ISBN 978-82-326-4848-1 (printed ver.)
ISBN 978-82-326-4849-8 (electronic ver.)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Peng Liu

Doctoral thesis

Doctoral theses at NTNU, 2020:250

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

NTNU

Doctoral theses at NTNU, 2020:250

Peng Liu

Exploiting Latent Context for Effective Social Recommendation

NTNU
Norwegian University of
Science and Technology

Peng Liu

Exploiting Latent Context for Effective Social Recommendation

Thesis for the Degree of Philosophiae Doctor

Trondheim, September 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Computer Science

ISBN 978-82-326-4848-1 (printed ver.)
ISBN 978-82-326-4849-8 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2020:250

Printed by NTNU Grafisk senter

Abstract

With the rapid proliferation of online social networks, the information overload problem becomes increasingly severe, and recommender systems play a critical role in helping online users discover useful information matching their individual preferences. Significant recommendation researches have focused on the explicit context such as time, location and weather etc. Despite effectiveness, obtaining explicit contexts is usually a resource-demanding task, and it is not always available in real-world recommender systems. In contrast, the latent contexts, which could be learned automatically from raw data by applying machine learning techniques, are much easier to obtain but lack of comprehensive study. Moreover, the cold start issue and the special properties of social networks, such as multilingualism, rich temporal dynamics, heterogeneous and complex structures with millions of nodes, render the most commonly used recommendation approaches (e.g. Collaborative Filtering) inefficient. Therefore, in this thesis, we investigate the latent contexts provided by three prevalent sources in the social network for effective social recommendation: (i) user-generated reviews, (ii) social links and (iii) multimedia data.

To begin with, user-generated reviews have been seen as a valuable information source to build a fine-grained user preference model and enhance personalized recommendation. First, we propose a probabilistic generative model (DTSA) to extract topics and topic-specific sentiments from textual reviews and analyze their evolution over time simultaneously. To further explore the contextual information the DTSA neglects, and multilingual resources in social media, we then devise a multilingual review-aware deep recommendation model which can not only extract aligned aspects and their associated sentiments in different languages, but also leverage the extracted information as multilingual contexts for overall rating prediction and item recommendation.

Furthermore, social links indicate different types of social connections associated with users and/or items, which form a heterogeneous user-item (HUI) network. To address the issues of temporal dynamics, cold start and context awareness in the social recommender system, we propose a dynamic graph-based embedding model (DGE) that jointly captures the temporal semantic effects, social relationships and user behaviour sequential patterns in a unified way by embedding the HUI network into a shared low dimensional space. Considering the global pattern of vertices, we then extend our DGE model by incorporating the community information derived from network structure into graph embedding model for social recommendation.

Last but not least, visual information considered as an essential part of multimedia data can also be a significant complementary resource when performing recommendations for some types of items such as movies, clothing, etc. To fully exploit visual contexts, we propose an Attentive Recurrent Neural Network (Ante-RNN) for the dynamic explainable recommendation which could provide multi-model explanations according to the user dynamic preference. We further analyze and study a variety of fusion strategies for mutual association learning across modalities, and find that the attention-based fusion robustly achieves the best results.

By performing extensive experiments on real-world datasets from social networks, our proposed methods outperform the competitive baselines including cold-start scenario both in efficiency and effectiveness.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfilment of the requirements for the degree of philosophiae doctor.

This doctoral work has been performed at the Department of Computer Science (IDI), NTNU, Trondheim, with Professor Jon Atle Gulla as main supervisor and with Professor Kjetil Nørvåg and Associate Professor Xiaomeng Su as co-supervisors.

The thesis has been part of the research project Recommendation Technologies (RecTech), supported by the Research Council of Norway, grant number 245469 under the BIA innovation research program.

Acknowledgements

Pursuing a PhD degree is a tough road for me, and it would not have been possible to do without the support and help that I received from many people. First and foremost, I would like to express my sincere gratitude to my main supervisor, Professor Jon Atle Gulla. Without his timely advice, encouragement and immense knowledge, I would not be able to finish my PhD career. Under his elaborative guidance, I have not only learned the state-of-the-art data mining knowledge, but also acquired a conscientious research attitude through critical thinking. The knowledge and methodology I learnt are invaluable to my research, and even to my life. I would also like to thank my co-supervisor Professor Kjetil Nørvåg and Associate Professor Xiaomeng for their supports and help during these years.

I would like to thank all the members of the SmartMedia group: Dr. Jon Espen Ingvaldsen, Dr. Cristina Marco, Dr. Özlem Özgöbek, Yujie Xing and Nils Barlaug. I appreciate all their efforts on the research work, and I enjoy the experience of the collaborations with them all the time. I would also like to thank my fellow friends at IDI for their friendly support and help: Shang Gao, Meng Zhu, Liyuan Xing, Jie Xu, Hong Guo, Zhe Li, Jin Zhang and Eliezer de Souza da Silva. Thanks to the administrative and technical staff at IDI for providing me kindly assistance. I would also like to thank all my colleagues at IDI for having created a pleasant and multicultural working environment.

I am immensely grateful to my father Jingfan Liu and my mother Zunhong Wang for supporting me in all my pursuits and for raising me with their great love and patience. Thanks also to my brother Kui Liu for continuous supporting. I would also like to thank my parents-in-law An Zhang and Xiaochan Zhao for bringing up my wife and their understanding, encouragement and help when we are in Norway. Finally, I would like to give a special thank to my loving wife Lemei Zhang and our wonderful daughter Xiaobei Liu for the love, support and happiness I have from them.

Contents

List of Figures

List of Tables

Part I

Introduction and Background

Chapter 1

Introduction

This chapter introduces the main topics of the thesis. Section 1.1 explains the main motivation behind this thesis. Some important limitations and challenges of the state-of-the-art are introduced therein. Further, we describe the research context of our PhD work in Section 1.2. The research questions are then formulated and presented in Section 1.3. We discuss the research approach in Section 1.4, followed by our main contributions composed to address the proposed research questions in Section 1.5. Publications included in this thesis are listed in Section 1.6. At the end of this chapter, we present the organization of the rest of the thesis.

1.1 Motivation

Nowadays, the rapid development of Web 2.0 and smart mobile devices have resulted in the dramatic proliferation of online social networks. According to Twitter statistics, the number of users is estimated to have surpassed 300 million, generating more than 500 million tweets per day[1]. Under such circumstances, recommender systems (RSs) are designed to provide an effective way of alleviating the information overload problem by suggesting to online users the information that is potential of interests.

Collaborative filtering (CF), which has been extensively investigated in the research community and widely used in industry, is one of the most popular recommendation techniques due to its accuracy and scalability. It makes predictions about the user's interests based on preferences of his/her like-minded users. However, such approaches inherently suffer from data sparsity and cold start problems.

Recently, there have been numerous studies on the social recommendation which exploit different types of contextual information in social networks to address these issues and

[1] https://www.omnicoreagency.com/twitter-statistics/

improve the quality of recommendations [1-4]. Meanwhile, the pervasive usage of social media like Twitter, Facebook and Goodreads allows users to connect to each other, to participate in online activities, and to generate shared contents or opinions which produce a large amount of social contextual information, such as social relations, locations, weather conditions and item reviews. In general, according to the different acquisition methods, social contextual information can be classified into the following two categories:

- *Explicit context* describes known user situations that can be acquired either according to a predefined set of context conditions specified by domain experts or from the user's explicit inputs inferring the current context such as time, location and weather etc.
- *Latent context*, on the other hand, can be obtained by applying machine learning techniques on available raw data. In contrast with explicit context which is directly from user inputs or knowledgeable domain experts, latent context is comprised of hidden context patterns and represented by learned numerical vectors.

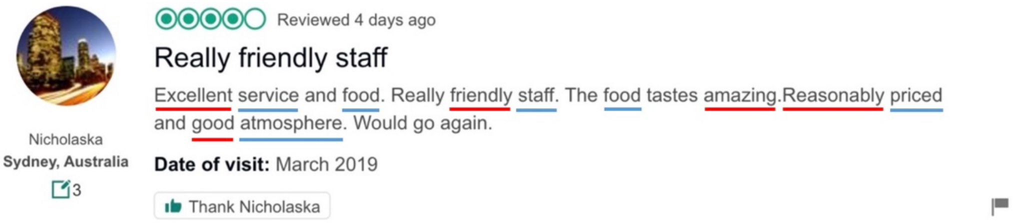
Many studies incorporating explicit contexts have demonstrated their effectiveness in improving the recommendation performance. Besides, explicit context can be better explained by human experts and users than latent context, since it describes known user situations. Despite these advantages that explicit context possesses, the motivation of leveraging latent context stems from the privacy issues, the usage considerations and the availability of data resources. The high degree of readability and unreasonable use of explicit context may raise privacy issues since the recommender systems are fully aware of the exact context of the user, which is not the case for latent context. Compared with the acquisition of explicit context which is usually a resource-demanding task, latent context can be obtained automatically by applying machine learning techniques. Moreover, another benefit of using latent context lies in its ability to reveal complex correlations within the data. Therefore, in this thesis, we focus on the usage and integration of latent contexts in social recommendation tasks.

1.1.1 Latent Context in Social Recommendation

Online social networks provide independent and diverse information sources for the recommendation, which present both opportunities and challenges. This section describes several prevalent sources in social networks and their latent contexts that can be incorporated into the social recommendation.

User-generated reviews

Online social and e-commerce websites allow users to naturally write reviews to describe their opinions and experience towards items and events. These reviews are usually in the form of free text and play the role of carriers that express the reasons why the users like or dislike the items or events they concerned. They can, therefore, be a rich source of context data which can be exploited to build a fine-grained user preference model and enhance personalized recommendations [5, 6]. Figure 1.1 shows a restaurant review of a real-world user from TripAdvisor. Topics/Aspects mentioned in the review include service, food, staff, price, atmosphere. Sentiment words including positive expressions such as excellent, friendly, amazing, good, reasonably, together with aforementioned topics/aspects reflect the user's overall assessment (4-star rating) and multi-faceted preference of the restaurant, which should be taken into account when performing recommendations.



Nicholaska
Sydney, Australia
3

●●●●○ Reviewed 4 days ago

Really friendly staff

Excellent service and food. Really friendly staff. The food tastes amazing.Reasonably priced and good atmosphere. Would go again.

Date of visit: March 2019

Figure 1.1: A restaurant review example from TripAdvisor[2] (note that the blue underline represents topic/aspect words and the red underline represents sentiment words).

It is well recognized that the sentiment polarities of words are usually dependent on their corresponding aspects, especially in user reviews. The polarities of sentiment words vary from aspect to aspect. Although the methodologies on how to extract aspect-based sentiments from user reviews has been well studied, there are still many challenges in front of online social networks: (1) Many topic/aspect-sentiment extraction models are suitable for one specific language but cannot be readily adopted in other languages, since they usually require external resources such as sentiment lexicon and rich corpus which are not publicly available. (2) Considering the dynamic nature of data streams, topics/aspects and their corresponding sentiments are also evolving over time. However, most recent studies assume that the training data are all available prior to model learning, and thus the whole

[2] https://www.tripadvisor.com/

model needs to be retrained when new data arrives. (3) The user interactions among reviews can easily lead to inaccurate topic extraction and sentiment classification.

Social links

With the rapid growth of Internet, huge networks with heterogenous information which contain multi-typed objects and social links are ubiquitous. Social links indicate different types of social connections associated with users and/or items, such as friendships in Facebook, co-purchased item connections in Amazon and trust relations in Epinions. The idea to utilize social links as an information source for their recommendations is based on semantic correlations among items and social correlation theories — homophily and social influence — which indicate that there are correlations between two socially connected users. Rich types of social proximity relations can be preserved in community structures, and meanwhile community detection algorithms have been explored in collaborative filtering to enhance the performance of RSs [7, 8].

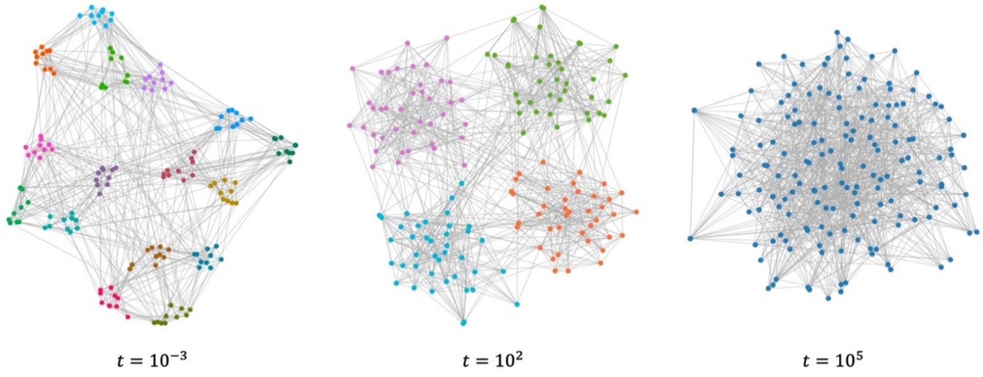


Figure 1.2: An example of community evolution along different timestamps[3].

Despite its popularity and widely recognized applicability, community detection algorithm in social recommendations still suffers from several challenges and limitations: (1) Different from widely used homogeneous networks which include only same-typed objects or links, the heterogeneous network is seldom studied but more commonly seen in the real world. (2) Most online social networks are intrinsically dynamic with addition/deletion of edges and nodes. Meanwhile, similar to network structure, node attributes also change

[3] https://zexihuang.com/projects/

naturally such that new content patterns may emerge and outdated content patterns will fade. An example that illustrates the aforementioned properties can be seen in Figure 1.2, in which different colours represent different communities in the heterogeneous network. The communities evolve at different timestamps, so does the network structure. (3) It is normally not the case that one node in the social network solely belongs to only one community. The significant overlaps among communities make most existing community detection algorithms ineffective.

Multimedia data

To accurately predict the next item the user may interest in or the rating of the item the user may concern, it is essential to capture users' preference and items' characteristics in different aspects by analyzing textual features generated by users and social relations of the target user. Besides, for some types of items such as movies, clothing, videos, etc., visual information considered as an important part of multimedia data, can also be a significant complementary resource when performing recommendations. As shown in Figure 1.3, a user chooses to see different movies at different timestamps according to the movies' poster design style and their plot descriptions. If we want to recommend next movie the user would see, we should consider both the textual and visual information. In the scope of this thesis, visual information refers to image features only.

