding technique to generate the keyword vector, thus neglecting the contextual meaning of keywords. Another model, FDSA [12] utilizes the attribute information by applying a separate self-attention block for item in the user history as well as for the features. Thus, most SR models including SASRec and BERT4Rec consider only

implicit or explicit feedback based on item identifier for next item recommendation and neglect the auxiliary data (textual descriptions, keywords, reviews etc). Accuracy of next item prediction task in SR can be improved by incorporating additional information.

In spite of the efficacy and notable outcomes in different natural language processing models, researchers face challenge due to the existing sparse interaction matrices and dataset interaction noises. To address these issues in SR, contrastive learning (CL) has been introduced to models based on transformers to increase the standard of learned representations by utilizing data augmentation techniques. These techniques transform the input data into different perspectives, helping the model learn invariant properties of the sequence. The core idea is to pull positive views (augmentations of the same input) closer together while pushing negative views (augmentations from different inputs) farther apart. This technique helps capture the nuanced preferences of users by focusing on the contrast between what users interact with and what they do not. In the context of sequential recommendation, contrastive learning can effectively enhance the model’s ability to understand temporal patterns and user behavior by generating robust embeddings that represent the sequence of interactions. By employing methods such as data augmentation to create diverse positive and negative samples, contrastive learning enables models to learn richer and more discriminative representations, leading to improved recommendation accuracy and the ability to generalize better to new or unseen sequences. This approach has proven to be particularly powerful in scenarios where explicit feedback is sparse, as it leverages implicit signals more effectively to infer user preferences. CL4SRec [13] creates positive and negative samples through sequence level innovative augmentation techniques. This involves transforming the entire sequence in various ways to ensure that the model learns robust and invariant representations. CoSeRec (Contrastive Self-supervised Sequential Recommendation) [14] is an advanced framework that applies contrastive learning principles specifically to sequential recommendation tasks. CoSeRec leverages self-supervised learning by generating

multiple augmented views of a user’s interaction sequence through techniques such as cropping, masking, and reordering. DuoRec [15] utilizes unsupervised dropout and supervised positive sampling to create positive samples. ICLRec [16] devises expectation maximization to iteratively refine the representations by maximizing the likelihood of observing the positive samples and minimizing the likelihood of observing the negative samples. It is an advanced approach in the field of recommender systems, particularly tailored to capture the underlying intentions behind user interactions. Unlike traditional contrastive learning, which primarily focuses on distinguishing between positive and negative examples to learn effective representations, intention-oriented contrastive learning aims to understand and model the specific intentions driving user behavior. CBiT [17] combines dropout mask with cloze task mask to generate diverse and informative pairs of views, enhancing the learning of invariant features uses a masking operation.

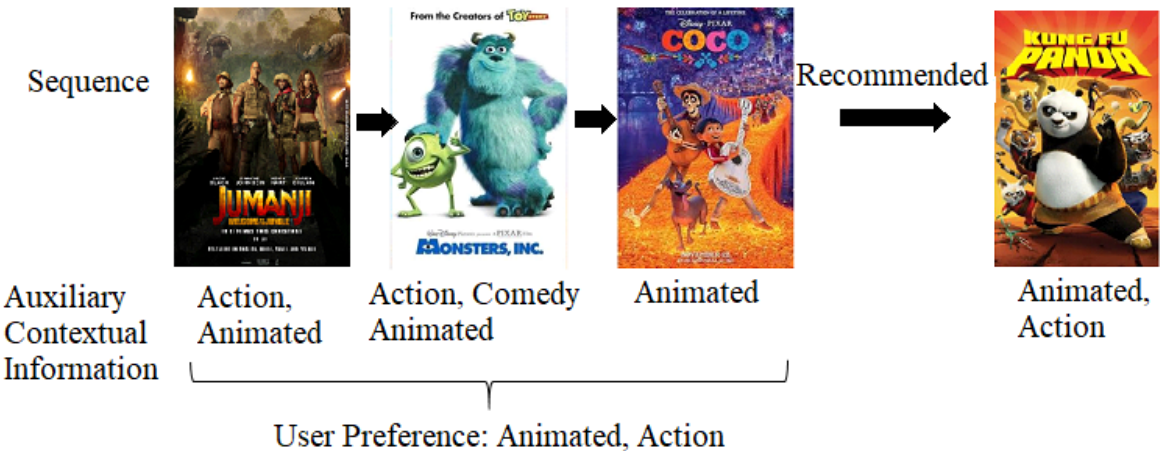


Figure 1.1: Relevant movie is recommended with the help of auxiliary contextual information available in the form of movie categories.

Although, these SR models including CL based methods show significant performance gain, however, they do not exploit contextual features (textual descriptions, keywords, reviews etc) to generate meaningful representations neglecting the rich sig-

nals from auxiliary data . For instance, Figure 1.1 depicts the sequence of movies watched by a user with different categories. Identification of relevance among these set of movies only having the sequential information is difficult. But we can accurately infer the preference of user and recommendation of appropriate movies with the auxiliary data (textual descriptions, keywords, reviews etc). Moreover, deciding maximum length of descriptive features and user historical interactions is crucial for recommender systems due to computational constraints and the need to efficiently process and analyze large amounts of textual data in order to capture more fine-grained contextual information. Therefore we anticipate that by incorporating additional information in CL methods and handling lengthy input sequences , prediction accuracy of next items can be increased particularly under sparse situations.

1.1 Research Objectives

Objectives of the research work are as under:

- To design an efficient and accurate bidirectional transformer based context aware self attentions module embedded with item rich features.
- To design a robust contrastive learning objective that creates rich set of positive samples by leveraging cloze task mask, random occlusion mask and dropout mask.
- To introduce rolling glass step technique to handle lengthy user sequences and descriptive features of respective item to optimize the computational cost and capture more fine grained contextual information.
- Compare the suggested model with baseline models and recently developed state-of-the-art techniques.
- To improve predictions accuracy in sequential recommender systems.

1.1.1 Research Contribution

Within the framework of research, a novel framework called Bidirectional Transformers driven Contextual Sequential Recommendation with Contrastive Learning (CCLRec) has been designed to address the challenge of overlooking the valuable self-supervision signals available as contextual auxiliary information of descriptive features in contrastive learning methods. Significant contributions of the research are highlighted as follows:

- a novel framework architecture which efficiently incorporates the auxiliary contextual information into the user behavior and considers the supervision signals constructed from the descriptive text for CL.
- a novel context driven sequence encoder which generates contextual embedding of textual description of the items using Sentence-BERT and simultaneously captures the intrinsic association between the respective item and the respective descriptive text by combining the outputs of multiple attention mechanisms into a single representation.
- a novel context driven contrastive learning (CLL) objective, that learns effective representations from augmented versions of the original data, through three types of augmentations: the cloze task mask, random occlusion mask and the dropout mask.
- evaluation and performance comparison of the proposed model with existing state-of-the-art techniques to achieve significant performance.

1.2 Outline Report

This thesis is divided into 5 chapters as under:

- Chapter 1: This chapter contains introduction, objectives and the contributions made in this research.

- Chapter 2: This chapter presents review of existing relevant research work done in our domain.
- Chapter 3: This chapter presents the proposed framework architecture in detail.
- Chapter 4: This chapter discusses the experiment details and analysis of the results by comparing with baseline models along with the brief explanation of the evaluation metrics being used to evaluate the model.
- Chapter 5: This chapter outlines the summary of research work.
- Chapter 6: This chapter concludes the report and highlights the direction for future research in our domain.

Chapter 2

Literature Review

2.1 Sequential Recommendation System

Sequential recommendation system (SRS) is a specialized type of recommendation system (RS) that leverages the user interaction sequences to anticipate and recommend an item a user might be interested in near future[18]. Unlike traditional recommendation systems that rely on static user profiles or item features, sequential recommendation systems focus on the temporal dynamics and patterns in user interactions. These systems leverage techniques from machine learning, particularly sequence modeling approaches like recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and more recently, transformer models. By capturing the order and context of past behaviors, sequential recommendation systems can provide more personalized and timely suggestions. They are particularly effective in domains such as e-commerce, content streaming, and social media, where user preferences can change rapidly and context plays a crucial role in decision-making. The ability to adapt to evolving user interests and provide relevant recommendations in real-time makes sequential recommendation systems a powerful tool for enhancing user engagement and satisfaction. SRS aims to recommend future product by considering historical behavior of users as shown in Figure 2.1. This historical behavior is also known as next item prediction.

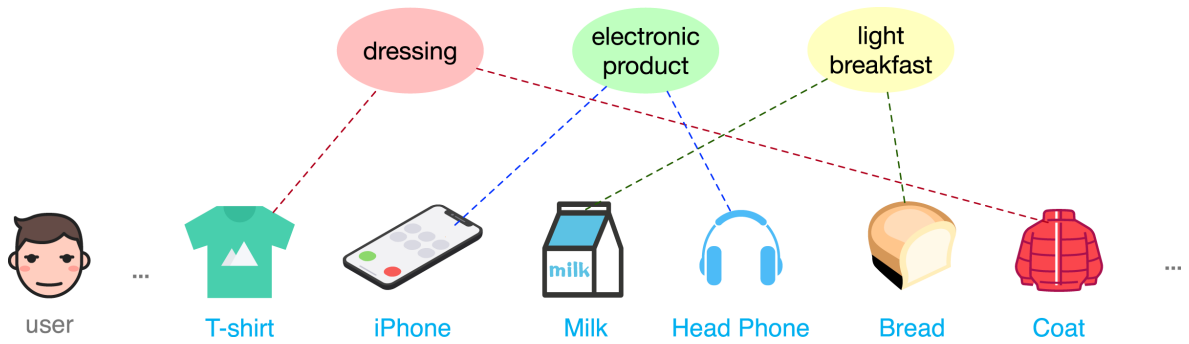


Figure 2.1: Next item is recommended with the help of sequence of user interactions.

Earlier, the SRS were introduced using Markov Chains (MC) models for capturing sequential patterns from the user historical preferences [19]. The next item preferred by the users are predicted depending upon the last item, thus interpreting only the adjoining sequential behavior. Models based on RNNs leverage Gated Recurrent Unit (GRU) [20] and Long Short Term Memory (LSTM) [21] to show substantial performance gain for SR. RNNs enforce rigid sequential patterns for encoding user preferences for making predictions. Besides RNN, a number of Convolutional Neural Network (CNN) [4] based RS have also been introduced that also target problems related to the SR. Transformer models based on attention mechanism [8] have revolutionized the field of deep learning with their extraordinary performance across various domains such as text classification [22], image captioning [23], and machine translation. These models have become the new standard because of their capability to effectively handle high-range dependencies and parallelized computations.