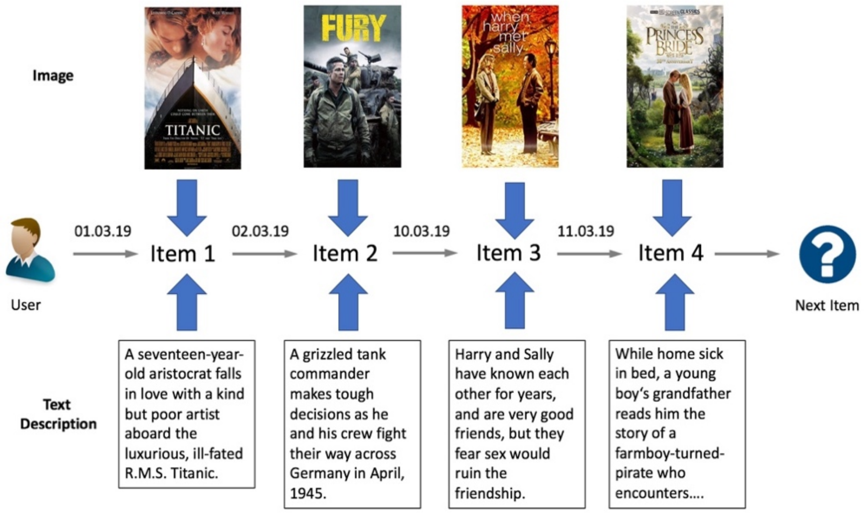


Image
User
01.03.19
Item 1
02.03.19
Item 2
10.03.19
Item 3
11.03.19
Item 4
Next Item
Text Description
A seventeen-year-old aristocrat falls in love with a kind but poor artist aboard the luxurious, ill-fated R.M.S. Titanic.
A grizzled tank commander makes tough decisions as he and his crew fight their way across Germany in April, 1945.
Harry and Sally have known each other for years, and are very good friends, but they fear sex would ruin the friendship.
While home sick in bed, a young boy's grandfather reads him the story of a farmboy-turned-pirate who encounters....

Figure 1.3: Diagram of a user's watching sequence.

Though the effectiveness of incorporating visual information in social recommendations has been verified by recent studies [9, 10], how to make full use of image features is still a thought-provoking issue. Apart from the usage of improving the recommendation performance and alleviating item cold start problem, visual information can also be utilized to strengthen the explainability and increase the transparency of the recommender systems. Meanwhile, the fusion strategies for mutual association learning across modalities inferred from multiple heterogeneous data sources such as reviews and images, remains a challenge in recommendation tasks.

Many challenges and limitations on latent context extraction and modelling in social networks have been discussed from previous parts. However, social recommender systems also face challenges that remain unsolved. Firstly, as many researchers have pointed out, RSs suffer from data sparsity and cold start problems. And they become even more severe in huge social networks. Secondly, facing the abundance of multilingual information in user-generated reviews, RSs need to evolve to effectively deal with the challenge of recommending interesting items with their review languages different from that the users adopted to express their preferences. Apart from these, considering the online environment and frequently changing velocity of social networks, the scalability and updating complexity of learning algorithms in recommendation tasks should also play a pivotal role and be seriously reckoned.

To address the aforementioned gaps, the general goal of our research can be summarised as exploring different aspects of problems on the latent context in social networks. We specifically focus on the areas of latent context extraction which includes topic/aspect-based sentiment analysis, social link analysis and visual context learning, as well as its integration into social recommender systems.

1.2 Research Context

The research in this PhD thesis has been carried out as a part of a four year PhD program at the Department of Computer Science, Norwegian University of Science and Technology. The PhD project is a formal part of the RecTech (Recommendation Technologies) project, funded by the Research Council of Norway under the BIA innovation research program with project number 245469. RecTech is performed in cooperation with Adresseavisen/ Polaris Media, Cxense in Oslo, NTNU in Trondheim and VTT in Finland.

The RecTech project focuses on research and development of the next generation recommender systems for news as well as other social media streams. User profiling and deep content analysis are two main tasks, of which key technologies including computational linguistics, machine learning and big data mining play a central role in RecTech.

In the four years, 25% of the time was spent on teaching duties in courses *TDT4215 - Web Intelligence*, *TDT4290 - Customer Driven Project*, *TDT4110 - Information Technology, Introduction* and *TDT4900 - Computer Science, Master Thesis*. As part of the PhD program, five courses have been attended and successfully passed: *TDT4215 - Web Intelligence* and *DT8108 - Topics in Information Technology* in Spring 2016, *DT8116 - Web Mining* in Autumn 2016, *DT8109 - Business Systems* in Autumn 2017 and finally *DT8122 - Probabilistic Artificial Intelligence* in Summer 2019.

1.3 Research Questions

After having specified the overall challenges and scopes of our work, the main research question conducted in our research is as follows:

- **[RQ]** – *How can we exploit the social network structure and user-generated contents provided in social media streams to improve the effectiveness of recommender systems?*

The principal research question can be divided into the following sub-research questions:

- **[RQ1]** – *How can we extract topics and topic-specific sentiments from social media streams and analyse their evolution?*

  In this research question, we aim to find the possibility of extracting topics and topic-based sentiments simultaneously from news and other social media streams. In particular, we want to explore the influence of interactions among users in online environment on sentiment polarities. We also analyse the efficiency of the extraction algorithms of topics and topic-based sentiments, as well as their evolution over time.

- **[RQ2]** – *To what extent can multilingual topic/aspect and sentiment information extracted from user reviews be used to improve the effectiveness, diversity and novelty of recommendation approaches?*

  With the rapid development of the internet, the Web is becoming more and more multilingual, and users tend to be polyglot. In this context, recommender systems need to evolve to deal with the increasing amount of multilingual information. The intention of this research question is to investigate a potential mechanism which can jointly extract aligned topics/aspects and their associated sentiments in different languages simultaneously. We also want to explore the possibility of leveraging multilingual topics/aspects as well as their associated sentiments as potential resources to improve the interpretability and diversity of recommendation tasks.

- **[RQ3]** – *How can the temporal contexts from large-scale heterogeneous networks be exploited to enhance social recommendation in real-time?*

  Our focus in this research question is on temporal contexts. In particular, this work investigates to learn temporal semantic effects, social relationships and user behavior sequential patterns which are extracted from the heterogeneous network, and the potential possibility to address the issues of temporal dynamics, cold start and context awareness in the social recommender system.

- **[RQ4]** – *Can community information induced from the network structure improve existing graph embedding models for the task of social recommendation?*

  With this research question, the intention is to employ global context, namely community information derived from network structure, in graph embedding models for social recommendations. Specifically, we explore the possibility of learning global community context and local context among users and/or items in a joint manner, as well as tracking the evolution of network structures over time. We also look into the overlapping communities in heterogeneous networks.

- **[RQ5]** – *Can social recommendation benefit from incorporating visual context in terms of performance and interpretability?*

  As visual context can be a significant factor for some domains and thereby affects users' clicking/rating behavior, this research question investigates the possibility of employing visual information for social recommendations. Unlike most previous studies, we intend to develop a feasible mechanism that learns visual context embeddings aligned with its textual descriptions that can improve the recommendation performance, and meanwhile provides explanations on recommendation results.

1.4 Research Approach

The work presented in this thesis followed the Design Science (DS) paradigm in information systems [11], which we consider as the most appropriate research approach for the purpose of our research. In the design science paradigm, the aim is to design and apply new and/or innovative artifacts aimed at human and organizational use. Knowledge and understanding of a problem domain and its possible solution are achieved through the design, application and evaluation process of the artifacts, often performed in iterations [12]. More specifically, the paradigm incorporates six activities in our study: problem identification and motivation, definition of the objectives for the solution, model design and development, experiment design and development, evaluation, and conclusion.

- **Problem Identification and Motivation.** To identify research problems on the basis of the literature review and state-of-the-art knowledge in my specific research field is a crucial step to start my PhD study. To know the existing solutions, as well as their limitations and challenges, is required as a prerequisite before diving into the methodology design and development procedure. The literature is mainly from prestigious international conference proceedings, e.g., ACL, WWW, SIGIR, WSDM, etc, and good international journals, e.g., TKDE, TOIS, etc.
- **Definition the Objectives of the Solution.** After the problem has been identified and formulated, a possible solution will be suggested under certain conditions, such as hypothesis or specific dataset. The objectives are usually from quantitative and qualitative two perspectives, while in our domain, we mainly focus on the former one. The quantitative objectives expect better performance for social recommendation tasks on hit rate, accuracy, ranking, etc., from which exhibits the ability to address the existing limitations and challenges. When developing the manual evaluation scheme related to the recommended item explanation, the use of a questionnaire is also explored.
- **Model Design and Development.** To achieve the desired objectives, there is a need to design our own approaches. Apart from programming knowledge, other domain knowledge related to the study should also be involved, such as statistical learning, probability theory, neural network, etc.
- **Experiment Design and Implementation.** In this stage, a series of experiments through programming are needed to test the approaches and models proposed during our research for a specific purpose. The experiments are designed and implemented in consideration of the available datasets and testing platform.

- **Evaluation.** In our researches, the experimental results are analysed according to quantitatively empirical methods. The results evaluation process can help to assess the consistency of the hypothesis and the designed framework. To achieve this goal, evaluation metrics and baseline approaches should be selected and developed.
- **Communication.** Observations and conclusions can be extracted from the experiments and results analysis in the evaluation step. The derived conclusion can be formulated as research papers or reports.

1.5 Research Contributions

This section summarises five main research contributions in the thesis, in accordance with the research questions presented in Section 1.3.

1.5.1 An novel probabilistic generative model to extract topics and topic-specific sentiments from news streams [C1]

We have proposed a dynamic topic-based sentiment analysis model (DTSA) to extract topic and topic-specific sentiments simultaneously. The proposed approach is inspired by the well-known online multiscale dynamic topic models (OMDT) [13] which is on the basis of the Dirichlet-multinomial framework by assuming that current topic-specific distributions over words were generated based on the multiscale word distributions of the previous epoch. We extend this work by exploring the generation process of online comments, and introducing new parameters in the model learning process that represent the co-effects caused by user interactions and time factor. Thus, the new proposed model of DTSA can capture the evolution of topics and topic-specific sentiment polarities over time.

1.5.2 Multilingual topic/aspect-based sentiment analysis model for improving the interpretability and diversity of review-aware RSs [C2]

With the growth of the Web and the expansion of the international market, textual reviews of different languages have become prevalent in social media and e-commerce, which arouses the interest of multilingual recommendation aiming to model language-independent user/item profiles in a fine-grained scale and thus improve the recommendation performance. In particular, we introduce a multiple instance learning framework for multilingual aspect-based sentiment analysis which uses freely available multilingual word embeddings and only requires light supervision (user-provided ratings). Then a novel dual interactive attention mechanism is proposed which considers both popular and long-tail items for effectively modelling the fine-grained user-item interactions, as well as balancing between recommendation accuracy and diversity.

1.5.3 An advanced graph representation learning approach for enhancing the efficiency of social recommendation [C3]

We construct a heterogeneous user-item (HUI) network and maintain it incrementally for dynamic social recommendation task. In particular, we analyse temporal contexts including the temporal semantic effects, social relationships and user behavior sequential patterns in a unified way by embedding HUI into a shared low dimensional space. Then recommendations can be generated using the encoded representation of temporal contexts through simple search methods or similarity calculations. Our empirical analysis is performed on two real large-scale datasets.

1.5.4 A novel dynamic network embedding model with considering local and global contexts for social recommendation [C4]

We propose a novel multi-granularity dynamic network embedding model (m-DNE) that can recommend relevant users and interesting items by performing an improved network representation learning algorithm on the constructed heterogeneous user-item (HUI) network. Unlike most previous studies which only consider first- or second-order proximity of nodes in HUI network, we incorporate the community-aware high-order proximity of nodes which brings more information to the learned encoded representations for the final recommendation task.

1.5.5 An improved attention-based Recurrent Neural Network with textual and visual fusion for the dynamic explainable recommendation [C5]

Recommender system often works like a black-box where no explanatory information is provided to users while performing recommendation tasks. However, the explanation of recommendation results can usually make it easier for users to make decisions, increase conversion rates and lead to more satisfaction and trust. In this research, the proposed Attentive Recurrent Neural Network (Ante-RNN) model can provide explanations for users by integrating and learning textual and visual contexts. Different from most previous studies which learn image representations only with images, we learn image representations with textual alignment and text representations with topical attention mechanism in a parallel way. Therefore, the explanation of recommendation results can be presented through images aligned with their textual descriptions.

1.6 Publications

In this section, we present the list of scientific papers published during the PhD studies that cover the above contributions. For each paper, we refer to the corresponding chapter in which the content of the paper is included and point out the relevance of the aforementioned research questions.

P1. Peng Liu, Jon Atle Gulla, and Lemei Zhang: *Dynamic topic-based sentiment analysis of large-scale online news*. In Proceedings of the 17th International Conference on Web Information Systems Engineering. WISE 2016.

Summary: The content of this paper is included in Chapter 3 and is aimed at answering the research question **RQ1**.

P2. Peng Liu, Lemei Zhang and Jon Atle Gulla: *Multilingual Review-Aware Deep Recommender System via Aspect-based Sentiment Analysis*. ACM Transactions on Information Systems, TOIS, 2nd round review.

Summary: The content of this paper is included in Chapter 4 and is aimed at answering the research question **RQ2**.

P3. Peng Liu, Lemei Zhang, and Jon Atle Gulla: *Real-time social recommendation based on graph embedding and temporal context*. International Journal of Human-Computer Studies. IJHCS 2019.

Summary: The content of this paper is included in Chapter 5 and is aimed at answering the research question **RQ3**.

P4. Peng Liu, Lemei Zhang, and Jon Atle Gulla: *Learning Multi-granularity Dynamic Network Representations for Social Recommendation*. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. ECML-PKDD 2018.

Summary: The content of this paper is included in Chapter 6 and is aimed at answering the research question **RQ4**.

P5. Peng Liu, Lemei Zhang, and Jon Atle Gulla: *Dynamic attention-based explainable recommendation with textual and visual fusion*. International Journal of Information Processing & Management. IPM 2019.

Summary: The content of this paper is included in Chapter 7 and is aimed at answering the research question **RQ5**.

As a summary, Table 1.1 presents the relations between the papers and our research contributions listed in Section 1.4.

Table 1.1: relations between contributions and publications.

	Contributions				
Papers	**C1**	**C2**	**C3**	**C4**	**C5**
P1	•				
P2		•			
P3			•		
P4				•	
P5					•

Additional Publications. The following papers were published in the course of this PhD, but are not included in the thesis because they are not directly connected to its research topic.

A1. Peng Liu, Cristina Marco and Jon Atle Gulla: *Semi-supervised sentiment analysis for under-resourced language with a sentiment lexicon*. In Proceedings of the 7th International Workshop on News Recommendation and Analytics. INRA 2019.