Transformers, primarily modeled for natural language processing, have shown revolutionary impact in the field of sequential recommendations. For modeling the sequential data, only the encoder part of the Transformer is used that aims at mapping the items sequences which represent the interaction history of user into the sequence of vector representations. Transformers leverage the self-attention mechanism, which allows the model to focus on different parts of the input sequence when making predictions. This helps in capturing long-range dependencies and relationships within the sequence,

which is crucial for understanding user behavior patterns. Using Transformer in SR, a sequence of items are passed as input that is encoded through embedding layer followed by concatenating with the positional embedding (vector representations that learns the item’s placement in the sequence) and processed through Transformer layer. A single Transformer block as illustrated in Figure 2.2 comprises of a ”multi-head self attention” layer and a ”position-wise feed forward” layer [8].

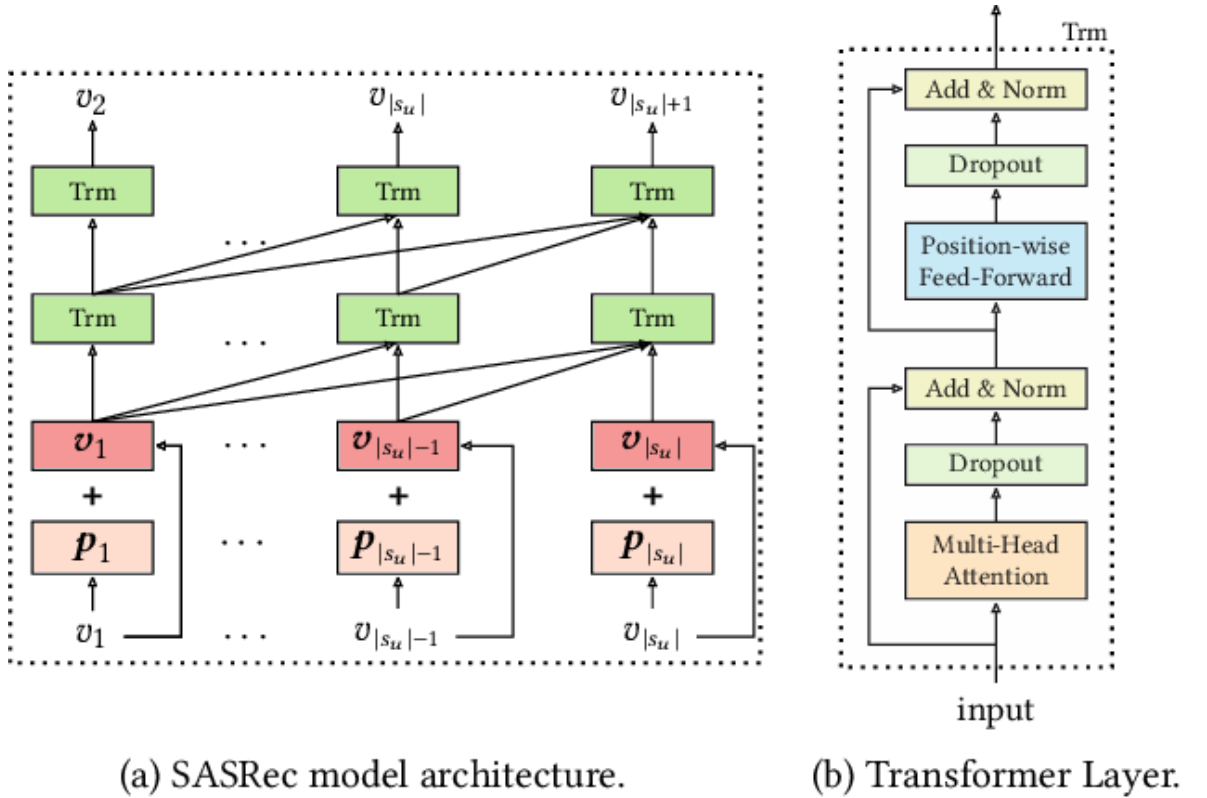


Figure 2.2: Framework Architecture of Transformers

SASRec exploits the item identifiers for modelling the user interaction. It leverages the transformer architecture, specifically utilizing the self-attention mechanism to model the sequential behavior of users. Unlike traditional approaches that rely on RNNs or LSTMs, SASRec can capture long-range dependencies and interactions within user behavior sequences more effectively. By focusing on the most relevant parts of a user’s interaction history, SASRec provides more accurate and personalized

recommendations. The model’s ability to process sequences in parallel also enhances computational efficiency, making it suitable for large-scale recommendation systems [9].

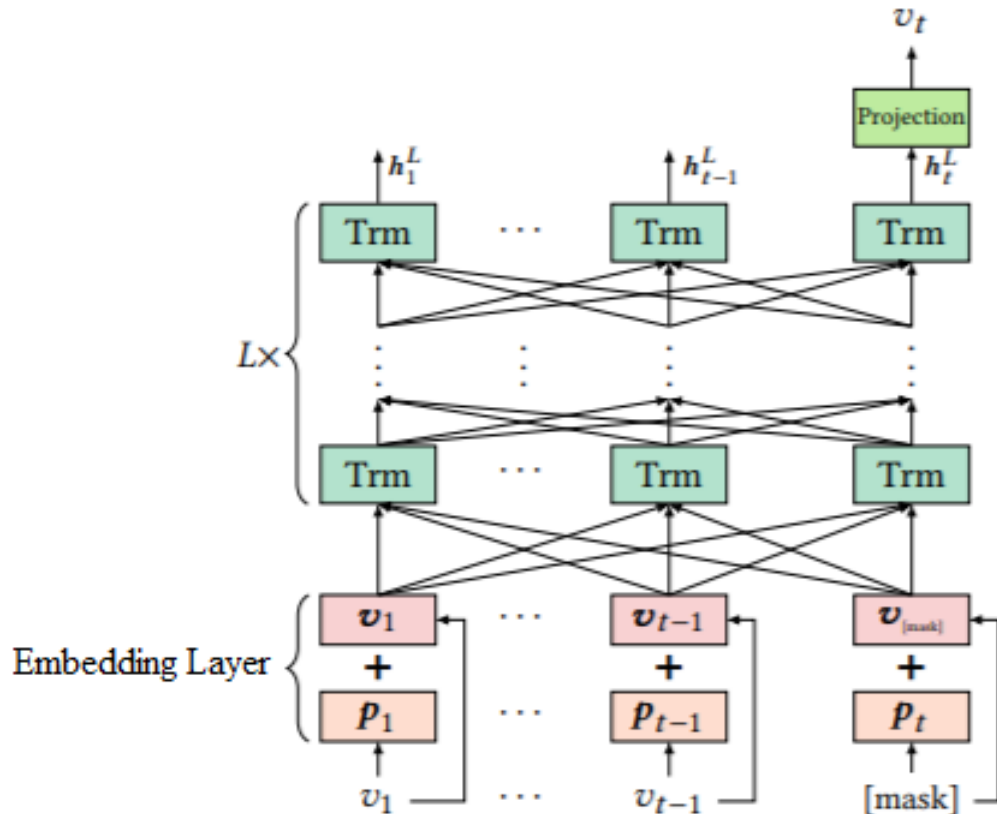


Figure 2.3: Framework Architecture of Bert4Rec

On the other hand, unlike traditional sequential recommendation models that typically predict the next item in a sequence based on past interactions, BERT4Rec utilizes bidirectional self-attention to capture the dependencies in both forward and backward directions within user behavior sequences as shown in Figure 2.3. This allows BERT4Rec to better understand the context and relationships among all items in a sequence, leading to more accurate and robust recommendations. The model employs a masked item prediction task during training, where model learns to anticipate the

masked items based on their context by masking certain items in the sequence. This pre-training approach enables BERT4Rec to effectively capture complex patterns and nuances in user behavior [10]. KeBERT4Rec uses keywords along with item identifiers for next item prediction [11]. However, keywords representations are not extracted through any of the contextual embedding technique, thus losing the context meaning. These sequential recommendation techniques exploit the item identifiers for next item recommendation. [24] proposed *S³Rec*, a self supervised SR model that utilized the attribute data of item to learn the correlation among them. A feature level deeper self attentive model [12] exploits segregated attention blocks for input sequence and respective features to predict next item. LSSA [25] proposes a novel architecture to fetch both long-term user choices and short-term sequential user interactions through attention mechanism. SR-GNN (Session-based Recommendation with Graph Neural Networks) [26] and GC-SAN (Graph Convolutional Self-Attention Network) [27] leverage GNNs to capture global dependencies and transitions across different sessions. DDGHM (Dynamic Graph-based Hybrid Model) [28] integrates attention scores and dynamic graph modeling to improve the capability of model to take into account the specific aspects of the data from multiple domains. Models like CL4SRec [13], CoSeRec [14], and DuoRec [15] leverage CL to achieve better generalization and robustness in sequence prediction tasks.

2.2 Context driven Recommendation

Context-driven recommendation focuses on developing recommendation systems that considers various contextual features for provision of more personalized and pertinent user recommendations. It is an advanced type of recommendation engine that takes into account the contextual information surrounding the user and the items being recommended. Different from traditional recommendation systems that primarily focus on user-item interactions and historical data, context-driven systems consider various contextual factors such as time, location, mood, device type, and current activity. This

approach allows for more personalized and relevant recommendations, as it adapts to the user’s current situation and preferences.

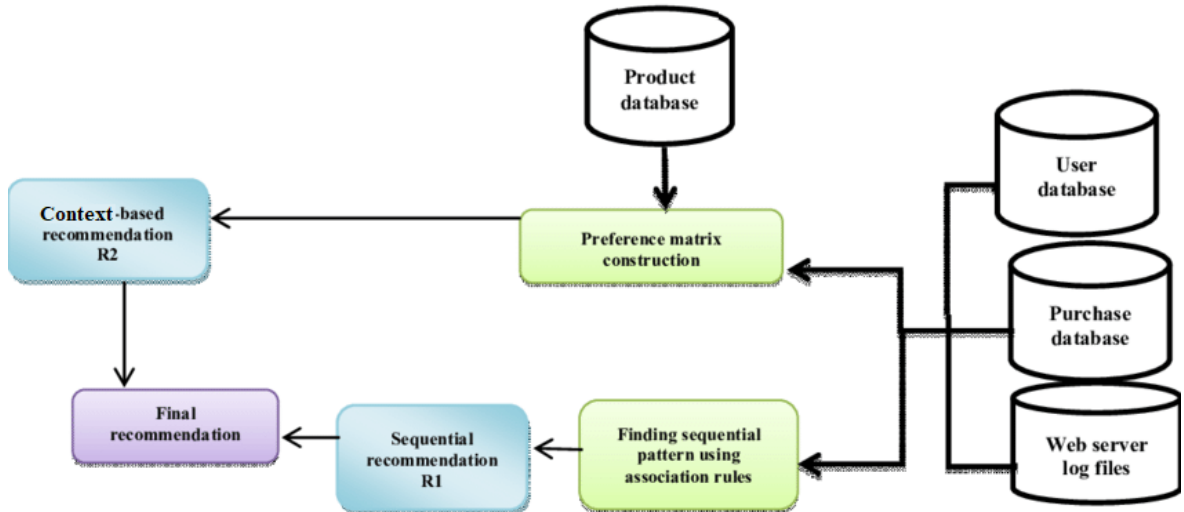


Figure 2.4: Context driven Recommendation System.

A context-driven recommendation system is illustrated in Figure 2.2. For a music streaming service It might suggests different playlists depending on whether the user is at the gym, commuting, or relaxing at home. Similarly, an e-commerce platform could tailor its product suggestions based on the user’s browsing time and location, offering different items during a lunchtime break at work compared to a weekend evening at home. By integrating contextual data, these systems enhance user satisfaction and engagement, providing a more intuitive and seamless experience. [29], [30]. S^3Rec , a self supervised SR model that utilized the attribute data of item to learn the correlation among them [24]. KeBERT4Rec [11] leverages the keyword by integrating them with item identifier for the prediction of next item in sequence. However, keywords representations are not extracted through any of the contextual embedding technique, thus losing the context meanings. A feature level deeper self attentive model [12] introduced by exploits segregated attention blocks for items and their associated features to predict next item. GRU4RecBE, an extension of GRU4Rec [5] model uses

the rich item features embedding generated through pre-trained BERT and processed through the GRU-RNN layer [26]. FCLRec introduces a CL objective that enhances the model’s ability to differentiate between similar and dissimilar sequences by leveraging feature-specific information [31]. However type and length of features are not mentioned thus lacking the contextual meaning. Different from these works, our proposed model combines auxiliary informaion and item identifiers to create embeddings using the Sentence BERT embedding technique. This enhances item recommendation and prediction accuracy by capturing contextualized representations.

2.3 Contrastive Learning based Recommendation

Contrastive learning (CL) is self-supervised learning based methodology used to learn representations by contrasting examples having positive and negative pairs.

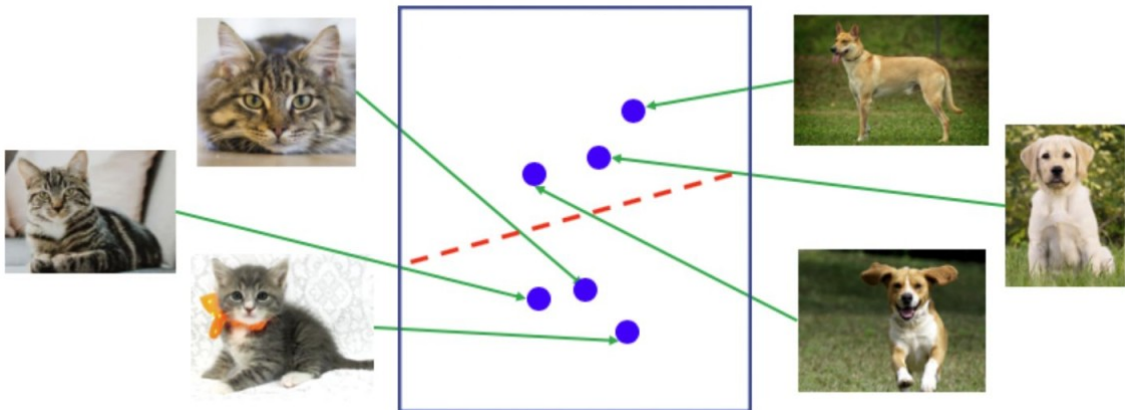


Figure 2.5: Core concept of Contrastive Learning where similar samples are places together where as dissimilar samples are pushed away in embedding space.

The core concept is to draw similar (positive) examples closer together in the embedding space while distancing dissimilar (negative) examples as illustrated in Figure 2.3. This approach has become particularly popular in tasks where labeled data is scarce or expensive to obtain, and it is extensively used in numerous domains such as natural language processing, computer vision, and sequential recommendation systems. [31].

Moreover, it has significantly improved the process of future item prediction by focusing on the similarities and differences between sequences. SimCLR’s augmentation strategy can help capture various aspects of user behavior, while Siamese networks’ pair-based approach can leverage labeled data to refine recommendations [32]. CL4SRec [13] introduces CL into the SR process to enhance the quality of sequence representations. CoSeRec [14] leverages self-supervised learning by introducing contrastive objectives to improve sequence representation learning. DuoRec [15] adopts model augmentation as a approach to improve item embedding distribution and overcome representation degeneration problem in contrastive learning. ICLRec (Intent Contrastive Learning for Sequential Recommendation) [16] is an advanced model designed to enhance the quality of recommendations by focusing on the underlying user intents within interaction sequences. By leveraging contrastive learning techniques, ICLRec aims to learn robust representations that capture the nuanced preferences and intents of users. CBiT [17] introduces a BERT-based architecture with CL to capture temporal dynamics in SR. ContraRec [33] leverages CL objective by combining supervision signals from conceptual and computational level.

Detailed overview of research findings is displayed in Table 2.1

Table 2.1. Comparison and Analysis of Present Studies

Title	Methodology	Model
MC-based SRS [2]	Captures sequential patterns from user historical preferences using Markov Chains (MC).	Markov Chains
RNN-based SRS [5]	Uses Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) to encode preferences of user for predictions.	GRU, LSTM
CNN-based SRS [4]	Utilizes Convolutional Neural Networks (CNNs) to address sequential recommendation problems.	CNN
Transformer-based SRS [8], [9], [10]	Leverages the encoder part of Transformers to map sequences of user interactions to vector representations, utilizing multi-head self-attention and position-wise feed-forward layers.	Transformer, SASRec, BERT4Rec
KeBERT4Rec: Keyword-Augmented BERT4Rec for Sequential Recommendation [11]	Combines keywords with item identifiers for next item prediction.	KeBERT4Rec
S3Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization [24]	A self-supervised model utilizing item attribute data to learn correlations among items.	S3Rec

Title	Methodology	Model
Feature-level Deeper Self-Attentive Model for Sequential Recommendation [12]	Uses segregated attention blocks for input sequences and features to predict the next item.	Feature-level deeper self attentive model
LSSA: Long- and Short-term Self-Attention for Sequential Recommendation [25]	Captures long-term preferences and short-term sequential dynamics through attention mechanisms.	LSSA
SR-GNN: Session-based Recommendation with Graph Neural Networks [26]	Combines Graph Neural Networks (GNNs) with self-attention to fetch global dependencies and transitions.	SR-GNN
GC-SAN: Graph Convolutional Self-Attention Network for Sequential Recommendation [27]	Utilizes GNNs with self-attention to capture both local and global transitions in sequences.	GC-SAN
DDGHM: Dynamic Graph-based Hybrid Model for Cross-Domain Sequential Recommendation [28]	Integrates dynamic graph modeling with attention scores for cross-domain sequential recommendation.	DDGHM
CL4SRec: Contrastive Learning for Sequential Recommendation [13]	Introduces contrastive learning to enhance sequence representation quality in sequential recommendation.	CL4SRec
CoSeRec: Contrastive Self-Supervised Learning for Sequential Recommendation [14]	Leverages self-supervised learning with contrastive objectives to improve sequence representation learning.	CoSeRec

Title	Methodology	Model
DuoRec: Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation [15]	Adopts model augmentation as a approach to improve item embedding distribution and overcome representation degeneration problem in contrastive learning	DuoRec
ICLRec: Intent Contrastive Learning for Sequential Recommendation [16]	Enhances recommendation quality by focusing on user intents within interaction sequences using contrastive learning techniques.	ICLRec
CBiT: Contrastive BERT-based Temporal Dynamics for Sequential Recommendation [17]	Introduces a BERT-based architecture with contrastive learning to capture temporal dynamics in sequential recommendations.	CBiT
ContraRec: Contrastive Learning for Sequential Recommendation by Combining Conceptual and Computational Supervision Signals [34]	Leverages contrastive learning objectives by combining supervision signals from conceptual and computational levels to improve sequence prediction.	ContraRec

Chapter 3

Proposed Framework

3.1 Framework Architecture

In this chapter, the proposed framework "Bidirectional transformers driven Contextual sequential recommendation with Contrastive Learning (CCLRec)" is illustrated. The proposed model is developed based upon Transformer architecture that adapted the deep bidirectional BERT model for SR prediction task as described in Figure 3.1. For each user sequence s_u , respective auxiliary information and m different masked sequences are generated that is the textual description of the items in the form of sentences. Before passing these sequence of items to the model proposed, the auxiliary features of these items are passed as input to the Sentence-BERT to extract the contextual dense feature representation. These dense embedding are extracted prior to training phase to reduce the model training time.

Subsequently, during the training process, the auxiliary information's embeddings of items within a sequence are extracted and then passed to the context aware bidirectional transformers as shown in Figure 3.1 where these embeddings are then concatenated with the positional embedding and item's embedding to fetch the items sequential behavior. Only the encoder part of Transformer is used to compute the hidden representation using self attention mechanism for each item. These layers share information bidirectionally across each position in hierarchical manner. A final learned hidden representation is projected at output layer that contemplated the future item

recommendation for a user.