A2. Cristina Marco, Peng Liu∗ and Jon Atle Gulla: *Cross-lingual sentiment analysis for under-resourced languages using machine translation and sentence embeddings*. In review with Computational Linguistics. (∗ Contributing equally with the first author)

A3. Zhang, Lemei, Peng Liu, and Jon Atle Gulla: *A neural time series forecasting model for user interests prediction on Twitter*. In Proceedings of the 25th International Conference on User Modeling, Adaptation and Personalization. UMAP 2017.

A4. Gulla, Jon Atle, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su: *The Adressa dataset for news recommendation*. In Proceedings of the International Conference on Web Intelligence. WI 2017.

A5. Zhang, Lemei, Peng Liu, and Jon Atle Gulla: *A deep joint network for session-based news recommendations with contextual augmentation*. In Proceedings of the 29th ACM

Conference on Hypertext and Social Media. HT 2018.

A6. Zhang, Lemei, Peng Liu, and Jon Atle Gulla: *Dynamic attention-integrated neural network for session-based news recommendation*. Machine Learning, 108(10), pp.1851-1875, 2019.

1.7 Thesis Structure

The thesis is divided into four main parts. Part I gives an introduction to the main topics of the thesis and summarises the technical background in these areas. Then, Part II-IV present our research on improving the effectiveness of social recommendation by exploiting different latent contexts. In particular, Part II focuses on textual contexts including topics and sentiments, Part III focuses on network structures, and Part IV focuses on visual contexts. Finally, we conclude the thesis and give an overview of future work in Part V. A more detailed outline of the contents is given in the following:

Part I Introduction and Background

Chapter 1 introduces the motivation of our research and the research context. We also summarize the research questions studied, research approach and contributions of the thesis.

Chapter 2 presents an overview of background knowledge and techniques that are needed to understand the work in this thesis.

Part II Exploiting Textual Contexts in Social Recommendation

Chapter 3 describes a framework for extracting topics and topic-specific sentiments from news and other social media streams as well as analyzing their evolutions over time.

Chapter 4 presents a novel multilingual review-aware deep recommendation model for overall rating prediction and item recommendation. It also introduces a multiple instance learning framework for multilingual aspect-based sentiment analysis without using any external resource.

Part III Analyzing Network Structures for Social Recommendation

Chapter 5 describes a framework for integrating the temporal semantic effects, social relationships and user behavior sequential patterns into the recommendation process to alleviate cold start issues.

Chapter 6 studies the problem of detecting communities in social networks which can be utilized to enhance the user and item representation learning as well as improving the recommendation performance.

Part IV Utilizing Visual Contexts for Social Recommendation

Chapter 7 presents a novel framework for combining visual image information with text descriptions in the social recommender system and providing multi-model explanations on the recommendation results.

Part V Conclusions and Future Work

Chapter 8 presents the conclusions of our research, and gives possible research directions for future work.

Chapter 2
Background and Literature Review

In this chapter, we briefly describe fundamental techniques and related work in the research area of recommender systems that can facilitate the understanding of the content of this thesis. The chapter is organised as follows. In Section 2.1, we review the state-of-the-art techniques related to recommender systems, including traditional and deep learning based methods. Besides, common challenges on recommender systems that need to be adressed are also discussed. In Section 2.2, several popular evaluation metrics for recommender systems are introduced and explained. Finally, we present recent related work on the extraction of dynamic latent contexts defined in chapter 1, i.e. topic-based sentiment, community and visual feature, and their applications in social recommender systems in Section 2.3.

2.1 Recommender Systems

2.1.1 Traditional Recommender Systems

Recommender systems have shown to be a valuable part of modern applications to deal with such situations of information overload. Generally speaking, the basic machenism of recommender system is to recommend relevant items to users by modeling users' preferences upon the historical records or provide additional items that users may be interested in. This subchapter summerises some well-known and prevalent recommendation approaches including *collaborative filtering*, *content-based recommendations* and *hybrid approaches*, from which the modern recommender systems are derived from.

Collaborative Filtering Recommendation

The term "collaborative filtering" refers to the use of ratings from multiple user in a collaborative way to predict missing ratings [14]. Collaborative filtering (CF) systems recommend items to a user based on the notion that if other users have similar rating behavior with the target user, they will also have similar preferences with other items in the future. Traditional CF does not consider user and item attributes but focus on user-item interactions. A general classification of CF includes two main sub-categories: memory-based [16] and model-based [17] approaches.

Memory-based CF approaches, also called neighborhood-based CF, can be broadly devide into user-item CF and item-item CF. For user-item CF approach, the recommender system recommends items to the target user according to his/her neighbour users who have similar tastes. By the contrast, item-item CF system typically first finds users that like the same item with the target user, and then recommends other items that these users also liked. Thus, the key point of memory-based CF is to find a number of reliable neighbors for the target users or items when generating recommendations. In practice, memory-based CF needs to solve the problem of computing similarty and aggregating ratings [18]. Some basic and widely used similarity metrics include cosine similarity [16], Pearson correlation [19] and Jaccord coefficient [20]. In the user-item CF scenario, let vector $\boldsymbol{x}$ and $\boldsymbol{y}$ represent user X and Y. Thus, the cosine similarity can be defined as:

$$similarity(\boldsymbol{x},\boldsymbol{y}) = \cos(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{xy}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$

Pearson correlation coefficient, $pcc(\boldsymbol{x},\boldsymbol{y})$ can be defined as:

$$similarity(\boldsymbol{x},\boldsymbol{y}) = pcc(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}$$

Where x_i and y_i are elements of $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\bar{x}$ and $\bar{y}$ are expectations of $\boldsymbol{x}$ and $\boldsymbol{y}$.

Standard Jaccard similarity usually is used for calculating similarity of binary vectors. As for real-value vector pairs, a generalized version of Jaccard similarity is more suitable, which can be defined as:

$$similarity(\boldsymbol{x},\boldsymbol{y}) = J(\boldsymbol{x},\boldsymbol{y}) = \frac{\sum_i \min(x_i, y_i)}{\sum_i \max(x_i, y_i)}$$

The standard definition of Jaccard similarity is:

$$similarity(\boldsymbol{x},\boldsymbol{y}) = J(\boldsymbol{x},\boldsymbol{y}) = \frac{|\boldsymbol{x} \cap \boldsymbol{y}|}{|\boldsymbol{x} \cup \boldsymbol{y}|}$$

Once the similarity scores are achieved, recommendations can be generated according to the most top-N similar users. Disadvatages of memory-based approaches include data sparsity and cold-start issues when only considering user-item rating matrix. Detailed description on these challenges can be found in Section 2.1.3. Another key problem a memory-based CF has to solve is scalability. More memory space is needed with the increasing number of users and items since full data should be available for generating recommendations [21]. The main advantages of memory-based recommendations are that they are simple to implement and the generated recommendations can be well-explained [14].

In contrast to memory-based approaches, model-based approaches assume a model to generate the ratings and apply data mining and machine learning techniques to find patterns from training data [18] instead of explicitly calculating the similarities between users and items as memory-based CF. In practice, model-based approaches can usually generate better recommendation performance since they can better adapt the training data in offline model and scale up for large-scale datasets in online prediction process [22]. Typical examples of model-based methods include latent factor models (also called embedding models) [23, 24], SVM [25], representation learning based models [26, 27], tree models [28], clustering based models [29, 30], and neural network based models [31, 32], etc. Among these, latent factor models, representation learning based models and neural network based models are most successful in recent literature and are also the fundamental models of our proposed approaches in this thesis. Full details on these models can be found in Section 2.1.2.

Content-based Recommendation

As mentioned in the previous part, traditional CF based approaches only consider the explicit or implicit feedback between users and items such as ratings, clicks, etc without leveraging domain knowledge. In contrast, content-based (CB) recommendation approaches use the description of items to build item profiles. User profiles can be derived and built by analyzing user behavior and feedback or by asking explicitly about interest and preferences. Then CB recommender systems generate recommendations by analyzing the features of items or characteristics of users with achieved item and user profile to recommend unseen items similar with which the target user liked before. Similarity can be achieved through dot product of item and user profile. Different from CF methods in which unit elements (in vectors or sets) represent ratings or other implicit feedback, CB methods measure similarities between two sets of multi-valued charicteristics, such as keywords, genre, actors, etc. in movie domain.

In order to extract useful information from structured or unstructured textual descriptions and convert to machine readable format, a series of preprocessing procedure needs to perform, for instance, splitting sentences into keywords. These keywords need to be filtered and selected according to certain measurements. One of the widely used measures for keyword selection is *Term Frequency/Inverse Document Frequency (TF-IDF)* [33]. Assuming $D = \{d_1, d_2, \ldots, d_{|D|}\}$ is a set of documents (items) and k_i represents keyword. Let $f_{i,j}$ represents the number of times k_i appears in d_j. Then the term frequency of keyword k_i in document d_j can be defined as

$$TF_{i,j} = \frac{f_{i,j}}{\max f_{z,j}}$$

Where $\max f_{z,j}$ represents the maximum number of times keyword k_z that appears in document d_j. However, some keywords with high term frequency in many documents are with little use in distinguishing between relevant document and a non-relevant one. Therefore, IDF is often adopted in combination with TF.

$$IDF_i = \log(\frac{|D|}{n_i})$$

Where n_i denotes the number of documents that keyword k_i appears. Then, we can get a score $w_{i,j}$ representing the importance (weight) that keyword k_i is in distinguishing relevant document d_j from the other ones.

$$w_{i,j} = TF_{i,j} \times IDF_i$$

The document is, therefore, represented by a vector with weights of TF-IDF values instead of unary values or rating scores.

Item profile can be built with TF-IDF scores of keywords, or/and other attributes like tags, meta-data, etc. User profiles thus can be built by aggregating profiles of items rated or consumed before. Candidate items can be achieved according to the dot-product between item profile and user profile.

CF recommendation approaches generally have two advantages. First, it does not require large user groups or a rating history to achieve reasonable recommendation accuracy.

Additionally, item cold-start issues can be alleviated by integrating new item attributes [6]. On the other hand, the main challenge is the acquisition of some of the features which may need costly manual effort. Besides, user cold-start problem cannot be solved by CB approaches.

Hybrid Recommendation Approaches

Purely CF or CB approaches often have their own limitations as mentioned in previous parts. Thus in practice, recommender systems usually combine both CF and CB approaches which are refferd to as the hybrid recommendations. There are several ways to combine CF and CB methods into a hybrid recommender system [33]:

(1) Implement CF and CB methods separately and combine their predictions. The combination can be a weighted summation of the results from different separate parts [34] or through a voting scheme [35].

(2) Incorporate some CB features into CF systems. In [35], the author constructs the content-based profiles for each user which are then used to calculate the similarities between two users. Data sparsity issues that often appear in CF methods can be somewhat alleviated. Similarly, [36] uses a CF method where the traditional user's ratings vector is augmented with additional ratings, which are calculated using a pure content-based predictor.

(3) Incorporate some CF characteristics into CB systems. Dimensionality reduction techniques are the most popular in such category. An typical example can be found with latent semantic indexing (LSI) model adopted to create a collaboraive view of user profiles, resulting in the improvement of the recommendation performance compared with CB method [37].

(4) Construct a unified model that integrates both CF and CB characteristics. This approach has become more and more popular in recent years. For instance, [38] propose a unified probabilistic method for combining collaborative and content-based recommendations based on probabilistic latent semantic analysis. In [39], Liu et al. propose to combine topic clustering, attention degree and CF method into a mobile recommender system for blog articles.

2.1.2 Deep Learning-based Recommender Systems

Deep learning has proved its effectiveness in Speech Recognition, Computer Vision and Natural Language Processing. Recently, a surge of interest in applying deep learning to recommendation tasks has emerged, and some recent advances show state-of-the-art performance. To name a few, He et al. [32] propose a neural network based CF framework NCF, which combines classic matrix factorization (MF) and multi-layer perceptron. NCF has demonstrated its improvements compared with MF. In [40], He et al. extend their work by designing Neural Factorization Machine (NFM), of which additional neural layers are stacked on top of the embedding vector of factorization machine (FM). Likewise, NFM has proved its superior performance compared with FM.

Both MF and FM belong to latent factor models (LFM), which decomposes the high-dimensional user-item rating matrix into low-dimensional user and item latent matrices. The final predictions can be achieved through the dot product of user and item embeddings representing latent features of the user and the item. Though LFM show the high efficiency in recommendations, our main focus lies on the deep learning based approaches. Thus the following part is dedicated to discussing deep learning based recommendaitons, especially from two perspectives: representation learning and neural network based approaches.

Representation Learning-based Approaches

Representation Learning (RL) based approaches which aim to learn low-dimensional node embedding have proved to be effective. In contrast with LMF which employ global statistical information of user-item interaction data to learn the model, RL based approaches capture local item relationships. In the following part, some basic RL methods are briefly introduced especially two well-known models of DeepWalk and LINE, which also are basis of our algorithm in this theis. Then we shortly recapitulate RL based recommendation algorithms.

DeepWalk [41] learns the latent vertex representation in networks by making an analogy between natural language sentence and short random walk sentence, which generalizes the idea of the Skip-Gram model [42] that utilizes word context in sentences to learn latent representations of words. In specific, it firstly performs random walks over the given network G which generates a set of random walk sequences. Assuming a random walk sequence is $W_{v_i} = \{v_1, v_2, \dots, v_L\}$ with length L. Following Skip-Gram, DeepWalk learns the latent representation of a given vertex v_i by the optimization problem:

$$\min_f - \log \Pr\left(\{v_{i-w}, \dots, v_{i-1}, v_{i+1}, \dots, v_{i+w}\} | f(v_i)\right)$$

Where $\{v_{i-w}, \dots, v_{i-1}, v_{i+1}, \dots, v_{i+w}\}$ represents the context verties of vertex v_i within window size of w . $f(v_i)$ denotes a mapping function representing the latent social representation associated with v_i in G. Given a vertex sequence $s = \{v_1, \dots, v_{|s|}\}, v_i \in s$, making conditional independence assumption, the optimization problem can be formulated by maximizing the average log probability:

$$\mathcal{L}(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-w \le j \le i+w} \log \Pr\,(v_j | f(v_i))$$

Solving the optimization problem build representations that capture the shared similarities in local graph structure between vertices in a continuous vector space. Vertices which have similar neighborhoods will acquire similar representations which means they will be represented closely in the new embedding space. The time complexity of DeepWalk is $O(|V| \log |V|)$.