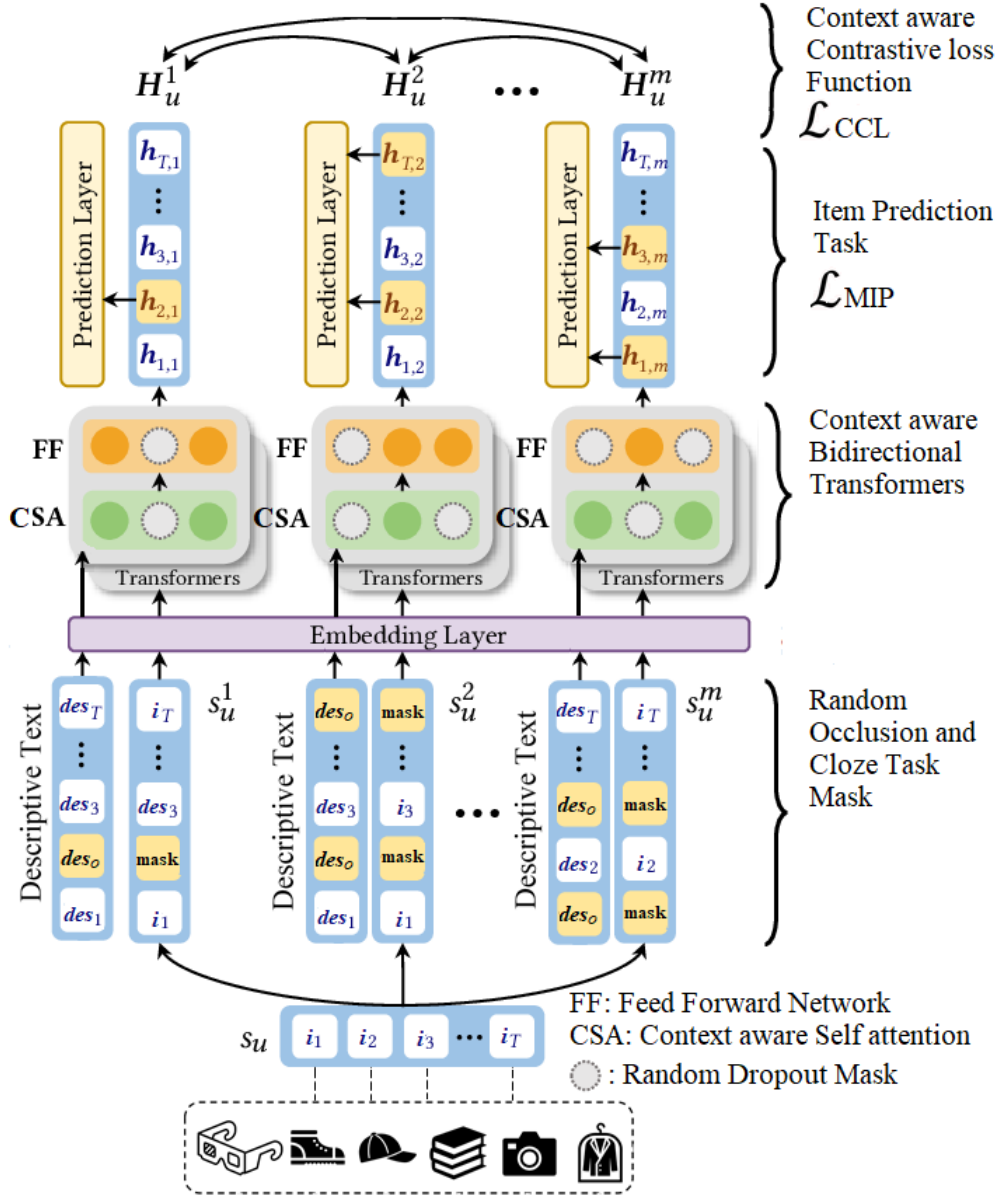


Figure 3.1: Model Architecture of Contextual Sequential RS using Contrastive Learning.

Numerous Experiments were performed using three bench mark datasets including movielens-1m, Amazon Beauty and Toy to establish the efficiency of the proposed model. The layers of proposed framework are assembled using embedding layer, trans-

former layer and the output layer.

3.1.1 Mathematical Formulation of Proposed Model

Let set of users U and items I be shown mathematically as:

$$\begin{aligned} U &= \{u_1, u_2, u_3, \dots, u_{|U|}\} \\ I &= \{i_1, i_2, i_3, \dots, i_{|I|}\} \end{aligned} \tag{3.1}$$

Each item has an item description in textual form (auxiliary information) represented as $TD = \{des_1, des_2, des_3, \dots, des_{|I|}\}$. User u interacts with items in a sequence in historical order is denoted as $S = \{i_1, i_2, i_3, \dots, i_n\}$, where i_n is a particular item from I that the user has acted upon previously. Provided the history of sequence S , Sequential Recommendation System (SRS) aims to predict the future item i_{n+1} in which a user is interested. The model predicts the probability of $i_{n+1} = i$ given S as:

$$p(i_{n+1} = i|S)$$

3.2 Embedding Layer

The embedding layer in CCLRec plays a crucial role in transforming input data into dense, continuous vectors that the model can process effectively. In CCLRec, the embedding layer consists of two main components: Sequence embedding that consists of item embeddings and positional embeddings, and descriptive text embeddings. Item embeddings constitute every unique dataset item as a high-dimensional vector, capturing semantic information about the items. Positional embeddings are added to these vectors to retain the order of items, which is essential for understanding the sequence of user interactions. Descriptive text embeddings are used to capture the semantics of items. By combining these embeddings, the CCLRec model gains a rich, context-aware representation of the input data, enabling it to capture complex patterns in user

behavior and make accurate sequential recommendations. This embedding layer forms the foundation for the subsequent transformer layers, allowing the model to process and learn from the embedded sequences effectively.

3.2.1 Sequence Embedding

By integrating both item embedding matrix $M \in \mathbb{R}^{I \times d}$ and positional embedding matrix $P \in \mathbb{R}^{L \times d}$, CCLRec constructs rich, context-aware hidden representations of sequences where L is maximum length of sequence and d is hidden dimensions. These representations enable the model to efficiently seize and utilize the items order and characteristics, improving the accuracy and relevance of recommendations. Thus, for item i at t timestamp, the input representation is shown as:

$$h_{0_t} = m_t + p_t, \quad 1 \leq t \leq L \quad (3.2)$$

where embedding vectors are m_i and p_t with respect to position t and item i . Each h_{0_t} is stacked together to construct the embedding of complete sequence

$$H_0 = [h_{0_1}, h_{0_2}, \dots, h_{0_t}, \dots, h_{0_L}].$$

In contrast to recent research, we address the limitation of fixed length of user sequence due to the computational constraints by introducing rolling glass of step size L over any lengthy user sequence and descriptive features at the training stage. The sequence is disintegrated into overlapping segments that fit within the maximum length constraint. This allows the model to process longer sequences incrementally while maintaining context continuity. Specifically, for a combined lengthy user sequence including item description s_d with $|s_d| > L$, we extract different sub sequences $\hat{s}_d^i = [i_{1+l}, i_{2+l}, \dots, i_{L+l}]$ as multiple input instances, where $l \in \{0, \Omega, 2\Omega, \dots, k\Omega\}$, $0 \leq k\Omega \leq |s_d| - L$, and Ω denotes the rolling glass step size which helps to preserve all the training data and process sequence of user historical behavior at a deep contextual level.

3.2.2 Descriptive Text Embedding

Although efficient recommendations can be made by making use of the positional embedding along with the item identifier embedding, thus memorizing the sequential order of the input. However, the pair alone does not describe the contextual representation of the input and does not recommend contextually especially under sparse conditions. Descriptive text embeddings are powerful representations that capture the semantic essence of text data in a dense, continuous vector space. These embeddings transform textual information into numerical vectors, enabling machine learning models to process and understand language at a deeper level. For instance, in a product recommendation system, descriptive text embeddings of product reviews, descriptions, and user comments can be utilized to discern the underlying themes and sentiments, facilitating more accurate and contextually relevant recommendations. By converting text into these meaningful numerical representations, descriptive text embeddings allow for sophisticated analysis and integration of textual data into various applications, enhancing the model's proficiency to engage in meaningful conversations and produce human communication patterns effectively. The proposed model utilizes the Sentence-BERT [35] for capturing contextual representation of the item descriptions.

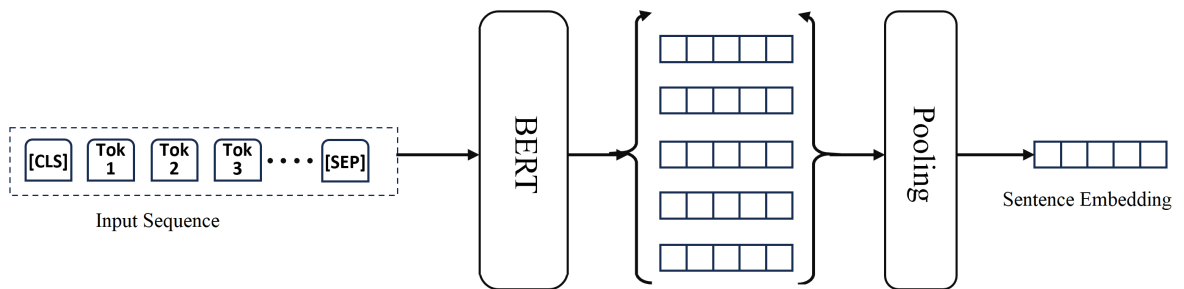


Figure 3.2: Framework Architecture of Sentence-BERT

The architecture of Sentence-BERT for extracting sentence embedding is depicted in Figure 3.2. Sentence-BERT (SBERT) is a powerful variant of BERT (Bidirectional Encoder Representations from Transformers) designed to generate high-quality sen-

tence embeddings for tasks requiring semantic understanding, such as semantic search, clustering, and paraphrase detection. Unlike the original BERT, which is optimized for token-level tasks and requires complex pairwise comparisons for sentence-level applications, SBERT modifies BERT. It creates fixed-size sentence embeddings by augmenting model with pooling operation after embedding layer. Input in the form of sentences or text of various length is injected to the selected SBERT model, which in turn generates contextualized word embedding for all input tokens in the sentence. Secondly, these word embedding are passed through a pooling layer to generate a fixed sized vector representation. Among various pooling options available, the model utilizes the mean pooling to produce a fixed dimensional output embedding vector. SBERT generates descriptive text embedding $c_t \in \mathbb{R}^{N \times d}$ for each associated item i_t where N represents dimensionality of sentence embedding.

3.3 Context aware Bidirectional Transformers

Based upon the transformer architecture, we utilize deep bidirectional Transformers based encoder to capture the interaction sequence of user from left and right directions. To augment our framework’s ability to handle both sequential and auxiliary information, we utilize the bidirectional transformer to integrate an innovative context aware self attention module. This module effectively captures the intricate association between sequence of items and their respective descriptive text simultaneously. This is achieved through self-attention layers, where each word in the input sequence is compared with every other word to calculate attention scores, and these scores are then utilized to produce weighted representations of the input. This allows the model to capture complex dependencies and nuanced meanings that would be difficult to identify with traditional sequential models.

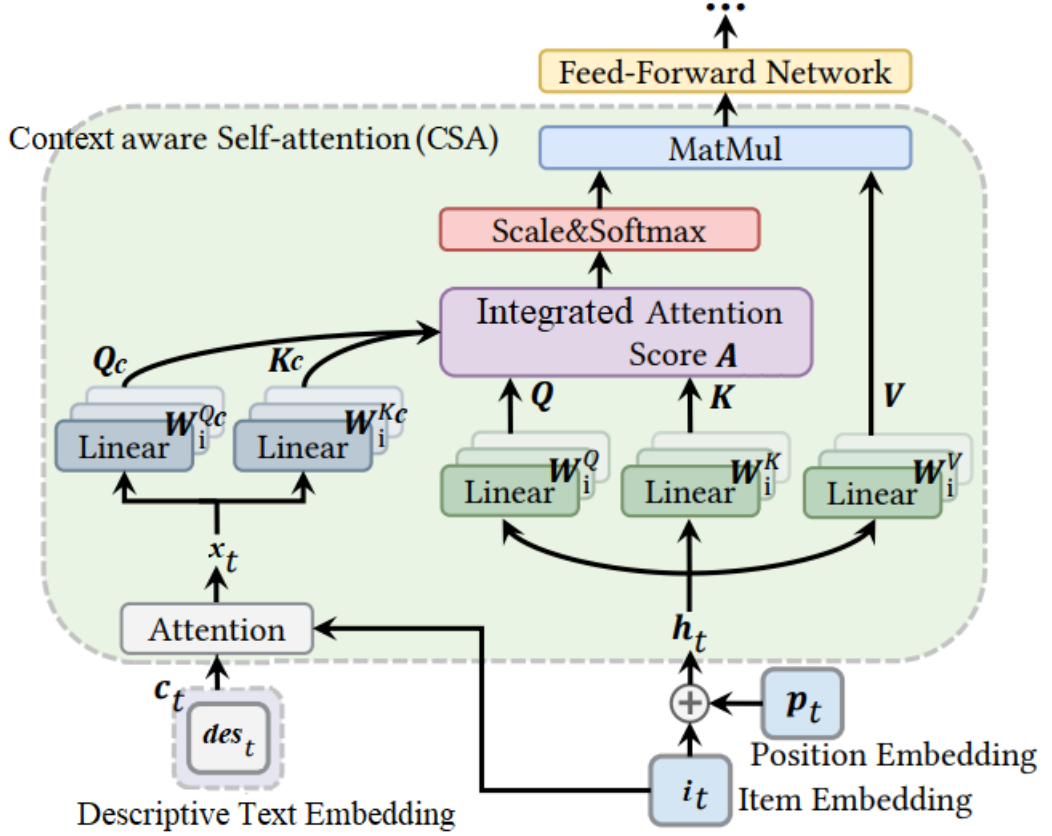


Figure 3.3: The proposed Framework Architecture of CSA module.

3.3.1 Context aware Self attention

To integrate and simultaneously design the intricate association between interaction sequences and their associated descriptive text, we introduce a novel context aware self attention (CSA) module as shown in Figure 3.3. This model leverages an advanced attention mechanism to effectively capture and model the dependencies between items and its corresponding descriptive text in a sequence, taking into account the context in which these elements appear. The attention mechanism works by assigning varying levels of weights, or importance to the input sequence of items. This enables the model to concentrate on the most relevant information when making predictions or generating output. In a context-aware setting, this means that the model can dynamically adjust its attention based on the surrounding context, enabling it to better understand and

interpret the relationships between words, sentences, or other sequential data. In this module, the attention layer obtains the context representation x_t for each item by processing the item embedding i_t as query which refers to its corresponding descriptive text embedding c_t .

$$x_t = \text{softmax} \left(\frac{(i_t W_C^Q)(c_t W_C^K)^\top}{\sqrt{d}} \right) (c_t W_C^V) \quad (3.3)$$

where $W_C^Q, W_C^K, W_C^V \in \mathbb{R}^{d \times d}$ are learnable parameters. Provided the n th layer sequence embedding $H^n \in \mathbb{R}^{L \times d}$ and the descriptive text representation $X \in \mathbb{R}^{L \times d}$, integrated attention score between the descriptive text representation X and the sequence representation H^n is computed as follows:

$$\text{IAS}(H^n, X) = \text{join}(h_1; h_2; \dots; h_h) \cdot W^O$$

$$h_i = \text{IntgAttn}(H^n W_i^Q, H^n W_i^K, H^n W_i^V, X W_i^{Q_C}, X W_i^{K_C}) \quad (3.4)$$

where learnable parameters are $W_i^Q, W_i^K, W_i^V, W_i^{Q_C}, W_i^{K_C} \in \mathbb{R}^{d \times d/f}$ and $W^O \in \mathbb{R}^{d \times d}$ and f is the number of attention heads. We scale integrated attention score A with $\sqrt{4d/f}$ and take product of softmaxed attention score and the value as follows:

$$\text{IntgAttn}(Q_C, K_C, V, K, Q) = \text{softmax} \left(\frac{A}{\sqrt{4d/h}} \right) V \quad (3.5)$$

where Q_C, K_C depict query, key for the descriptive text and Q, K, V are query, key and value for the input sequence. We compute A by evaluating the cross relationships between sequence of items and their corresponding descriptive text as:

$$A = QK^\top + QK_C^\top + Q_C K^\top + Q_C K_C^\top \quad (3.6)$$

3.3.2 Feed Forward Network

Position wise feed forward network (P_{FFN}) is incorporated to handle non linear projections right after CSA module. It is achieved through the use of activation functions applied to the outputs of each layer. Initially, each layer performs a linear transformation

on the input data using a weight matrix and bias vector. To introduce non-linearity, activation function GeLU is applied to the transformed data. This function allows the network to learn complex patterns and representations by adding non-linearities. The combination of linear transformations and non-linear activation functions is applied layer by layer. The output of each layer serves as the input for the next. This process enables the network to build complex, hierarchical representations of the input data, progressively capturing higher-level features. By incorporating non-linear activation function, GeLU feedforward networks approximates any continuous function, allowing them to model intricate relationships in the data and solve a wide variety of complex tasks across different domains. Mathematical notation of (P_{FFN}) is as under:

$$P_{FFN}(H^n) = [S_{FFN}(h_1^n)^\top; S_{FFN}(h_2^n)^\top; \dots; S_{FFN}(h_T^n)^\top]$$

$$S_{FFN}(h_i^n) = \text{GeLU}(h_i^n W_1 + e_1) W_2 + e_2 \quad (3.7)$$

where trainable hyper-parameters communicated at all layers are W_1 , W_2 , e_1 and e_2 . Transformer block is formed by fusing feed forward network and context aware self attention (CSA) module. Context aware bidirectional transformer encoder is constructed by stacking multiple transformer blocks. Complexity of the model is reduced using residual connection [36] at each sub layer. In order to avoid overfitting, we apply dropout [37] followed by layer Normalization [38], LNorm. Layer normalization and dropout are two effective techniques used in neural networks to avoid overfitting and improve generalization. Layer normalization works by normalizing the inputs of each layer across the features, ensuring that the mean and variance of the inputs are consistent. This helps stabilize the learning process and makes the training more robust to varying batch sizes, leading to improved efficiency. However, dropout is a regularization approach where randomly selected neurons are neglected, or "dropped out," in the course of each training iteration. This prevents the network from becoming too reliant on specific neurons and forces it to learn more robust features. By reducing inter dependencies among neurons, dropout mitigates overfitting, as the network is less likely

to memorize the training data. Together, layer normalization and dropout enhance the network’s ability to generalize from the training data to unseen data, resulting in a more reliable and effective model. The context driven bidirectional Transformer encoder Trm_c is constructed as under:

$$\begin{aligned}
 H^n &= Trm_c(H^{n-1}, X), \quad \forall n \in [1, \dots, N] \\
 Trm_c(H^n, X) &= \text{LayerNorm}(A^n + \text{Dropout}(PFN(A^n))) \\
 A^n &= \text{LayerNorm}(H^n + \text{Dropout}(CSA(H^n, X)))
 \end{aligned} \tag{3.8}$$

In a nutshell, after obtaining user sequence embedding H^0 along with its corresponding descriptive text representation X , H^0 is passed through N layers of Transformer blocks. The end result of sequence encoder that is in the form of hidden representation H^N , is captured from the last layer N .

3.4 Output Layer

To train bidirectional transformers, our primary training objective is masked item prediction task **MIP**(occlusion task and the cloze task). m multiple masked sequences $s_u^1, s_u^2, \dots, s_u^m$ are constructed during each iteration by using various random seeds. In each masked sequence s_u^y ($1 \leq y \leq m$), the token of masked item $[mask]$ is randomly changed with ratio ρ of all items and corresponding descriptive text with the mask description des_0 . P_u^y shows the masked items position indices. The task of the model is to take into account the contexts of remaining unmasked items for generating the masked items. We define the loss function for mask item prediction task as under:

$$\mathcal{L}_{MIP} = - \sum_{y=1}^m \sum_{t \in P_u^y} \left[\log \text{sigmoid}(p(v_t | s_u^y)) + \sum_{i_t^- \notin s_u} \log(1 - \text{sigmoid}(p(i_t^- | s_u^y))) \right] \tag{3.9}$$

Randomly sampled one negative item i_t^- is paired with target item i_t . It is pertinent to mention that while calculating the loss function, only masked items are considered.

The item prediction layer generates the probability $p(i)$ and transforms h_t which is the final out put in the form of hidden representation at position t .

$$p(i) = W^P h_t + b^P \quad (3.10)$$

where weight matrix is denoted by $W^P \in \mathbb{R}^{|I| \times d}$ and prediction layer bias term is denoted by $b^P \in \mathbb{R}^{|I|}$.

3.5 Context driven Contrastive Learning

Contrastive learning (CL) is self-supervised learning based methodology used to acquire representations by contrasting examples having similar and dissimilar pairs. Normally, if we are having a batch of sequences $\{s_u\}_{u=1}^D$ with batch size D , a pair of hidden representations H_u^a, H_u^g originating from s_u are treated as positive samples whereas the remaining $2(D - 1)$ stemming from s_u are treated as negative samples [32]. Based on InfoCE, single pair CL loss [39] is illustrated as under:

$$\mathcal{L}(H_u^a, H_u^g) = -\log \frac{e^{\langle H_u^a, H_u^g \rangle / \tau}}{e^{\langle H_u^a, H_u^g \rangle / \tau} + \sum_{k=1, k \neq u}^N \sum_{c \in \{a, g\}} e^{\langle H_u^a, H_k^c \rangle / \tau}} \quad (3.11)$$

where temperature hyper-parameter is denoted as τ . To compute the similarity score among two hidden representations, cosine similarity function $\langle \phi_1, \phi_2 \rangle = \phi_1^T \cdot \phi_2 / \|\phi_1\| \cdot \|\phi_2\|$ is adopted.

This research devises a Context driven Contrastive Learning (CCL) objective to incorporate the descriptive text information in contrastive learning for SR tasks. Specifically, three types of augmentation namely context level random occlusion mask, data level cloze task mask and model level dropout mask are used to produce a rich set of similar pairs. More precisely, these positive samples are generated by integrating $H_u^1, H_u^2, \dots, H_u^m$ (final output of the m hidden representations) and corresponding masked sequences $s_u^1, s_u^2, \dots, s_u^m$. The CCL loss function is computed as under:

$$\mathcal{L}_{CCL} = \sum_{a=1, x \neq g}^m \sum_{g=1}^m l(H_u^a, H_u^g) \quad (3.12)$$

where $l(H_u^a, H_u^g)$ is a CL pair.

3.6 Training

We train the proposed model by integrating together mask item prediction task **MIP** which is our primary training objective and auxiliary task which is context driven contrastive learning objective **CCL**. Hence, we formulate the accumulative loss function for our model as under:

$$\mathcal{L} = \mathcal{L}_{MIP} + \mu\mathcal{L}_{CCL} \quad (3.13)$$

where μ depicts the weighting hyperparameter that regulates the influence of CCL.