Another highly successful embedding method is **Large-scale Information Network Enbedding (LINE)** [43]. Different from DeepWalk which adopts random walks to preserve network structure, LINE learns vertex representations by explicitly modelling the first-order and the second-order proximity. The first-order proximity can be preserved through minimizing the following objective function:

$$O_1 = d(\hat{p}_1(\cdot,\cdot), p_1(\cdot,\cdot))$$

Where $d(\cdot,\cdot)$ represents the KL-divergence of the two distributions. $p_1(\cdot,\cdot)$ denotes the joint distribution modeled by the latent representations u_i and u_j of a vertex pair v_i and v_j. $\hat{p}_1(\cdot,\cdot)$ is the empirical probability between v_i and v_j pair.

To preserve the second-order proximity is to minimize the objective function as bellow:

$$O_2 = \sum_{i \in V} \lambda_i d(\hat{p}_2(\cdot\,|v_i), p_2(\cdot\,|v_i))$$

Where λ_i denotes the prestige of vertex v_i in the network. $p_2(\cdot\,|v_i)$ represents the context conditional distribution, while $\hat{p}_2(\cdot\,|v_i)$ is the empirical conditional distribution. In the calculation of the second-order proximity, each vertex $v_i \in V$ has two roles: the vertex itself and context of other vertices.

By training the two objective functions separately, LINE can preserve the first- and second-order proximity, and the final vertex representation can be achieved through the concatenation of the two output embeddings. The complexity of LINE is $O(a|E|)$ where a is the average degree of the graph.

Other representative RL approaches include Node2Vec [44] which extends DeepWalk with a controlled path sampling process which requires $O(|V|\log|V| + |V|a^2)$. Similar with DeepWal and LINE, Node2Vec does not design for heterogeneous networks nor aware of community structure. Metapath2vec [45] extends the network embedding methods to heterogeneous network by introducing metapath based random walk with the complexity of $O(a|E||V|)$. PTE [46] utilizes labels of words and constructs a large-scale heterogeneous text network to learn predictive embedding vectors for words with complexity of $O(a|E|)$. The above-mentioned approaches can model heterogeneous network but are still not community preserving. There is little work that tries to take into account community structure and dynamic environment with RL based approaches. Cavallari et al. [47] proposes a community embedding framework, ComE, which adopt global community structure to optimize node embedding results with relatively lower complexity of $O(|V| + |E|)$ but on homogeneous and static network. In [48], DANE performs network embedding in a dynamic environment also for homogeneous network with barely local structure of nodes, and thus ignores the importance of the high-order proximity. The online complexity of DANE is $O(|V|)$. M-NMF [49] constructs the modularity matrix, then applies non-negative matrix factorization to learn node embedding and community detection together with a higher complexity proportional to $O(|V|^2)$ based on static network.

In recommendation domain, an item in a user's consumed sequence can be seen as an analogy to a word appearing in a sequence, which paves the way for the RL based RSs. Many ideas fallen in this type attempt to learn representations of users and items in an embedding space. The authors of [50] propose Item2Vec which adopts Word2Vec to capture the relations between different items in CF datasets resulting in competitive experimental results with an item-based CF using SVD. A contemporaneous work of [27] proposes Meta-Prod2Vec model that leverages past user interactions with items and their attributes to compute low-dimensional embeddings of items. Specifically, the item metadata is injected into the model as side information to regularize the item embeddings.

Different from Item2Vec and Meta-Prod2Vec that only learn item representations with item embedding techniques while ingoring user representations, some approaches take endeavors to learn both user and itm representations for personalized recommendations. Grbovic et al. [51] propose the User2Vec model that simultaneously learns user and product representations by considering the user as a "global context". Later, inspired by the success of word embedding model, CoFactor model proposed by Liang et al. [26] decomposes the user-item interaction matrix and the item-item co-occurrence matrix with shared item latent factors. The co-occurrence matrix encodes the number of users that have consumed both items for each pair of items. The authors of [52] design a unified Bayesian framework MRLR to learn user and item embeddings from a multi-level item organization to achieve the goal of personalized recommendation.

Neutral Network-based Approaches

Recently, several studies have been done to use neural network based models including deep learning techniques for recommendation tasks. Yu et al. [53] represent a basket acquired by pooling operation as the input layer of RNN, which outperforms the state-ofthe-art methods for next basket recommendation. Song [54] propose a multi-rate Long Short-Term Memory (LSTM) with considering both long-term static and short-term temporal user preferences for commercial news recommendation. Hidasi et al. [55] propose to use RNN to model whole sequences of session click IDs. In a later work, they [56] extend their previous work by combining rich features of clicked items such as item IDs, textual descriptions, and images. They use different RNNs to represent different types of features and train those networks in a parallel fashion.

More recently, with the ability to express, store and manipulate the records explicitly, dynamically and effectively, external memory networks (EMN) [57] have shown their promising performance for recommender systems. For instance, Chen et al. [58] proposed a novel framework integrating recommender system with external User Memory Networks which could store and update users' historical records explicitly. Huang et al. [59] proposed to extend the RNN-based sequential recommendation by incorporating the knowledge-enhanced Key-Value Memory Network (KV-MN) for enhancing the representation of user preference.

Autoencoders have been another popular choice of neural network based approaches. In [60], the authors propose the marginalized Denoising Auto-Encoder (mDAE) model that performs better than denoising auto-encode but with fewer training epochs by taking into consideration infinitely many corrupted copies of the training data in every epoch. Later, to improve the top-N recommendations, Wu et. al [61] present a denoising auto-encoder based approach, Collaborative Denoising Auto-Encoder (CDAE) which learns distributed representations of users and items via formulating the user-item feedback data with denoising auto-encoder structure. The authors of [62] generalize contractive auto-encoder paradigm into MF framework that jointly model content information as representations and leverage implicit user feedback to make recommendations.

Other approaches for recommendation tasks adopt neural network to extract features from unstructured content such as music or images which are then used together with more conventional CF models. Wang et al. [31] introduced a more generic approach whereby a deep network is used to extract generic content-features from any types of items, these features are then incorporated in a standard collaborative filtering model to enhance the recommendation performance. Van den Oord et al. [63] proposed a somewhat similar hybrid method exploiting a convolutional deep network to learn features from content descriptions of songs, which are then used in a CF model to tackle the data sparsity problem. The difference is that they use CNNs for feature learning rather than auto-encoders.

2.1.3 Common Challenges in Recommender Systems

Recommender systems have undoubtedly gained much success and enhanced customers' satisfactions in various domiains. However, there are still many open issues and challenges that need to be addressed and considered as research topics. This sub-section describes several prevalent challenges concerned by recent studies.

Cold-Start Problem

As one of the most known problems in RSs, cold-start issue has gain much attention in research area, where no prior events like ratings or clicks, are known for certain users or items. There are usually two types of cold-start problems: cold-start users and cold-start items. In the case of cold-start users, the system does not have information about their preferences in order to make recommendations, which may lead to the loss of new users due to the low accuracy of recommendations in the early stage. The case of cold-start items usually refers to news items arrived in the system or items that have not consumed by any users. The item cold-start problem may make new item miss the opportunity to be recommended and remain "cold" all the time [64]. Pure CF approaches usually encounter such issue as they essentially use information about the users and the items, whereas CB approaches are less affected by cold-start items but still need to face the problem of new users. Auxiliary information of new users and items are often required to solve cold-start problems. For instance, the authors of [65, 66] interview new users to extract their interest for user cold-start scenario. The friend relationships in social networks have been verified for cold-start issues [67, 68]. In addition, the authors of [31] try to solve the itme cold-start issue by using content of new items which is then adopted to calculate the correlation between the new items and existing ones.

Data Sparsity Problem

Cold-start problem is normally refered together with data sparsity issue especially in pure CF approaches as the user-item matrix become sparse with the increasing number of users and items in social network. Another reason behind data sparsity is that most users do not rate most of items and the available ratings are usually sparse, which also refers to long-tail phenomenon [69]. Many researches have attempted to reduce this problem. An example can be found in [70], the authors propose to use trust inferences which refer to transitive associations between users in an underlying social network as valuable auxiliary sources to deal with data sparsity problem.

Over-specialization Problem

Many recommendation algorithms, especially CB methods, are known to be over-specialized in the sense of recommending items that are very similar to those already known by the users [71]. Providing a list of very similar recommendations, though relevant to the users' preferences, does somehow limit the quality of recommendation in the diversity scope [72]. As a solution, in [73], the authors propose to embed CF method in order to provide users with a diverse choice of items.

Scalability

With enormous growth of information over social network, recommender systems are facing an explosion of data, and thus it is a great challenge to handle with continuously increasing demand such as millions of users and items in real time. The problem of systems in processing growing amount of information in a graceful manner is called scalability issue. A common way to deal with this issue is to apply clustering algorithms. In [74], Das et al. use a combination of (MinHash) clustering and distributed computing based on the MapReduce framework to make the approach scalable in the Google News system. The authors of [75] propose to build a hierarchy of news clusters with different clustering techniques, which is then used to offer a set of articles similar with the users' reading preferences. In addition, Medo et al. [76] construct a local neighborhood network of users by only keeping the most similar meighbors for each user according to the similarity of their rating behavior. The item rated by the user can be propagated along the direct edge of the graph to his/her "followers" further. Finally, the recommendations are provided on the basis of the ratings of the target user's neighbors.

Though many possible solutions have been proposed to address scalability issue to a certain degree in recommeder systems, the effect is limited facing the continuous increasing data online. Therefore, scalability remains one of the key problems in recommendation domain and further researches and scalability-oriented evaluations are necessary.

2.2 Evaluation of Recommender Systems

For evaluation, the recommendation dataset will first be split into training and testing set. The recommendation models are learned on training set and then evaluated on testing set. For instance, we can use 80% of dataset for model training, while the rest 20% is left for testing. Model parameters can be tuned through cross validation or leave-one-out evaluation. In this thesis, all evaluation is based on the offline protocol. Evaluation process is performed on testing set. Many metrics can be used to assess the recommendation performance.

2.2.1 Evaluation on Prediction Accuracy

Metrics that are widely used to measure prediction accuracy of recommender systems include Mean Absolute Error (MAE), Mean Square Error (MSE) and Root Mean Square Error (RMSE).

Assuming that E is the set of entries in the test set used for offline evaluation. Each entry in E represents a user-item index pair denoted as (u, j). Let r_{uj} and $\hat{r}_{uj}$ represent ground truth rating of user u on item j and predicted rating accordingly. Thus the evaluation metrics on prediction accuracy can be defined as:

Mean Absolute Error (MAE) measures the average absolute deviation between the real and predicted rating.

$$MAE = \frac{1}{|E|} \sum_{(u,j) \in E} |\hat{r}_{uj} - r_{uj}|$$

Mean Square Error (MSE) put emphasis on large errors compared with MAE.

$$MSE = \frac{1}{|E|} \sum_{(u,j) \in E} \left(\hat{r}_{uj} - r_{uj}\right)^2$$

Root Mean Square Error (RMSE) is the square-root of the MSE, and it is in unit ratings. Similar with MSE, RMSE tends to disproportionately penalize large errors.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{|E|} \sum_{(u,j) \in E} \left(\hat{r}_{uj} - r_{uj}\right)^2}$$

The abovementioned metrics are designed for rating predictions but are unfit to evaluate if the recommeder system is capable of making relevant recommendations. Fundamental and well-known metrics that are designed to measure a set of ranked list generated by recommender systems are Precision, Recall and F1-measure. Assuming the top-k set of ranked items generated by recommender systems is shown to the user. Let $S(k)$ be the set of recommended items and $|S(k)| = k$. Let R be the true set of relevant items (ground-truth positives) that are consumed by the user. Note that k is usually defined far less than the number of all items since users care more about the items ranked at the top positions. Then the relevant metrics can be defined as follows:

Precision measures the fraction of relevant items recommended in the list.

$$Precision = \frac{|R \cap S(k)|}{|S(k)|}$$

Recall measures which fraction of the relevant items have been consumed in the set of recommendations.

$$Rcall = \frac{|R \cap S(k)|}{|R|}$$

F1-measure is the harmonic mean between the precision and the recall. It provides a better quantification than either precision or recall since the trade-off between precision and recall is not necessary monotonic, which is to say, an increase in recall does not always lead to a reduction in precision.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

2.2.2 Evaluation on Ranking

As mentioned in Section 2.2.1, users usually receive a ranked list generated from recommender systems with top-k items. Ideally, the user's interested item should appear in the first place of the list. Thus recommendations can be studied as a ranking problem and ranking measures such as Mean Reciprocal Rank (MRR), the average reciprocal hit rate (ARHR) and discounted cumulative gain (DCG) can be adopted to evaluate the performance of recommendations.

Let p_j be the rank of item j in the top-k list, U be the set of users and I be the set of items in the test set. r_{uj} represents the rating socre of user u on item j. However, for implicit feedback such as click event, $r_{uj} \in \{0,1\}$ is a binary value with $r_{uj} = 1$ being a "hit" where the user has clicked on item j and $r_{uj} = 0$ otherwise.

Discounted Cumulative Gain (DCG) considers a log-arithmic decay in users' interest and it is normally calculated over a recommendation list of specific size k instead of using all the items.

$$DCG = \frac{1}{|U|} \sum_{u \in U} \sum_{j \in I, v_j \leq k} \frac{2^{r_{uj}} - 1}{\log_2(v_j + 1)}$$

Another widely used metric that is derived from DCG is its normalized verion, called the **normalized discounted cumulative gain (NDCG)** which is defined as ratio of the DCG to its ideal value.

$$NDCG = \frac{DCG}{IDCG}$$

Where IDCG denotes ideal discounted cumulative gain computed with the same equation as DCG except using the ground-truth rankings in the computation.

Average Reciprocal Hit Rate (ARHR) is designed for implicit feedback and $r_{uj} \in \{0,1\}$ where $r_{uj} = 1$ represents "hit" and otherwise $r_{uj} = 0$. The default values are set to 0.

Similarly, ARHR is calculated over a recommended list of size k.

$$ARHR = \sum_{j \in I, v_j \leq k} \frac{r_{uj}}{v_j}$$

Mean Reciprocal Rank (MRR) denotes the average of the reciprocal ranks (RR) of all users, where RR measures the rank of the first relevant item in the item list for a user $RR = 1/v_j$. Different from ARHR, the calculation of RR or MRR is not limited to the ranked list with size k.

$$MRR = \frac{1}{|U|} \sum_{U} \frac{1}{v_j}$$

2.2.3 Evaluation on Diversity and Novelty

Diversity implies that the set of proposed recommendations within a single recommended list should be as diverse as possible [14]. Supposing that a recommender system provides a user with top-k items, these k items should present various characteristics in terms of genre, actors, director etc. according to the definition of diversity. Diversity can be measured in terms of the content-centric similarity between pairs of items in the recommendation list. Assuming i and j are two different items in the list, $d(i,j)$ can be used to denote the distance between these two items. Then Intra-List Distance (ILD) of any list $L(k)$ of items with length k recommended to a particular user is given by:

$$ILD = \frac{1}{|I|(|I| - 1)} \sum_{(i,j) \in L(k)} d(i,j)$$

Expected Intro-List Distance (EILD) which is the average similarity between all pairs of items can be reported as the diversity with high values indicating high diversity.

$$EILD = \frac{1}{|U|} \sum_{U} ILD$$

The novelty of a recommender system evaluates the likelihood of a recommender system to give recommendations to the user that they are not aware of, or that they have not seen before. The basic idea of evaluating novelty in offline mode is that novel systems are better at recommending items that are more likely to be selected by the user in the future, rather than at the present time [14]. Another measurement of novelty assums that popular items are less likely to be novel and less popular (long-tail) items are more likely to be unknown to users and their recommendation will lead to higher novelty levels [77]. In the context of this thsis, we adopt the latter measurement to evaluate the novelty of recommender systems.

$$novelty(i) = -\log_2 pop(imp|i)$$

Where $pop(\cdot)$ denotes the popularity of item i, and imp represents the implicit feedback such as click or consume of the item.

2.3 Social Recommender System

With the prevalence of online social networks (e.g. Twitter, Facebook and Goodreads), more and more people like to express their opinions of items and spread them to their social connections (e.g., friends in a undirected social network and followers in a directed social network). The social recommender systems, which leverage different types of contextual information from social networks to enhance recommender systems, have emerged as a promising direction in recent years. Many studies prove that using social contextual information in the recommendation process enhances prediction accuracy [1, 2], reduces the effect of the data sparsity and cold start problems [4, 6], and increases the user's satisfaction.

Social contextual information can be provided by independent and diverse information sources in social networks. In this section, we will provide a detailed literature review about the recommendation problems with three prevalent sources defined in Chapter 1 which aim to exploit latent contexts extracted from these sources to tackle the inherent issues of recommender systems, and thus achieving high recommendation performance. They are respectively review-aware recommendation, community-aware recommendation, and visually-aware recommendation. At last, we will discuss the online learning of social recommendations.

2.3.1 Review-aware Recommendation

Modeling Topic/Aspect-based Sentiment Analysis of Reviews

Sentiment analysis, also known as opinion mining, is one of the hot topics in the field of information retrieval and computational linguistics and has attracted a lot of research attention. The task of sentiment analysis can be conducted on three different levels: review-level, sentence-level, and word/phrase-level. Review-level analysis [78, 79] and sentence-level analysis [80] attempt to classify the sentiment of a whole review or sentence to one of the predefined sentiment polarities, including positive, negative and sometimes neutral. Generally, there are two main approaches for sentiment analysis: machine learning based and lexicon based. Machine learning based approaches [81-83] rely on the famous ML algorithms to solve the sentiment analysis as a regular text classification problem that makes use of syntactic and/or linguistic features. The lexicon-based approaches [84, 85] mostly use a dictionary of sentiment words with their associated sentiment polarity, and incorporate negation and intensification to compute the sentiment polarity for each sentence (or document). Especially, with the revival of interest in deep learning, incorporating the continuous text representation as features has been proved effectively in sentiment classification tasks. Tang et al. [86] develop three neural network models to learn the sentiment-specific word embedding from tweets containing positive and negative emotions. In a later work, they [87] extend their previous method by encoding the intrinsic relations between sentences in the document representation learning which is used for document-level sentiment classification.

In a typical user review scenario, mining user's sentiments at the review-level and sentence-level is useful as it provides more granular analysis of orientation. However, such information is insufficient to support consume decisions without knowing the target aspects or topics that the opinion is expressed on. On the other hand, it is well recognized that the polarities of opinion words are usually dependent on the corresponding aspects/topics. For instance, in mobile phone reviews, we may expect the long battery time but not enjoy the long response time of the operating system. Taking news comments as another example. The word "offensive" is used as a positive orientation in the phrase "offensive player" when discussing sports news, whereas it also has a negative orientation when used in the phrase "offensive behaviour" referring to political news comments. Therefore, it is necessary and appealing to consider aspects/topics when conducting sentiment analysis for social recommender systems.

There are many existing works on topic/aspect-based sentiment analysis. One approach is to use frequent itemset mining algorithm to extract frequent nouns and noun phrases as aspect candidates [88–90]. The main limitations of frequent-based methods are that they do not group related aspects together and can not extract implicit aspect expressions. Sequential labeling techniques are also used to extract aspects and sentiments from reviews [91–93]. These supervised methods suffer from the hardness to obtain labeled training data.

In recent years, several unsupervised unified topic/aspect and sentiment models have been proposed. Lin et al. [15] introduce sentiment polarities into topic modelling for the first time and present a framework called joint sentiment-topic (JST) model which can extract the mixture of aspects and different sentiment polarities for products and services. More specifically, as described by Figure 2.1, the JST is a four-layer hierarchical Bayesian model, where sentiment labels are associated with documents, under which topics are associated with sentiment labels and words are associated with both sentiment labels and topics.

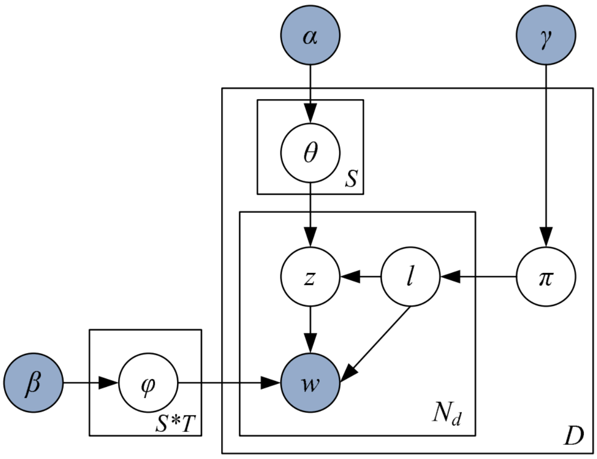


Figure 2.1: Graphical model representation of JST [15].

The main contribution of JST compared with the basic LDA (Latent Dirichlet Allocation) is that the authors introduce an additional sentiment layer between the document and the topic layers to model document sentiments. Different from the generation process of LDA, the procedure for generating a word w_i in document d under JST boils down to three stages. First, one chooses a sentiment label l from the per-document sentiment distribution π_d. Following that, one chooses a topic from the topic distribution $\theta_{d,l}$, where $\theta_{d,l}$ is conditioned on the sampled sentiment label l. Finally, one draws a word from the per-corpus word distribution conditioned on both topic and sentiment label. It is worth noting that the topic distribution of JST is different from that of LDA. In LDA, there is only one topic distribution θ for each individual document. While in JST, each document is associated with S (the number of sentiment labels) topic distributions, each of which corresponds to a sentiment label l with the same number of topics. This feature essentially provides means for JST to predict the sentiment associated with the extracted topics.

The posterior distribution in the JST model can be approximated using variational inference with the expectation-maximization algorithm [94] or Gibbs sampling [95]. Here the authors introduce Gibbs collapsed sampling for inferring the posterior distributions over topics and sentiments. For each iteration during the sampling process, given the current values of all other variables and data, the conditional posterior for z_t and l_t by marginalizing out the random variables φ, θ, and π is

$$P(z_t = j, l_t = k|\boldsymbol{w}, \boldsymbol{z}^{-t}, \boldsymbol{l}^{-t}, \alpha, \beta, \gamma) \propto \frac{N_{k,j,w_t}^{-t} + \beta}{N_{k,j}^{-t} + V\beta} \cdot \frac{N_{d,k,j}^{-t} + \alpha}{N_{d,k}^{-t} + T\alpha} \cdot \frac{N_{d,k}^{-t} + \gamma}{N_d^{-t} + S\gamma}$$

where the superscript $-t$ denote a quantity that excludes data from tth position, V is the vocabulary size, S is the number of distinct sentiment labels, and T is the total number of topics. $N_{k,j,w_t}$is the number of times word w_t appeared in topic j and with sentiment label k, $N_{k,j}$ is the number of times words are assigned to topic j and sentiment label k. $N_{d,k,j}$ is the number of times a word from document d being associated with topic j and sentiment label k, and $N_{d,k}$ is the number of times sentiment label k being assigned to some word tokens in document d. N_d is the total number of words in document d.

Then samples obtained from the Markov chain are used to approximate the per-corpus sentiment-topic word distribution φ , per-document sentiment label specific topic distribution θ and per-document sentiment distribution π as follows:

$$\varphi_{k,j,i} = \frac{N_{k,j,i} + \beta}{N_{k,j} + V\beta} ; \quad \theta_{d,k,j} = \frac{N_{d,k,j} + \alpha}{N_{d,k} + T\alpha} ; \pi_{d,k} = \frac{N_{d,k} + \gamma}{N_d + S\gamma}$$

Following the JST method, Aspect-and-Sentiment Unification Model (ASUM) proposed in [96] and Sentiment-Topic model with Decomposed Prior (STDP) proposed in [97] are all based on LDA, which extract sentiments about topics in a static way without consideration of the dynamic nature of documents. Besides, in the field of deep learning, Maas et al. [98] introduce a probabilistic topic model by inferring the polarity of a sentence based on the embeddings of each word it contains. Xiang et al. [99] develop a topic-based sentiment mixture model with topic-specific data in a semi-supervised training framework. Ren et al. [100] extend Xiang's work and propose to learn topic-enriched multi-prototype word embeddings for Twitter sentiment classification. These algorithms improve the sentiment classification accuracy by using topic words and topic distributions, but none of them aim to extract topics together with sentiments and track their evolution at the same time. Moreover, these methods do not take into account the sentiment polarity transformation caused by user interactions.

In recent years, there has been a surge of interest in developing topic models to explore topic evolutions over time. The continuous time dynamic topic model (cDTM) [101] uses Brownian motion to model the latent topics through a sequential collection of documents. In [13], the authors propose online multiscale dynamic topic models (OMDT) which could trace the topic evolution with multiple timescales. It is on the basis of the Dirichlet-multinomial framework by assuming that current topic-specific distributions over words were generated based on the multiscale word distributions of the previous epoch. Wang et al. [102] propose a Temporal-LDA or TMLDA method to mine streams of social text such as the Twitter stream for an author, by modeling the topics and topic transitions that naturally arised in such data. Different from the work of [101], it focuses more on learning the relationship among topics.

None of the aforementioned models take into account time-aware topic-sentiment analysis. Mohamed et al. [103] propose an LDA based topic model for analyzing topic-sentiment evolution over time by modeling time jointly with topic and sentiments, and derive inference algorithm based on Gibbs Sampling process. However, this time-aware topic sentiment (TTS) model could not consider adjusting model parameters in real-time and process online news streams. He et al. [104, 105] introduce Dynamic-JST based on the previously proposed JST model to capture the dynamic temporal characteristic of topic and sentiment tendency. But it does not consider the interactions between user comments and cannot display the topic and sentiment evolution in a separate way. [106] presents a probabilistic model called topic sentiment trend model (TSTM), based on probabilistic latent semantic analysis (PLSA) model. Thus it exists the problems of inferencing on new documents and overfitting the data.

Cross-lingual Sentiment Analysis

The lack of annotated data in under-resourced languages motivates the need to develop cross-lingual sentiment analysis methods. The main task of cross-lingual sentiment analysis is to apply resources in one language to another language [107]. Common approaches for performing cross-lingual sentiment classification include the use of machine translation and parallel corpora.

Pilot studies on cross-lingual sentiment analysis find that machine translation (MT) has reached a point of maturity that enabled the transfer of sentiment across languages. Kim et al. [108] use a machine translation system and subsequently employ a subjectivity analysis system that is developed for English to create subjectivity analysis resources in other languages. Banea et al. [109] apply bootstrapping to build a subjectivity lexicon for Romanian, starting with a set of 60 words which they translate and subsequently filter using a measure of similarity to the original words, based on latent semantic analysis (LSA) scores. There are also a lot of research work focusing on the best combination of translation direction, classifiers and features. These approaches provide a straight-forward way to create new resources by translating annotated corpora or sentiment lexicons. However, they will introduce some noises which hurt the performance of the classifier and cross-lingual adaptation problems (the distribution of words and their polarities sometimes change in cross-lingual contexts). To reduce the noise that translation introduces, Wan et al. [107] create a bilingual co-training approach to leverage annotated English resources to sentiment classification in Chinese reviews. In this work, firstly, machine translation services are used to translate English labelled documents (training documents) into Chinese and similarly, to translate Chinese unlabelled documents into English. The authors use two different views (English and Chinese) in order to exploit the co-training approach into the classification problem. Pan et al. [110] develop a bi-view non-negative matrix tri-factorization model which allows for the incorporation of the sentiment lexical knowledge and training document label knowledge. Lu et al. [111] propose a joint bilingual model to simultaneously learn better monolingual sentiment classifiers for each language by exploiting an unlabeled parallel corpus together with the labeled data available for each language.

There are also approaches which concentrate on parallel corpora instead of machine translation. Meng et al. [112] propose a generative cross-lingual mixture model to leverage unlabeled bilingual parallel data. By fitting parameters to maximize the likelihood of the bilingual parallel data, the proposed model could learn previously unseen sentiment words from the large bilingual parallel data and improve vocabulary coverage significantly. Zhou et al. [113] also use parallel corpora and stacked denoising autoencoders to learn language-independent high-level semantic representations of documents for cross-lingual sentiment classification. Rasooli et al. [114], instead, make use of multiple annotations and bilingual word embeddings to perform cross-lingual sentiment analysis. Popat et al. [115] use parallel corpora and learn clustering algorithms to learn useful cross-lingual features. All of these approaches, however, require large amounts of parallel data, which are not always available in under-resourced languages.

Most of the aforementioned studies are designed on document level or sentence-level. But when we move to a more fine-grained level (i.e., topic or aspect level), there are only a handful of researches that deal with them. One of the difficulties at topic/aspect-level is that the sentiments attach to specific groupings of words, and if these words are mistranslated or their sentiments are incorrectly inferred, there is no way to correctly predict them. Lambert et al. [116] propose a method, based on constrained SMT, to transfer opinionated units across languages by preserving their boundaries. The classifiers trained on this SMT data achieve comparable results to their monolingual version. Almeida et al. [117] introduce dependency-based opinion mining, where dependency trees are used as features for a classifier. Then, with the word aligned parallel text, they leverage bitext projection to transfer the dependency trees from English to Portuguese and perform aspect-level sentiment analysis. Klinger et al. [118] propose a filtering approach based on machine translation quality estimation measures to select only high-quality sentence pairs for annotation projection. Experiments on the German and English product reviews show that this method leads to improvements over using the full set of noisy translations. However, all of the previous approaches assume there is a high-quality machine translation system available for each language pair, which is not always true for under-resourced languages.