Chapter 4

Experimental Results and Analysis

4.1 Experiments

The present part of the research elaborates the datasets used in the proposed model and their preparation followed by experiment setup, evaluation metrics and performance comparison.

4.1.1 Datasets Pre-Processing

Three benchmark datasets including movielens-1m¹, Amazon-Beauty² and Amazon-Toys³ have been utilized to train and demonstrate the proposed model's performance. Datasets are described in detail as under:

- **MovieLens:** A well-known dataset most commonly used for evaluating the performance of SRS. MovieLens ratings dataset contains the user id, item id (IDs of the movies from "movies" table), ratings and timestamp for movie ratings from each user. The auxiliary information (movie plot summary) for MovieLens is extracted through IMDbPY⁴ using the **ImdbId** unique identifier, thus, making it information rich dataset.

¹<https://grouplens.org/datasets/movielens-1m/>

²<http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/metaBeauty.json.gz>

³<http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/metaToysandGames.json.gz>

⁴<https://imdbpy.github.io/>

- **Amazon-Beauty and Toys:** It is a set of dataset comprising of reviews of a number of products extracted from "Amazon.com". The data is disintegrated into different datasets based upon product categories on Amazon. In our experiments, "Beauty" and "Toys" categories are chosen having a "rating" and a "meta" file. To incorporate the auxiliary information in the "rating" dataset, "description" of each product is extracted from the "meta" dataset.

We eliminate redundant data of interactions and arrange every interactions in chronological manner in order to create user sequences. Following the methodologies of [10], [13], [17], we exclude users having lesser than five interactions and items associated with fewer than five users. We conduct testing with last item, validation with second last and rest for training by employing the leave-one-out evaluation approach. Table 4.1 summarizes the processed dataset statistics.

Table 4.1. Datasets statistics after Pre-Processing

Datasets	Beauty	Toys	ML-1M
# of Users	22,363	19,412	6,040
# of Items	12,101	11,924	3953
# of Interactions	198,502	167,597	1,000,209
Avg. Length	8.9	8.6	163.5
Sparsity	99.92%	99.93%	95.81%

4.1.2 Evaluation Metrics

For evaluating all models, we compute "Normalized Discounted Cumulative Gain" (NDCG) and "Hit Ratio" (HR). The degree of performance is measured by the higher values of these metrics. We utilize top-k ranking to evaluate the performance i.e Hit Ratio@ K (HR@ K) and Normalized Discounted Cumulative Gain@ K (NDCG@ K) where $K \in \{5, 10, 20\}$. Hit Ratio is used for measuring the ranking accuracy by com-

paring the test item set (T) with the ranked list. Mathematically it is expressed as:

$$\text{HR@K} = \frac{\text{Number of Hits@K}}{|T|} \quad (4.1)$$

HR@K computes the number of hits in a list with K size. A hit occurs if the item tested is available in ranked list. Whereas the relative position of that item is assessed using NDCG in the ranked list. It assigns higher scores if the item is present at top position in list. Mathematically it is evaluated by following formula:

$$\text{NDCG@K} = N_K \sum_{i=1}^K \frac{2^{z_i} - 1}{\log_2(i + 1)} \quad (4.2)$$

where N_K is the normalizer and z_i being the item’s graded relevance at position i . We compute both the metrics of every test user items and then take their mean.

4.1.3 Baselines

For performance comparisons, we consider the following methods from three groups.

4.1.3.1 Sequential Models

Sequential models refer to a class of neural networks designed to process sequential data, where the order of inputs is crucial for understanding the data’s meaning or context. These models are particularly adept at handling tasks where the temporal or sequential dependencies between elements matter. These models produce item and user representations using sequence encoders.

- **SASRec**[9]: SASRec (Self-Attentive Sequential Recommendation) leverages self-attention mechanisms to model sequential patterns in user behavior, offering robust recommendations based on deep understanding of item sequences. This is a unidirectional (left-to-right) self attentive framework for next item prediction.
- **BERT4Rec** [10]. BERT4Rec enhances recommendation systems by leveraging BERT’s contextual embeddings to capture nuanced user preferences and item

characteristics from textual data, improving the relevance and personalization of recommendations. This top of the line model uses bidirectional self attentive blocks and cloze task masking for the recommendation task.

- **KeBERT4Rec** [11]. KeBert4Rec integrates knowledge graphs with BERT to enhance recommendation systems, leveraging contextual information for more personalized and effective recommendations. This model extends BERT4Rec by integrating keywords as additional input layer.

4.1.3.2 Context driven sequential methods

These models incorporate both sequence and contextual information to enhance recommendation tasks. Some examples are as under:

- **S³Rec** [24]. It utilizes self-supervised learning techniques to enhance sequential recommendation systems. This approach focuses on maximizing mutual information between different parts of the user-item interaction sequence, thereby capturing meaningful patterns and dependencies in the data without explicit user feedback or ratings.
- **FCLRec** [31]. It adopts BERT4Rec to design intrinsic association between item sequences and associated features.

4.1.3.3 Sequential models with contrastive learning

Sequential models combined with contrastive learning represent a powerful approach to learning meaningful representations from sequential data. This combination leverages the strengths of both sequential modeling, which captures temporal dependencies, and contrastive learning, which enhances the discriminative power of representations by comparing positive and negative pairs. These models improve sequential recommendation with contrastive learning objective. Some examples are as under:

- **CL4SRec** [13]. It is an innovative approach that utilizes transformer architecture to integrate contrastive learning techniques into sequential recommendation system. This model focuses on improving SR by leveraging contrastive learning.
- **CoSeRec** [14]. It is a sophisticated framework that combines contrastive learning with self-supervised techniques to enhance the performance of recommendation. The primary aim of CoSeRec is to improve the learning of user-item interaction patterns by leveraging the strengths of both self-supervised learning and contrastive objectives.
- **DuoRec** [15]. It adopts model augmentation as a approach to improve item embedding distribution and overcome representation degeneration problem in contrastive learning.
- **CBiT** [17]. It is a BERT-based CL framework that utilizes cloze task and dropout mask. It fuses strengths of both contrastive learning and bidirectional transformer architectures to improve the performance and robustness of recommendation systems.

4.1.4 Implementation Details

All baseline models were implemented in Pytorch [40] by the codes provided by authors. Instructions from the authors of original papers were followed for setting hyper parameters. We implement our model in Pytorch framework and train on machine with NVIDIA Tesla T4 GPU, 1.59 GHz and 16 GB RAM, utilizing transformer encoder with Transformer blocks N and attention layers set as 2, hidden dimensions set as 128 and batch size set as 256. We use optimizer Adm [41] with initial learning rate lr 0.001 and 0.01 as weight decay. The masking probability ρ is set as 0.15 for cloze mask task. Other hyper-parameters are tuned as 3 to 18 the number of positive samples m , 0.1 to 5 τ , 0.1 to 0.8 dropout ratio, The hidden dimensionality d is tested from 64,128,256, 20 to 120 the rolling glass step size L , weighting hyper-parameter μ from 0.01 to 0.6.

The best NDCG@10 score is displayed by selecting checkpoint for testing with number of epochs set as 256 on the validation set.

4.2 Comprehensive Performance Analysis

Table 4.2 depicts the optimized outcomes of each baseline model on benchmark datasets. The performance efficiency of our proposed model in comparison to the best baseline method is displayed in last column.

We can establish from results that use of transformer based self attention models are more accurate than using traditional mechanisms. It is evident that SASRec performance falls behind as compare to BERT4Rec which depicts that bidirectional model like BERT4Rec is more powerful as compared to unidirectional model like SASRec. Furthermore, BERT4Rec is a SR model that relies only on the item identifiers for the purpose of generating representation/ embedding. This model ignores the auxiliary information that is already provided with the datasets. However, KeBERT4Rec, a variant of BERT4Rec, has modified the representation by adding keywords describing the items e.g. Genre of movie. The addition of this keywords embedding in the model makes KeBERT4Rec perform better than BERT4Rec. It is observed that context aware methods further produce better results than sequential methods. Thus, suggesting that incorporating some kind of side information along with item can improve the recommender’s performance. We also notice that model augmentation(DuoRec) and hybrid augmentation (CBiT) based CL methods perform significantly better than models based on data augmentation (CL4Rec, CoSeRec). While contrastive learning emphasizes the relative positioning of data points, contextual methods focus on the sequential nature and dependencies within the data. Integration of contextual aspect with contrastive learning objective offers depth in understanding sequential patterns. We infer that some original context of sequences might get corrupted by data augmentation and model augmentation cause minimal deviation avoiding any significant damage to original context.

Table 4.2. Comprehensive performance analysis of proposed model with referenced models for next item recommendations. The highest scores are shown in bold, while the second place scores are underlined.

Dataset	Metric	Sequential Methods		Contextual SR Models		Sequential Models with CL						Improv.
		SASRec	BERT4Rec	KeBERT4Rec	S ³ Rec	CL4SRec	CoSeRec	DuoRec	CBiT	FCLRec	CCLRec	
Beauty	HR@5	0.0371	0.0370	0.0436	0.0382	0.0396	0.0504	0.0559	0.0637	<u>0.0679</u>	0.0702	3.39%
	HR@10	0.0592	0.0598	0.0652	0.0634	0.0630	0.0726	0.0867	0.0905	<u>0.0954</u>	0.0983	3.04%
	HR@20	0.0893	0.0935	0.0958	0.0981	0.0965	0.1035	0.1102	0.1223	<u>0.1310</u>	0.1315	0.38%
	NDCG@5	0.0233	0.0233	0.0323	0.0244	0.0232	0.0339	0.0331	0.0451	<u>0.0480</u>	0.0501	4.37%
	NDCG@10	0.0284	0.0306	0.0358	0.0335	0.0307	0.0410	0.0430	0.0537	<u>0.0569</u>	0.0571	0.35%
	NDCG@20	0.0361	0.0391	0.0411	0.0429	0.0392	0.0488	0.0524	0.0617	<u>0.0658</u>	0.0674	2.43%
Toys	HR@5	0.0429	0.0371	0.0385	0.0440	0.0503	0.0533	0.0539	0.0640	<u>0.0641</u>	0.0664	3.59%
	HR@10	0.0652	0.0524	0.0571	0.0705	0.0736	0.0755	0.0744	0.0865	<u>0.0909</u>	0.0948	4.29%
	HR@20	0.0957	0.0760	0.0827	0.1008	0.0990	0.1037	0.1008	0.1167	<u>0.1239</u>	0.1285	3.71%
	NDCG@5	0.0248	0.0259	0.0245	0.0286	0.0264	0.0370	0.0340	0.0462	<u>0.0464</u>	0.0478	3.02%
	NDCG@10	0.0320	0.0309	0.0311	0.0369	0.0339	0.0442	0.0406	0.0535	<u>0.0551</u>	0.0566	2.72%
	NDCG@20	0.0397	0.0368	0.0432	0.0458	0.0404	0.0513	0.0472	0.0610	<u>0.0634</u>	0.0642	1.26%
ML-1M	HR@5	0.1078	0.1308	0.1801	0.1128	0.1142	0.1128	0.1930	0.2095	<u>0.2196</u>	0.2243	2.14%
	HR@10	0.1810	0.2219	0.2597	0.1969	0.1815	0.1861	0.2865	0.3013	<u>0.3090</u>	0.3146	1.81%
	HR@20	0.2745	0.3354	0.3185	0.3067	0.2818	0.2950	0.3901	0.3998	<u>0.4214</u>	0.4114	2.90%
	NDCG@5	0.0681	0.0804	0.8194	0.0668	0.0705	0.0692	0.1327	0.1436	<u>0.1517</u>	0.1541	1.58%
	NDCG@10	0.0918	0.1097	0.1582	0.0950	0.0920	0.0915	0.1586	0.1694	<u>0.1806</u>	0.1865	3.27%
	NDCG@20	0.1156	0.1384	0.2213	0.1189	0.1170	0.1247	0.1843	0.1957	<u>0.2090</u>	0.2152	2.88%