The cross-lingual topic model provides a potential solution to help the aspect-level sentiment classification in a target language by transferring knowledge from a source language. Zhang et al. [119] incorporate soft bilingual dictionary-based constraints into Probabilistic Latent Semantic Analysis (PLSA) so that it can extract shared latent topics in text data of different languages. Boyd-Graber et al. [120] develop the MUltilingual TOpic (MUTO) model to exploit matching across languages on term level to detect multilingual latent topics from unaligned texts. However, these models do not consider sentiment factors and thus cannot help crosslingual sentiment analysis. In [121], a topic model based method is proposed to group aspects from different languages into aspect categories, but this model cannot capture the aspect-aware sentiments because aspects and sentiments are not modeled in a unified way. Boyd-Graber and Resnik [122] propose a supervised holistic model which is based on LDA for cross-lingual sentiment classification. Lin et al. [123] propose a crosslingual joint aspect/sentiment model for sentiment classification. In a later work [124], they incorporate the Joint Sentiment/Topic (JST) model and the Aspect and Sentiment Unification Model (ASUM) into an unified framework for aspect-based cross-lingual topic modelling in sentiment classification. Then the framework is evaluated on hotel reviews and product reviews collected from popular websites in different languages: English, Chinese, Spanish, German, French, Italian and Dutch. There are mainly two major drawbacks of these approaches. First, they are unable to capture the contextual information of words which has been proven crucial to preserve topic coherence. Second, parameter-adjusting might be an onerous task when training these models since they have too many parameters.

Social Recommendation Based on Reviews

It is important to notice that the increasingly growing amount of textual reviews users generated contain rich information about user preferences and item descriptions. As shown in recent overviews of the advances in this area [5, 6], the main assumption in using reviews for recommendation is that there is a correlation between the overall star rating and the aspect opinions mentioned in the reviews. Therefore, users' preferences are constructed based on the aspects/topics and his/her opinions expressed in reviews. Some studies model the aspect opinions using latent factors. For example, Jakob et al. [125] utilize LDA and the Subjective Lexicon [126] to extract opinions about aspects from user reviews, and incorporate them into the Matrix Factorization (MF) model. The authors present a model that captures the five types of relations among users, movies, and movie aspects, namely user ratings, item aspects, user opinions on aspects, and rating- and aspect-based user similarities. The involved entities and relations are then treated as feature vectors for running the MRMF algorithm [127], by which the matrix related to each entity is trained under the influences of multiple relations. Wang et al. [128] adopt a semi-supervised method called Double Propagation [129] and LDA algorithms to extract aspect opinions, whereas recommendations are generated via a tensor factorization method that assembles the overall rating matrix $\boldsymbol{R}$ and K aspect rating matrices $\boldsymbol{R}^1, \boldsymbol{R}^2, \dots, \boldsymbol{R}^K$ into a 3-dimensional tensor $\mathcal{R}$, with which CF is performed. During recommendation, users who hold similar sentiments towards item aspects are considered as similar users. Wang et al. [130] propose a probabilistic regression model to infer latent ratings on aspects. The model assumes that a rating on an item is generated through a weighted combination of latent ratings over all the item aspects, where the weights represent the relative emphasis the user has placed on the aspects, and an aspect latent rating depends on the review fragment that discusses such aspect. Using their model, the authors proposed a CF method that personalizes a ranking of items by considering only the reviews written by the k reviewers whose aspect-level rating behavior is most similar to the target user's. Similarly, Ganu et al. [131] propose a clustering-oriented CF method based on the sentiment of aspects in the user reviews. They first build a multi-label text classifier based on the Support Vector Machine (SVM) to classify review sentences into different aspects (called topics in their study) and sentiment categories. Based on the classification of the sentences of a user's reviews, the user's profile for a particular item is then constructed with the weighted <aspect, sentiment> tuples. Using the generated user profiles, a soft clustering algorithm is applied to group users with similar aspect-level preferences. During the recommendation process, the predicted rating of an item I_t for the user U_t is calculated as

$$\Pr(U_t, I_t) = \frac{\sum_{i=1}^{m} U_t(c_i) \times Contribution(c_i, I_t)}{\sum_{i=1}^{m} U_t(c_i)}$$

where $U_t(c_i)$ denotes the probability that the user U_t belongs to cluster c_i, and m denotes the total number of clusters (which is fixed as 300 in their experiment). $Contribution(c_i, I_t)$ represents the contribution from cluster c_i, which is computed as follows:

$$Contribution(c_i, I_t) = \frac{\sum_{j=1}^{n} U_j(c_i) \times rating\left(U_j, I_t\right)}{\sum_{j=1}^{n} U_j(c_i)}$$

where $rating\left(U_j, I_t\right)$ refers to the rating given by user U_j to item I_t.

Instead of addressing the topic/aspect-based sentiment classification and recommendation tasks separately, McAuley and Leskovec [132] present a matrix factorization model called "Hidden Factors as Topics" (HFT) that combines ratings with review text for product recommendations. The model aligns hidden factors in product ratings with hidden topics in product reviews. Essentially, these topics act as regularisers for latent user and product parameters. In this context, an identified topic may not correspond to a particular aspect or may be associated with several aspects, and thus a user may express different opinions for various aspects in the same topic. Nonetheless, the authors show that HTF predicts ratings more accurately than other models that consider either of such data sources in isolation, especially for cold-start items, whose factors cannot be fit from only a few ratings, but from a few reviews. Diao et al. [133] proposed a probabilistic model (JMARS) based on CF and topic modeling. Similarly to Wang et al. [130], JMARS model assumes that review ratings arise from the process of combining ratings associated with different aspects of the evaluated items. In contrast, JMARS jointly models user and item aspect rating distributions. In the same line of the work, Wu et al. [134] propose a unified probabilistic model called Factorized Latent Aspect ModEl (FLAME) which combines the advantages of collaborative filtering and aspect based opinion mining. FLAME learns users' personalized preferences on different aspects from their past reviews, and predicts users' aspect ratings on new items by the opinions of other users with similar tastes. Finally, Chen et al. [135] present LRPPM, a tensor-matrix factorization algorithm that models interactions among users, items and features simultaneously, to learn user preferences from ratings along with textual reviews. Differently to previous work, the proposed method introduces a ranking-based, instead of a rating-based, optimization objective for better understanding user preferences at aspect level.

Another type of research exploits reviews to learn the user's weight preference (i.e., the weights s/he places on different aspects), rather than directly incorporating aspect opinions into the recommending process. In these researches, a user u_m's preference is represented as a vector $\boldsymbol{u}_m = \{w_{m,1}, w_{m,2}, \dots, w_{m,K}\}$, where $w_{m,k}$ denotes the relative importance (weight) of aspect a_k for u_m, and K is the total number of aspects. Particularly, in Liu et al. [136], the weight $w_{m,k}$ is determined by two factors, namely how much the user concerns about the aspect, and how much quality the user requires for such aspect, as follows: $w_{m,k} = concern(u_m, a_k) \times requirement(u_m, a_k)$. If user u_m commented on a_k very frequently in her/his reviews, but other users commented on it less often, $concern(u_m, a_k)$ increases. For $requirement(u_m, a_k)$, if user u_m frequently rates a_k lower than other users across different items, its value is higher. In the paper, the authors developed an adverb-based opinion-feature extraction method that can accommodate the characteristics of Chinese reviews. They also proposed a recommendation method that estimates the relevance score: $relevance(\boldsymbol{u}_m, \boldsymbol{i}_n) = \sum_{j=1}^{K} w_{m,j} \times \frac{v_{n,j}}{\sum_{j=1}^{K} w_{m,j}}$, where $v_{n,j}$ is the average of reviewers' opinions about aspect a_j of item i_n. The method recommends to u_m the top-N items with the highest relevance scores. Differently to Liu et al. [136], Chen et al. [137] focus on situations with sparse data due to scanty reviews supplied by each user. The authors propose a method that first derives the cluster-level preference denoting a group of users' common preference and then use it to refine the user's personal preference. The refined preference can in turn be used to adjust the cluster-level preferences, and the process continues until the two types of preference are stable. The inputs of this method are the extracted <aspect, opinion> pairs from reviews. In the aspect extraction stage, WordNet [138] and SentiWordNet [139] are utilized to group aspect synonyms and determine aspect opinion polarities, respectively. In the recommendation stage, all of the users are first clustered according to their cluster-level preferences, and then the heuristic user-based k-NN is applied within the cluster to which the target user belongs.

An alternative role of reviews in recommendation is considered. Instead of eliciting users' preferences, user reviews can be used to build the enhanced representation of items to augment item ranking. Aciar et al [140] propose an ontology-based item representation with two components: an item quality component containing the user's evaluation of item aspects, and an opinion quality component which indicates the opinion provider's expertise with the reviewed item. The authors use text mining tools to first classify the sentences of each item review as good, bad and quality (that refers to the quality of the opinion); Afterwards, the aspects mentioned in each of the classified sentences are extracted and utilized to build the item profiles. the authors developed a simple content-based recommendation model that ranks items according to both the item profiles and the user's current interest on the aspects, explicitly stated by the user in the current query, or estimated from the aspect frequencies in the user's reviews. Yates et al. [141] combine aspect opinions extracted from reviews and item technical specifications (e.g., a camera's lens and resolution) to build an item profile, which is called the "item value model" $V(i)$. This model indicates the intrinsic value of the item i for the average user. The item price is treated as an indicator of extrinsic value and the dependent variable in the training phase where a SVM model is built on new items to predict their intrinsic values. Assuming there exists a user u's personalized value model $V(u)$ in the same aspect space as $V(i)$, the difference $ChangeinValue(i,u) = \frac{V(u)-V(i)}{V(i)}$ reflects item i's suitability for user u. A user is then recommended with the items having the highest $ChangeinValue$ scores. Dong et al. [142] propose a case-based recommendation method, in which user preference is represented as a query case (i.e., an item the user inputs as the reference for the query). The item profile (to be matched to the query case) is composed of aspects, each of them with sentiment and popularity scores. They applied a shallow natural language processing technique and a statistical method to extract frequent single nouns and bi-gram phrases as item aspects, and identify the opinions expressed about aspects through the opinion pattern mining method proposed in [90]. When generating recommendations, the model matches the user's profile with items whose profiles are highly similar and produce greater sentiment improvements.

Later, Bauman et al. [143] propose a recommendation technique that not only can recommend items of interest to the user but also specific aspects of consumption of the items to further enhance the user experience with those items. In particular, the authors develop a Sentiment Utility Logistic Model (SULM) that simultaneously fits the opinions extracted from reviews and the ratings provided by the users. SULM assumes that a user u's overall level of satisfaction with consuming item i is measured by an utility value $V_{u,i} \in \mathbb{R}$. This overall utility is estimated as a linear combination of the individual (inferred) sentiment utility values for all the aspects in a review. Denoting the set of all parameters by θ, the model estimates θ such that the logistic transformation of the overall utility $\hat{V}_{u,i}(\theta)$ would fit binary ratings $r_{u,i} \in \{0,1\}$ that user u specified for item i. In SULM, the Double Propagation algorithm [129] was adopted to extract item aspect opinions from the user reviews. Musto et al. [144] propose a multi-criteria recommender system based on collaborative filtering (CF) techniques, which exploits the aspect opinion information conveyed by users' reviews to provide a multi-faceted representation of users' interests. For the user-based CF (the item-based case is analogous), the authors present an aspect-based user distance calculated as

$$dist(u,v) = \frac{1}{|I(u,v)|} + \sum_{i \in I(u,v)} \sqrt{\sum\nolimits_{a \in A(u,i) \cap A(v,i)} |R_a(u,i) - R_a(v,i)|^2}$$

where $I(u,v)$ is the set of items rated by both users u and v, $A(u,i)$ is the set of aspects commented in user u's review about item i, and $R_a(u,i)$ is the sentiment rating inferred for aspect a in that review. The similarity between users is then calculated as the opposite of the distance $dist(u,v)$, and ratings are calculated as the weighted sum approach on the top-k neighbors of the user, $R(u,i) = \sum_{j=1}^{k} \frac{sim(u,u_j) \cdot R(u_j,i)}{|sim(u,u_j)|}$. In the paper, the aspect opinions extraction is performed with the SABRE [145] engine. Recently, Li et al. [146] proposed a capsule network-based model, namely CARP, which was capable of reasoning the rating behaviour by discovering the informative logic unit embracing a pair of a viewpoint held by a user and an aspect of an item, and extracting the corresponding sentiments for rating prediction tasks.

2.3.2 Community-aware Recommendation

Community Detection

Community detection in social networks has been a hot research topic at the linkage-based structural analysis for decades. Community structure represents the latent social context of user interactions. Many NLP applications can benefit from knowledges of the underlying communities in a network, such as information retrieval, question answering and recommender systems. There are different notions of a community. Traditionally, a community is a group of people that are better connected within the community than outside of it [147], such as Facebook groups. While in our work, since we analyze the rich-context social networks which have multiple relations, a community refers to a cluster of entities interacting with one another in a coherent manner. Due to the variety of affiliations and interests that an individual may have, this often leads to communities which may have some overlapping structures.

Community detection in networks is typically considered as a graph partition problem, where one seeks to identify dense subgraphs of relationships with relatively weak connections to outlying nodes. Many methods have been proposed along this direction, e.g. graph cut based methods [148], flow based methods [149], modularity based methods [150] and spectral clustering based methods [151, 152] and so on. Graph cut based methods, including NCut [148], try to find an optimal graph partition with the edge weight between partitions minimized or edge weight inside a partition maximized. Due to the NP-complete complexity of this method, approximate solutions have been proposed. Flake et al. [149] propose approximate algorithms based on network flow ideas to partition the network by solving maximum flow problems, where they define community as a set of entities that has small inter-community cuts and large intra-community cuts. Girvan and Newman [153] introduce betweenness centrality to detect communities. After that, Newman and Girvan [150] introduce modularity to measure the overall quality of discovered communities. Modularity evaluates how entities in a community connect with other entities in that community and has been adopted by many community detection literature. Modularity can be optimized by using the eigenvectors of the modularity matrix which gives rise to those spectral clustering based methods [151, 152]. McCallum et al. in [154] study a new community discovery task on u2u-link data. Instead of finding densely connected entities, they seek to find out users with similar connection pattern such as similar voting patterns.

Palla et al. [155] propose a sequential clique percolation (SCP) to generate overlapping communities by merging overlapping k-cliques. [156] proposed link clustering for overlapping community detection by partitioning links instead of vertices.

However, these techniques only use linkage structure for discovering communities. Under the assumption that communities consist of the nodes within dense subgraphs, they generate only structurally meaningful communities, discarding content information. In fact, rich information is encoded in the content of networks such as node content and edge content [156, 158]. Utilizing only linkage structure may fail to detect the topically meaningful communities because the associated nodes/edges with similar contents are not within the same dense region. To overcome this problem, some recent works have shown significant improvements achieved by integrating node/edge content and linkage structure in community detection [159–163]. A discriminative model was proposed in [160] to combine linkage and content analysis for community detection, where a conditional model and a discriminative model were respectively used for linkage analysis and node content analysis. In [161], an edge-induced matrix factorization (EIMF) approach was used to integrate linkage structure and edge content for community detection. Liu et al. [159] developed a Topic-Link LDA model, which combines the topic similarity (edge content similarity) and linkage structure to jointly model topics and author community. Zhou et al. [162] proposed to integrate the structural and attribute similarities into a unified framework through graph augmentation, so as to consider both linkage structure and node attribute. Sachan et al. [163] addressed the problem of discovering topically meaningful communities from social networks by combining three types of information, namely, discussed topics, graph topology and nature of user interactions, whereby generative Bayesian models were introduced for extracting latent communities.

In real social networks, the context of user actions is constantly changing and co-evolving, e.g. with respect to other users' actions, emergent concepts and users' historic preferences. Hence the communities often contain time-evolving heterogeneous relations, and a lot of studies that consider such data characteristics arising from social media streams have been proposed. Specifically, to analyze communities in time-aware networks, in [164], an incremental density-based clustering algorithm IncOrder is proposed for detecting communities by using probabilistic memberships of nodes in each snapshot network. Besides, a Dynamic Bayesian Nonnegative Matrix Factorization (DBNMF) model was proposed in [165] to automatic detect the overlapping communities in dynamic networks with the use of a Bayesian probabilistic model in online and offine, two stages. Sun et al. [166] use the Minimum Description Length (MDL) principle to extract communities and to detect their changes. Lin et al. [167] use an evolutionary clustering criterion [168] to extract community structures based on both observed networked data and historic community structure. Zhu et al. [169] propose a joint matrix factorization combining both linkage and document-term matrices to improve the hypertext classification. Above all, in order to discovery both structurally and topically meaningful communities, the main challenge lies in how to integrate linkage structure and semantic information in the dynamic case in a seamless way.

Social Recommendation Based on Social Links and Communities

Community-aware recommendation has attracted lots of attention from researchers since it leverages diverse social relations to improve the recommendation process, and thus help mitigate the cold-start problem in collaborative filtering. The main premise in this line of research is that users' preferences are likely to be similar to, or influenced by their friends (homogeneity principle). Ma et al. [170] propose a probabilistic matrix factorization based approach to fuse user-item-rating matrix and user-user linkage matrix which is achieved from social networks. In a later work [171], they introduce the social regularization to constrain the matrix factorization objective function in the recommendation algorithm. Considering the taste diversity of each user’s friends, two regularization terms are proposed: (1) average-based regularization that targets to minimize the difference between a user’s latent factors and average of that of his/her friends; (2) individual-based regularization that focuses on latent factor difference between a user and each of his/her friends. The performance of different similarity measures (i.e., Vector Space Similarity and Pearson Correlation Coefficient) is also compared in this work. Jamali et al. [172] present a novel probabilistic matrix factorization model incorporating the mechanism of trust propagation. This model makes recommendations for a user based on the ratings of the users that have direct or indirect social relations with the given user. Shen et al. [173] propose a joint personal and social latent factor (PSLF) model for social recommendation, which utilize both users’ past behaviors and the social relationships. It extracts the social factor vectors of users from the social network based on the mixture membership stochastic blockmodel and integrates them into the user-item space. However, most of the above methods only consider direct friendships in the social network.

In recent years, the flourish of the heterogeneous social networks provides a new environment for recommendation targets. Kouki et al. [1] propose a hybrid approach, HyPER (Hybrid Probabilistic Extensible Recommender), to incorporate and reason over a wide range of information sources. Sun and Han [174] explore the meta structure of the heterogeneous information network to boost similarity searching and other mining tasks. Shi et al. [2] explore a weighted heterogeneous information network and weighted meta path based recommender system (SemRec) to predict the rating scores of users on items. Vahedian et al. [175] explore a random walk sampling approach in which the frequency of edge sampling is a function of edge weight, and applied it to generate extended meta-paths in weighted heterogeneous networks for recommendation. Shi et al. [176] propose a heterogeneous information network embedding based approach to utilize auxiliary information in networks for recommendation. The authors design a new random walk strategy based on meta-paths to derive meaningful node sequences for network embeddings and integrated them into an extended matrix factorization model using a set of fusion functions.

As rich types of social proximity relations can be preserved in community structures, there has also been work on using community detection for recommendation tasks. Ying et al. [177] propose a preference-aware community detection method for item recommendation based on the user preferences and social network structure simultaneously. In this model, communities are detected through the user's social factor and individual preference. Li et al. [8] propose two social recommendation models that incorporate the overlapping community regularization into the matrix factorization framework. One model is to ensure the latent feature vectors of users in the same community is close to each other. Another model is to force the user latent feature vectors to be close to those of her/his communities. Zhao et al. [178] propose a Community-based Matrix Factorization method based on communities extracted using Latent Dirichlet Allocation (LDA) on twitter social networks. Yin et al. [5] propose a two-phase framework systematically combines social activeness and temporal dynamic information to improve the quality of recommendations. The authors first employ a modified PLSA model to discover communities before applying matrix factorization on each community. Bellogin and Parapar [179] construct a user graph using Pearson correlation similarity and apply normalized graph cuts to find clusters of users. These clusters are then used for neighbor selection in user-based collaborative filtering. Cao et al. [180] propose an improved collaborative filtering recommendation algorithm based on community detection which employs the user similarity network. This approach adopts a novel discrete particle swarm algorithm for community detection to reduce the amount of computation in neighbors selection, and then predicts scores of ratings based on communities. However, these approaches do not take into account temporal dynamics of community structures to support recommendation.

In [181], the authors introduce time factor in the temporal and community-aware recommendation approach (TCNSVD). To capture the community drift for different time bins, TCNSVD reruns the community detection algorithm in each bin. However, the retrain procedure becomes time consuming and takes too much resources with the growth of social network. Meanwhile, the community dectection approaches [182, 183] used in TCNSVD do not actually consider dynamic characteristics of the community. While our model could detect the evolution of overlapping communities in heterogeneous network, and incorporate community information into the graph embedding based social recommendation.

2.3.3 Visually-aware Recommendation

Image Feature Extraction

The rapid development of Web 2.0 has enabled people to upload and share multimedia content (e.g., images and videos) in online social networks, such as Flickr and YouTube. The user-contributed multimedia content plays an important role in understanding users' behaviors and modeling items' characteristics. For example, the categories of the photos that a user usually posts in Flickr may reflect what kinds of items s/he likes to see. And users can easily determine whether they like a restaurant based on the images of food and interior ambience of the restaurant. Thus, the visual features which serve as another type of latent contexts are also important complementary information for social recommendations.

Recently, high-level visual features from Deep Convolutional Neural Networks ('Deep CNNs') have seen successes in tasks like image classification [184], object detection [185], and image captioning [186], among others. Furthermore, recent transfer learning studies have demonstrated that CNNs trained on one large dataset (e.g. ImageNet) can be generalized to extract CNN features for other datasets, and outperform state-of-the-art approaches on these new datasets for different visual tasks [185, 187]. These successes provide us an opportunity to incorporate CNN features with highly generic and descriptive abilities into recommender systems.

Generally, most image-based recommender systems leverage feature extractors in the image classification task to obtain visual features, for instance MobileNet [188], VGG [184], Inception [189], ResNet [190], Inception-ResNet [191], etc. He et al. [192] extract visual features using the Caffe reference model [193], and take the output of the second fully-connected layer to obtain an 4096 dimensional visual feature vector for each product image. Another similar example can be found in [194] which leverages the same model to extract visual features. Cui et al. [9] exploit the GoogLeNet [195] which has 22 layers and has been pre-trained on 1.2M ImageNet ILSVRC2014 images to obtain 1024 dimensional visual features. Unlike previous works that use a global vector as the image feature, many studies [196, 197] adopt spatial features of different regions which contain more information of the orginal image in their recommender systems. Specifically, they divide an image into an $N \times N$ grid, and then use the pre-trained VGG network to extract a D-dimensional feature vector for each region of grids. Thus, an image could be represented as a feature matrix $v_I = [v_1, v_2, \dots, v_M]$ where $v_i \in \mathbb{R}^D, i = 1,2, \dots, M$ and $M = N \times N$ is the number of regions in the image. Chen et al. [198] use the $res5c$ layer feature map in the ResNet-152 architecture to construct the region-level features.

With the development of object detection techniques, many algorithms could identify instances of objects or other entities of an image as regions to learn the feature representation with rich semantic meaning. A typical instance is the Faster R-CNN model [199] in conjunction with ResNet-101 [190] pre-trained by Anderson et al. [186] which is exploited in our social recommendation for the first time.

Social Recommendation Based on Images

Recent years have witnessed the increasing popular of image-based recommendation in both industry and academic communities. For effectively discovering user's preference in the visual dimensions, many promising recommender models have been proposed. For example, McAuley et al. [200] develop a recommender system to recommend clothes and accessories by modeling users' visual preferences with the use of visual contents extracted from cloth and accessory images. To improve the performance of top-n recommendation, He et al. [192] further extend the approach by representing each product image as a fixed length vector, which is then infused into the bayesian personalized ranking (BPR) framwork [201]. Geng et al. [202] propose a novel deep learning framework to learn the unified feature representations for topological user nodes and visual images in the large and sparse social network and applied the resulting model to recommender system. To make use of both visual- and textual- features, Cui et al. [9] integrate the product images and item descriptions together to make dynamic Top-N recommendation. Liu et al. [194] adopt neural modeling based on product images to learn the styles of items and preferences of users, which led to improved recommendation performance in the field of e-commerce. Wang et al. [203] introduce image features into point-of-interest (POI) recommendation, and propose an improved probabilistic matrix factorization to model visual content in the context of POI recommendation. Zhang et al. [204] integrated images with reviews and ratings in a multimodal deep learning framework for top-n recommendation. Chen et al. [198] introduce the attention mechanism into CF to model both item- and component- level implicit feedback for multimedia recommendation.

Although the recommendation performance has been improved by incorporating image representations extracted with (convolutional) neural networks, most of the above methods ignore an important advantage of leveraging images for recommendation – its ability to provide intuitive visual explanations. While we make a step further by modelling users' various attentions on different image objects that represents users' visual preferences, resulting in both the improvement of recommendation performance and reasonable visual explainations for the recommended items.

2.3.4 Online Social Recommendation

The notion of providing an effective recommendations has drawn more and more attention, which is to say recommender systems must evolve with their content and offer up-to-date recommendations to their users in real time. Such requirement restricts most offline recommendation methods as they hinder the system's ability to evolve quickly [205]. Thus, the demand for continuous learning and online learning recommender systems has increased. Stern et al. [206] adopt Assumed-Density Filtering (ADF) for online traing that can incrementally take account of new data so the system can immediately reflect the latest user preferences. In [74], an online CF based recommendation method for users of Google News is introduced. The system combines collaborative filtering using MinHash clustering, Probabilistic Latent Semantic Indexing (PLSI), and covisitation counts in a liner way. The authors of [207] develop a preference elicitation framework and an online learning settings to identify the users' preferences according to few questions and meanwhile address the cold-start problem in restaurant recommendations.

More recently, in order to capture the evolution of the recommender systems, Agarwal et al. [208] propose a fast online bilinear factor model to learn item-specific factors through online regression by using a large amount of historical data to initialize the online models and thus reducing the dimensionality of the input features. Diaz-Aviles et al. [209] present Stream Ranking Matrix Factorization, which utilizes a pair-wise approach to matrix factorization in order to optimize the personalized ranking of topics and follows a selective sampling strategy to perform incremental model updates based on active learning principles. Chen et al. [210] extend the online ranking technique and propose a temporal recommender system TeRec, through which, users can get recommendations of topics according to their real-time interests and generate fast feedbacks according to the recommendations when posting tweets. Huang et al. [211] present a practical scalable item-based collaborative filtering algorithm, with the characteristics such as robustness to implicit feedback problem. Subbian et al. [212] propose a probabilistic neighbourhood-based algorithm for performing recommendations in real-time. The recommendation strategies proposed by Huang et al. [211] and Subbian et al. [212] focus on scalability and real-time pruning in recommender system. Our proposed framework considers the combination of the heterogeneous characteristics of social networks and graph-based updating schemes on real-time condition, and thus is substantially different from the above-mentioned systems.

Part II

Exploring Textual Context in Social Recommendation

Chapter 3
Dynamic Topic-Based Sentiment Analysis

Many of today's online news websites and aggregator apps have enabled users to publish their opinions without respect to time and place. Existing works on topic-based sentiment analysis of product reviews cannot be applied to online news directly because of the following two reasons: (1) The dynamic nature of news streams require the topic and sentiment analysis model also to be dynamically updated. (2) The user interactions among news comments can easily lead to inaccurate topic and sentiment extraction. In this chapter, we propose a novel probabilistic generative model (DTSA) to extract topics and the specified sentiments from news streams and analyze their evolution over time simultaneously. DTSA incorporates a multiple timescale model into a generative topic model. Additionally, we further consider the links among news comments to avoid the error caused by user interactions. Finally, we derive distributed online inference procedures to update the model with newly arrived data and show the effectiveness of our proposed model on real-world data sets.

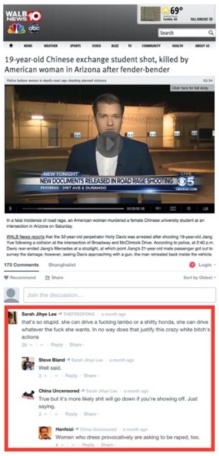
3.1 Introduction

With the growing popularity of both the social media and mobile news apps, an increasingly amount of significant information concerning user opinions and sentiments is being stored online. As important platforms used to describe events happening around the world, online news and comments are the efficient means of conveying positive or negative emotions underlying an opinion and also communicating an affective state, such as happiness, fearfulness, or surprise. It is valuable to extract topics as well as sentimental information from these texts. The governments can detect public sentiments toward policies and emergencies and give feedback in time. The marketers are able to acquire knowledge about the public sentiment environment which supports further analysis and decisions. However, the analysis is impossible to complete manually due to the huge amount of data, and the unstructured data increases the difficulty of machine analysis.