In accordance with the outcomes, on the benchmark datasets, our proposed model **CCLRec** clearly outperforms all baseline methods. This clearly demonstrates that more complex association between item sequence and its corresponding description can be extracted by the context driven bidirectional transformer and context driven contrastive learning objective is able to optimize model via three different augmentation types, introducing more accuracy and effectiveness. The proposed model gains a significant improvement by 5.69% to 6.34% in *NDCG@10*.

4.3 Hyperparameter Study

In the present part, the impact of significant hyperparameters such as number of positive samples m , hidden dimensionality d , dropout ratio and rolling glass step, in **CCLRec** is discussed by altering one hyperparameter at a time and keeping others constant.

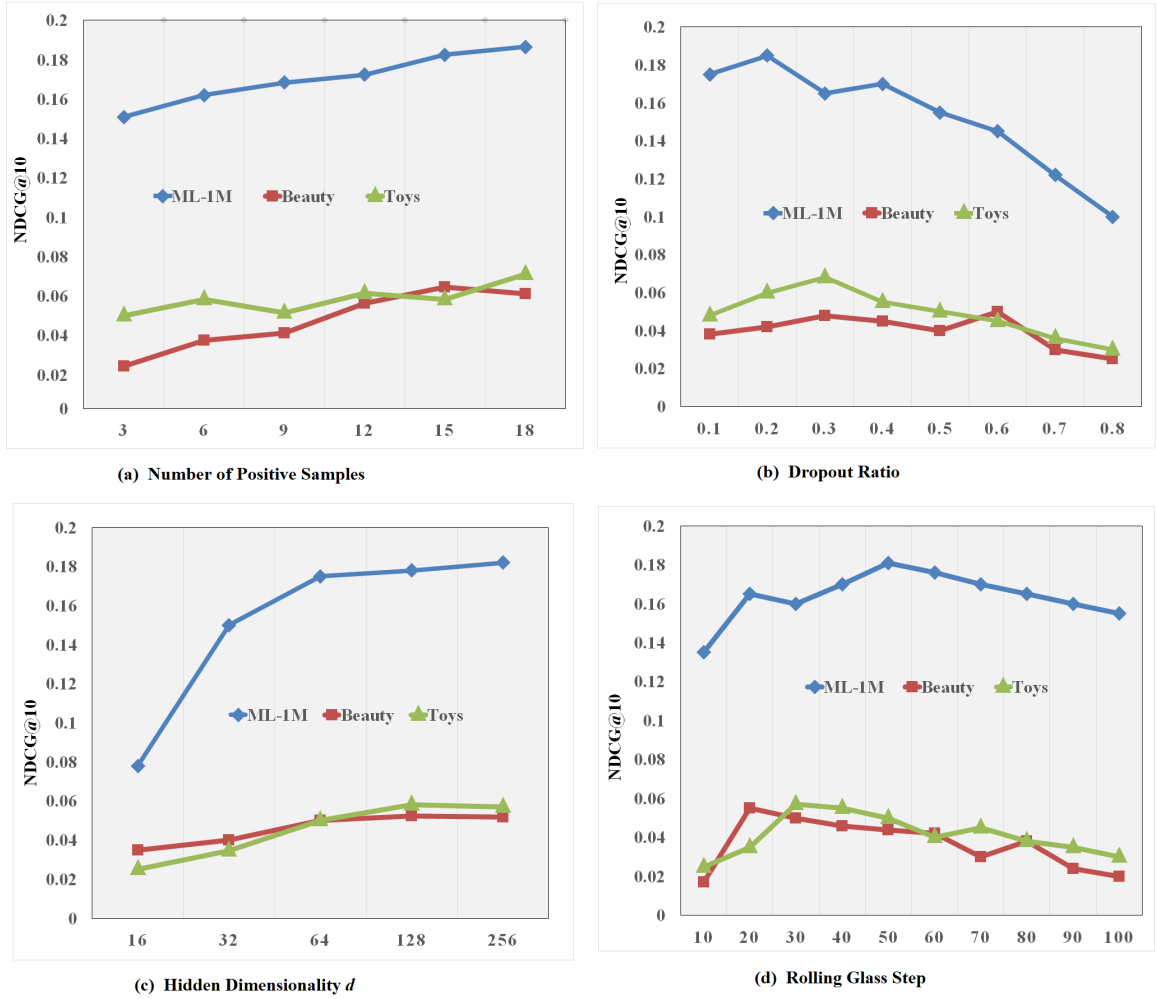


Figure 4.1: $NDCG@10$ performance analysis of three benchmark datasets with respect to other hyperparameters.

4.3.1 Number of Positive Samples

Figure 4.1a depicts the influence of number of positive samples m on the efficiency of CCLRec. More number of high-quality positive samples from the same sequence are contrasted with $m(N-1)$ number of negative samples from other sequences in the same batch. This approach helps the model learn more generalizable, meaningful and robust representations. These representations capture the underlying hidden structure and association within the items more effectively. However, balancing an adequate number

of positive samples is pertinent for training a robust model because there is a point of diminishing returns where adding more samples does not lead to further improvements in performance.

4.3.2 Dropout Ratio

The dropout ratio is a crucial hyperparameter in neural networks that impacts their performance by preventing overfitting and promoting generalization. Dropout involves randomly "dropping out" a fraction of neurons during training, which means setting their output to zero with a certain probability (the dropout ratio). Dropout ratio is a critical hyperparameter that needs careful tuning. We noticed from Figure 4.1b that CCLRec deteriorates in performance when the dropout ratio is excessively small or excessively large. The right balance should be in between 0.6 for Beauty, 0.3 for Toys and 0.2 for ML-1M to significantly enhance the model's ability to avoid overfitting or underfitting.

4.3.3 Hidden Dimensionality d

Hidden dimensionality d is a critical hyperparameter that influences the model's ability to learn from data. Balancing hidden dimensionality requires careful consideration of the trade-offs between underfitting and overfitting, computational resources, and the use of regularization techniques to achieve optimal performance. The hidden dimensionality d has a significant influence on the efficiency of recommendation system. Figure 4.1c exhibits the values of NDCG@10 on different benchmark datasets by varying hidden dimensionality d ranges between 16,32,64,128,256. It is obvious that with the increase of dimensionality, the graph converges. However, improved model performance is not always achieved with bigger value of hidden dimensionality, particularly on sparse datasets such as Beauty.

4.3.4 Rolling Glass Step

Rolling glass step L is a critical factor in sequential data modeling that influences the ability of a model to capture more fine-grained context, balance between overfitting and underfitting, and adapt to data variability. A smaller step size increases training data and accuracy but at the cost of higher computation. A larger step size reduces computational load but may miss important patterns. Balancing these factors and tailoring the step size to the specific application and data characteristics is essential for optimal performance. Glass size should be selected by keeping in consideration particular task requirements and validated through empirical experimentation. We observed from Figure 4.1d that appropriate rolling glass step should neither be too large or too small in order to avoid underfitting or overfitting. We keep L 50, 30 and 20 for ML-1M, Toys and Beauty datasets respectively.

4.4 Computational Complexity

CCLRec involves a combination of the complexities inherent in both contrastive learning and transformer architecture. For Bidirectional transformers, larger values of rolling glass step increase quadratic complexity due to the attention mechanism $O(L^2d)$. Therefore time complexity of CCLRec is $O((N - L + 1)L^2d)$ where $N - L + 1$ is the rolling glass step size.

4.5 Ablation Study

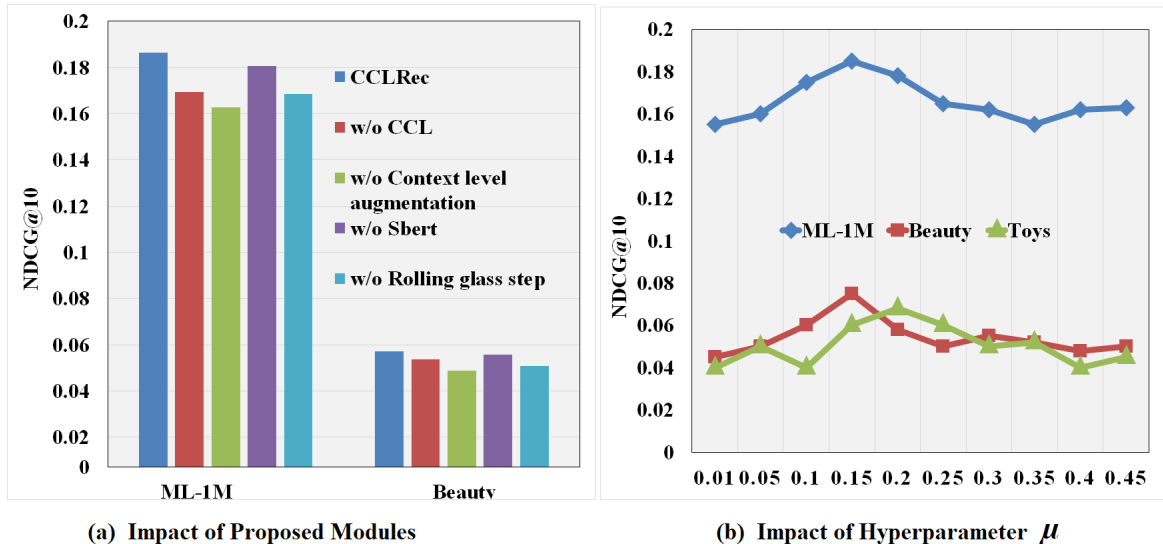


Figure 4.2: Ablation study ($NDCG@10$) on the impact of proposed modules and auxiliary contextual task

4.5.1 Impact of Proposed Modules

Table 4.3 shows some ablation experiments that were performed on two datasets. To visualize the impact of proposed modules (1) CCLRec presents results with all modules. (2) computes results without contrastive learning at all. (3) shows results with only data and model level augmentation. (4) presents results by testing proposed model using one hot encoding techniques to generate textual embedding. (5) presents impact on results by truncating user sequence s_u to fixed length. We can notice from Table 4.3 and Figure 4.2a that rolling glass step successfully improved results by capturing more fine grained representations. This also emphasize the use of meaningful and context embedding generating technique i.e SBert for model training to produce relevant results. Moreover contrastive learning with context level augmentation is better than absolutely no CL.

Table 4.3. Ablation Study on our proposed modules. The outcomes depict that proposed techniques enhanced the overall performance.

Modules	Ml-1m		Beauty	
	HR@10	NDCG@10	HR@10	NDCG@10
(1) CCLRec	0.3146	0.1865	0.0983	0.0571
(2) w/o CCL	0.3013	0.1694	0.0905	0.0537
(3) w/o Context level augmentation	0.2954	0.1628	0.0901	0.0488
(4) w/o SBert	0.3090	0.1806	0.0954	0.0558
(5) w/o Rolling glass step	0.2874	0.1685	0.0948	0.0507

4.5.2 Impact of Auxiliary Contextual Task μ

Some ablation experiments were performed to assimilate the influence of the auxiliary contextual information. We show performance of NDCG@10 of our framework under different values of weighting hyper-parameter μ , that optimizes context driven contrastive learning objective. Figure 4.2b depicts that performance of the model declines on removal of context driven contrastive learning objective i.e ($\mu = 0$), thus proving that auxiliary contextual task has significant impact on efficiency of the model.

Chapter 5

Summary of Research Work

Sequential recommender systems are essential for forecasting users' future preferences by examining their past interactions in a dynamic manner. These systems are essential in various online platforms to relieve information overload and deliver tailored and pertinent recommendations to users. To enhance the accuracy of sequential recommendations, advanced methodologies like contrastive learning have been introduced. Contrastive learning leverages data augmentation techniques and transformer-based models to learn rich representations by pulling positive views and pushing away negative views within batch examples. By addressing challenges such as extreme sparse interaction matrices and interaction noises in datasets, contrastive learning frameworks like CL4SRec, CoSeRec, and DuoRec have shown notable improvements in learning representations for sequential recommendation tasks.

This study introduces a novel framework called **Bidirectional Transformers driven Contextual Sequential Recommendation with Contrastive Learning (CCLRec)** to further enhance recommendation accuracy. CCLRec extends bidirectional transformers and Sentence-Bert to incorporate auxiliary information from textual features of items, enabling a more comprehensive understanding of user preferences. The framework introduces a context aware self-attention module to capture meaningful relationships between sequences and descriptive text, improving the modeling of complex user behaviors. By integrating a Context-driven Contrastive Learning (CLL) objective that

generates positive samples through various augmentations, CCLRec demonstrates remarkable improvements in recommendation accuracy compared to traditional methods. Furthermore, CCLRec presents a comprehensive approach to sequential recommendation by combining advanced methodologies like bidirectional transformers, contextual auxiliary information, and contrastive learning. The model’s innovative design, incorporating the context-aware self-attention module and the CLL objective, showcases significant improvements in recommendation accuracy. By identifying meaningful relationships between items and descriptive textual information, CCLRec provides a robust solution to the dynamic nature of user preferences, enhancing the overall recommendation process. Moreover, the proposed model incorporates rolling glass technique to handle lengthy user sequences which further brings innovation to the model by capturing deep semantics at fine grained level.

A state-of-art comparison with a similar work from literature shows the better performance of the proposed system. It has achieved higher accuracy than those reported in previous studies. The results highlight the effectiveness of CCLRec in improving recommendation accuracy and user satisfaction, validating its capability to address data noise, sparsity issues, and enhance user satisfaction in recommendation systems. This positions it as a promising framework for future developments in sequential recommendation systems.

In conclusion, CCLRec demonstrates an innovative development in the field of sequential recommendation systems by combining advanced methodologies such as bidirectional transformers, contextual auxiliary information, and contrastive learning. The framework’s innovative design, comprehensive experiments, and superior performance compared to existing models underscore its potential to revolutionize the way recommendations are made in various online platforms. CCLRec sets a new standard for accuracy, adaptability, and user satisfaction in sequential recommendation systems, paving the way for future advancements in personalized recommendation technologies.

Chapter 6

Conclusion

Self Attention and Transformer based recommendation system have proven to be more precise and accurate as compared to traditional RS. In this paper, a transformer driven contrastive learning based sequential RS have been proposed that enhances recommendation accuracy by incorporating contextual auxiliary information of items in a sequence. In order to generate the auxiliary information embeddings, a contextual based pre-trained model *sentence – transformer* is adopted. Additionally, the framework introduces a context-driven self-attention module to capture intricate relationships between user sequences and descriptive text, further enhancing the modeling of user preferences in recommendation systems.

One of the key contributions of CCLRec is the introduction of a Context-driven Contrastive Learning (CLL) objective, which generates positive samples through various data augmentation techniques. By leveraging cloze task masks, dropout masks, and random occlusion masks, CCLRec produces high-quality positive samples that improve the model’s adaptability and performance. The framework’s ability to handle lengthy user sequences and descriptive features through the rolling glass step technique showcases its adaptability and efficiency in processing large amounts of data. Comprehensive experiments on three public benchmark datasets shows remarkable improvements as compared to top of the line models.

Future research may extend our framework to Large Language Models embedding technique **LLM2vec** that holds great potential for improving the accuracy and effectiveness of recommendation systems by providing more nuanced and contextually rich representations of textual information, ultimately resulting in more tailored and pertinent recommendations for users.

Bibliography

- [1] G. Shani, R. I. Brafman, and D. Heckerman, “An mdp-based recommender system,” 2015.
- [2] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized markov chains for next-basket recommendation,” in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 811–820. [Online]. Available: <https://doi.org/10.1145/1772690.1772773>
- [3] R. He and J. McAuley, “Fusing similarity models with markov chains for sparse sequential recommendation,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 191–200.
- [4] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, “A simple convolutional generative network for next item recommendation,” 2018.
- [5] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,” 2016.
- [6] B. Hidasi and A. Karatzoglou, “Recurrent neural networks with top-k gains for session-based recommendations,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, Oct. 2018. [Online]. Available: <http://dx.doi.org/10.1145/3269206.3271761>

- [7] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.
- [9] W.-C. Kang and J. McAuley, “Self-attentive sequential recommendation,” 2018.
- [10] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer,” 2019.
- [11] E. Fischer, D. Zoller, A. Dallmann, and A. Hotho, “Integrating keywords into bert4rec for sequential recommendation,” in *KI 2020: Advances in Artificial Intelligence: 43rd German Conference on AI, Bamberg, Germany, September 21–25, 2020, Proceedings*, 2020.
- [12] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou, “Feature-level deeper self-attention network for sequential recommendation,” in *International Joint Conference on Artificial Intelligence*, 2019.
- [13] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, B. Ding, and B. Cui, “Contrastive learning for sequential recommendation,” 2021.
- [14] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. McAuley, and C. Xiong, “Contrastive self-supervised sequential recommendation with robust augmentation,” 2021.
- [15] R. Qiu, Z. Huang, H. Yin, and Z. Wang, “Contrastive learning for representation degeneration problem in sequential recommendation,” in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '22. ACM, Feb. 2022. [Online]. Available: <http://dx.doi.org/10.1145/3488560.3498433>

- [16] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, “Intent contrastive learning for sequential recommendation,” in *Proceedings of the ACM Web Conference 2022*, ser. WWW ’22. ACM, Apr. 2022. [Online]. Available: <http://dx.doi.org/10.1145/3485447.3512090>
- [17] H. Du, H. Shi, P. Zhao, D. Wang, V. S. Sheng, Y. Liu, G. Liu, and L. Zhao, “Contrastive learning with bidirectional transformers for sequential recommendation,” 2022.
- [18] A. Petrov and C. Macdonald, “A systematic review and replicability study of bert4rec for sequential recommendation,” 2022.
- [19] R. He and J. McAuley, “Fusing similarity models with markov chains for sparse sequential recommendation,” 2016.
- [20] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” 2014.
- [21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
- [22] J. Jiang, J. Zhang, and K. Zhang, “Cascaded semantic and positional self-attention network for document classification,” 2020.
- [23] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” 2016.
- [24] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, “S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM ’20. New

- York, NY, USA: Association for Computing Machinery, 2020, p. 1893–1902. [Online]. Available: <https://doi.org/10.1145/3340531.3411954>
- [25] C. Xu, J. Feng, P. Zhao, F. Zhuang, D. Wang, Y. Liu, and V. S. Sheng, “Long- and short-term self-attention network for sequential recommendation,” *Neurocomputing*, vol. 423, pp. 580–589, 2021.
- [26] X. Chen, Z. Wang, H. Xu, J. Zhang, Y. Zhang, W. X. Zhao, and J.-R. Wen, “Data augmented sequential recommendation based on counterfactual thinking,” *IEEE Trans. on Knowl. and Data Eng.*, vol. 35, no. 9, p. 9181–9194, sep 2023. [Online]. Available: <https://doi.org/10.1109/TKDE.2022.3222070>
- [27] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, “Graph contextualized self-attention network for session-based recommendation,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, ser. IJCAI’19. AAAI Press, 2019, p. 3940–3946.
- [28] X. Zheng, J. Su, W. Liu, and C. Chen, “Ddghm: Dual dynamic graph with hybrid metric training for cross-domain sequential recommendation,” 2022.
- [29] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang, “Context-aware sequential recommendation,” 2016.
- [30] M. Zhang, C. Guo, J. Jin, M. Pan, and J. Fang, “Sequential recommendation with context-aware collaborative graph attention networks,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.
- [31] H. Du, H. Yuan, P. Zhao, D. Wang, V. S. Sheng, Y. Liu, G. Liu, and L. Zhao, “Feature-aware contrastive learning with bidirectional transformers for sequential recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023.

- [32] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” 2021.
- [33] C. Wang, W. Ma, C. Chen, M. Zhang, Y. Liu, and S. Ma, “Sequential recommendation with multiple contrast signals,” *ACM Trans. Inf. Syst.*, vol. 41, no. 1, jan 2023. [Online]. Available: <https://doi.org/10.1145/3522673>
- [34] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020.
- [35] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” 2019.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, jan 2014.
- [38] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016.
- [39] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2019.
- [40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” 2019.
- [41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.