Most earlier studies [13, 96, 97] embrace topics or domains into sentiment analysis model, to improve the accuracy of sentiment classification. To a large extent, it is due to the tightly reliance on domains or topics of sentiment description. The same word in different topics may convey various sentiment polarities. For instance, the word "offensive" is used as positive orientation in the phrase "offensive player" when discussing sports news, whereas it also has negative orientation when used in the phrase "offensive behaviour" referring to political news comments. Thus, sentiment analysis based on topic or domain has far-reaching significance.

In recent years, among the many researches on the approaches to extract topic-based sentiments, most works have focused on analyzing product comments, which are very different from the comments on news and events [213]. More specifically, current studies assume that words in documents have static co-occurrence patterns, which may not be suitable for the task of capturing topic and sentiment shifts in a time-variant data corpus. What is more, the most popular topic models for sentiment analysis rely on batch mode learning which assumes that the training data are all available prior to model learning. When fitting large-scale news streams, the time and memory costs of such approaches will scale linearly with the number of documents analyzed. In addition, many algorithms regard comments as independent individuals, ignoring their connections. In fact, the socialized characteristic of the media platform makes it easier for users to interact with each other, which will result in more connections.

To have a better understanding of user interaction, we list some real comments with interactions of the WALB News website and their corresponding polarities and types in Figure 3.1. The first comment shows a negative sentiment towards the shooting news. The second comment agrees with the first comment's opinion using positive expressions whereas the third person has a little disagreement with the first one. The last comment is based on the previous critiques. In such a situation, we find some drawbacks in the existing methods. First, for example, in the comment "Well said", the existing methods cannot extract the corresponding topics unless considering the interaction with the original news comment. Second, the normal sentiment polarities of positive and negative cannot precisely describe the sentiment polarities of news comments between interactions, so the sentiment classification results will be inaccurate using existing methods. Therefore, user interaction affects both the extraction of topics and sentiments, which renders existing methods less useful.



WALB NEWS 10
69°
Search
NEWS WEATHER SPORTS VIDEO BUZZ TV COMMUNITY HEALTH ABOUT US
NEW TONIGHT
NEW DOCUMENTS RELEASED IN ROAD RAGE SHOOTING
PHOENIX - 31ST AVE & DURANGO
173 Comments
Shanghaiist
Login
Recommend
Share
Sort by Oldest
Join the discussion...
Reply · Share

19-year-old Chinese exchange student shot, killed by American woman in Arizona after fender-bender

In a fatal incidence of road rage, an American woman murdered a female Chinese university student at an intersection in Arizona on Saturday.

WALB News reports that the 32-year-old perpetrator Holly Davis was arrested after shooting 19-year-old Jiang Yue following a collision at the intersection of Broadway and McClintock Drive. According to police, at 2:40 p.m. Davis rear-ended Jiang's Mercedes at a stoplight, at which point Jiang's 21-year-old male passenger got out to survey the damage; however, seeing Davis approaching with a gun, the man retreated back inside the vehicle.

Sarah Jihye Lee → THEFREDFONG · a month ago
that's so stupid. she can drive a fucking lambo or a shitty honda, she can drive whatever the fuck she wants. In no way does that justify this crazy white bitch's actions

Steve Bland → Sarah Jihye Lee · a month ago
Well said.

China Uncensored → Sarah Jihye Lee · a month ago
True but it's more likely shit will go down if you're showing off. Just saying.

Hanfeizi → China Uncensored · a month ago
Women who dress provocatively are asking to be raped, too.

News Comments	Type (comment/response)	Polarity of News (positive/negative)	Polarity of Existing Methods (positive/negative)
That’s so stupid. She can drive a fucking lambo or a shitty honda, she can drive whatever the fuck she wants. In no way does that justify this crazy white bitch’s actions.	comment	negative	negative
Well said.	response	negative	positive
True but it’s more likely shit will go down if you’re showing off. Just saying.	response	negative	negative
Women who dress provocatively are asking to be raped, too.	response	negative	negative

Figure 3.1: News comments and the interactions between them.

In this chapter, we propose a dynamic topic-based sentiment analysis model (DTSA) which is capable of extracting topics and topic-specific sentiments from the online news comment and tracking their evolution over time simultaneously. The DTSA model incorporates the links among new comments to avoid the error caused by user interactions. To efficiently handle streaming data, we derive online inference procedures based on a stochastic Expectation Maximization (EM) algorithm, in which the model is sequentially updated using newly arrived data and the parameters of the previously estimated model. We applied our model to several real data sets and the experimental results demonstrate promising and reasonable performance of our approach.

In summary, our main contributions in this chapter are as follows:

– It proposes a DTSA model where the generation of current sentiment-topic word distributions are influenced by the multiple timescale word distributions at the previous epoch. Considering both the long-timescale dependency and the short-timescale dependency improves the robustness of the model.

– Two special sentiments which represent the transformation of user sentiments–approval and disapproval are introduced to model the links among news comments, which could improve the accuracy of topic-based sentiment classification.

– The proposed DTSA approach adopts a distributed online inference procedure to update the model with newly arrived data, which can be generalized to perform dynamic topic-based sentiment analysis on other large-scale social media streams.

The remainder of this chapter is organized as follows. In Section 3.2, we present our new model. We describe the data sets, experiment settings and the prior information we use in Section 3.3. Section 3.4 shows our experiment results. Finally, we present the conclusions and future work in Section 3.5.

3.2 The DTSA Model

In this section, we propose a novel dynamic topic-based sentiment analysis model (DTSA) for large-scale online news. Firstly, the problem is defined, including the relevant general terms and notations. Then a multiple timescale model and a graphical model are presented in detail. Finally, we describe the estimation and prediction of parameters.

3.2.1 Problem Definition

For convenience of describing the graphical model, we here define the following terms and notations:

In a time-stamped news comments collection, we assume comments are sorted in the ascending order of their time stamps. At each epoch t where the time period for an epoch can be set arbitrarily at an hour, a day, or a year. A stream of comments $C^t = \{c_1^t, c_2^t, c_3^t, \ldots, c_D^t\}$ are received with their order of publication time stamps preserved.

In C^t, D is the number of comments, K is the number of topics, S_1 is the number of normal sentiments (positive and negative), S_2 is the number of special sentiments (approval and disapproval), and $M = S_1 + S_2$ is the total number of sentiments. n_d^s is the number of sentiment words in comment d and n_d^o is the number of topic words in comment d. There are K topic models $\varphi_{z=1,\ldots,K}^o$ which denotes the multinomial distribution of words specific to topic z. For each topic z, there are S_1 topic-specific normal sentiment models $\varphi_{l=1,\ldots,S_1,z}^n$, which denotes the multinomial distribution of words specific to normal sentiment label l and topic z. There are S_2 special sentiment models $\varphi_{m=1,\ldots,S_2}^s$, which is the multinomial distribution of words specific to special sentiment label m. The variable θ denotes the distribution of topics in comment d, the variable π denotes the distribution of sentiments in comment d. Let d' be the comment that d interacts with, then the variables θ' and π' denote the distribution of topics and sentiments in comment d'.

In particular, we define an evolutionary matrix of topic z and sentiment label l, $E_{l,z}^t$, where each column is the word distribution of topic z and sentiment label l, $\sigma_{l,z,s}^{t}$, generated for comments received within the time slice specified by s. We then attach a vector of weights $\mu_{l,z}^{t} = \{\mu_{l,z,0}^{t}, \mu_{l,z,1}^{t}, \mu_{l,z,2}^{t}, \dots, \mu_{l,z,s}^{t}\}$, each of which determines the contribution of time slice s in computing the priors of $\beta_{l,z}^{t}$.

The Key Task of Dynamic Topic-based Sentiment Analysis (DTSA) is to estimate the model parameters σ^{t}, μ^{t}, θ^{t}, π, φ^{o}, φ^{n} and φ^{s} using a stochastic EM algorithm, then to extract topics and topic-specific sentiments of the online news and analyze their evolution over time simultaneously. Table 3.1 summarizes the notations of frequently used variables.

Table 3.1: Notations used in this chapter.

Symbol	Description
t	The index of timestamp
K	Number of topics
D	Number of comments
S_1	Number of normal sentiments (positive and negative)
S_2	Number of special sentiments (approval and disapproval)
M	The total number of sentiments
N_d^s	Number of sentiment words in comment d
N_d^t	Number of topic words in comment d
γ	Symmetric prior for sentiment labels
α	The prior for the topic distribution
β	The prior for the word distribution conditioned on sentiment labels and topics
φ_z^o	The multinomial distribution of words specific to topic z
$\varphi_{l,z}^n$	The multinomial distribution of words specific to normal sentiment label l and topic z
φ_m^s	The multinomial distribution of words specific to special sentiment label m
λ	The word prior for sentiment polarity information
θ	The distribution of topics in comment d
π	The distribution of sentiments in comment d
$\theta^{'}$	The distribution of topics in the comment that d interacts with
$\pi^{'}$	The distribution of sentiments in the comment that d interacts with
$E_{l,z}^t$	Evolutionary matrix of sentiment label l and topic z at epoch t
$\mu_{l,z}^t$	Weight vector which determines the contribution of time slice s in computing the priors of $\beta_{l,z}^t$
$\sigma_{l,z,s}^t$	The multinomial word distribution of sentiment label l and topic z with time slice s at epoch t

3.2.2 Multiple Timescale Model

Following the previous work [13], we could account for the influence of the past at different timescales to the current epoch. For example, we set time slice s equivalent to 2^{S-1} epochs.

Hence, if $S = 3$, we would consider three previous sentiment-topic-word distributions where the first distribution is between epoch $t-4$ and $t-1$, the second distribution is between epoch $t-2$ and $t-1$, and the third one is at epoch $t-1$. This would allow taking into consideration of previous long and short timescale distributions. However, this model would take more time and memory spaces and effective algorithm needs to be performed in order to reduce time/memory complexity.

Figure 3.2 illustrates the relationship among μ, E and β when the number of historical time slices accounted for is set to 3. Here, $\sigma^t_{l,z,s}$, $s \in \{1,2,3\}$ is the historical word distribution of topic z and sentiment label l within the time slice specified by s. As a form of smoothing to avoid the zero probability problem for unseen words, we set $\sigma^t_{l,z,0}$ for the current epoch as the uniform distribution where each element takes the value of $1/(vocabulary\ size)$. The evolutionary matrix $E^t_{l,z} = \{\sigma^t_{l,z,0}, \sigma^t_{l,z,1}, \sigma^t_{l,z,2}, \sigma^t_{l,z,3}\}$, and the weight matrix $\mu^t_{l,z} = \{\mu^t_{l,z,0}, \mu^t_{l,z,1}, \mu^t_{l,z,2}, \mu^t_{l,z,3}\}$. The Dirichlet prior for sentiment-topic-word distributions at epoch t is $\beta^t_{l,z} = \mu^t_{l,z} E^t_{l,z}$.

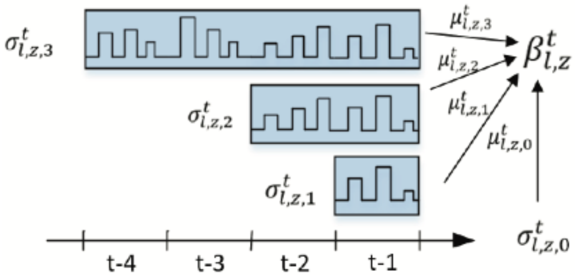


Figure 3.2: The relationship among μ, E and β.

3.2.3 Graphical Model

According to the real-world observation, we give two assumptions on sentiments as follow: (1) The sentiments of a comment do not exist independently, but depend on the comment it replies to and their relationship. (2) News comments can be divided into the reply comments and the original comments. The reply often omits the topic information, because it has the same topic with the original. We call this characteristic of user interaction “Topic Consistency”.

The graphical representation of DTSA is shown in Figure 3.3. The parameter definitions are shown in Table 3.1.

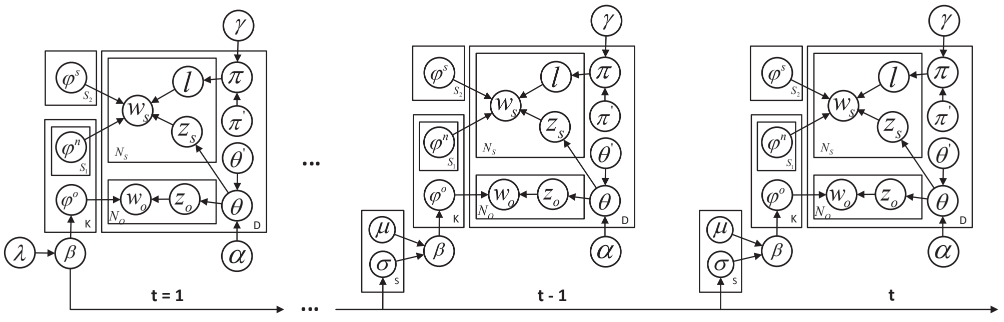


Figure 3.3: DTSA model.

Assuming we have already calculated the evolutionary parameters $\{E_{l,z}^{t}, \mu_{l,z}^{t}\}$ for the current epoch t, the formal generative process of DTSA model as shown in Figure 3.3 at epoch t is given as follows:

1. For each normal sentiment $l \in \{1, \dots, \mathrm{S}_1\}$:

   i. For each topic $z \in \{1, \dots, K\}$:

   Compute $\beta_{l,z}^{t} = \mu_{l,z}^{t} E_{l,z}^{t}$

2. For each topic $z \in \{1, \dots, K\}$:

   i. Choose a distribution $\varphi_z^o \sim Dir(\beta_z^o)$

   ii. For each normal sentiment $l \in \{1, \dots, S_1\}$:

   Choose a distribution $\varphi_{l,z}^n \sim Dir(\beta_{l,z}^n)$

3. For each special sentiment $m \in \{1, \dots, S_2\}$:

   Choose a distribution $\varphi_m^s \sim Dir(\beta_m^s)$

4. For each comment $d \in \{1, \dots, D\}$:

   i. Choose a distribution $\theta_{temp} \sim Dir(\alpha)$:

   Create a new distribution θ_d by combining θ_{temp} and $\theta_{d'}^{'}$

   ii. Choose a distribution $\pi_{temp} \sim Dir(\gamma)$:

   Create a new distribution π_d by combining π_{temp} and $\pi_{d'}^{'}$

   iii. For each topic word $w_{d,i}^o$ where $i \in \{1, \dots, n_d^o\}$:

   (a) Choose a topic $z_i^o \sim Mult(\theta_d)$

   (b) Choose a word $w_{d,i}^o$ from the distribution φ^o over words defined by the topic z_i^o .

   iv. For each sentiment word $w_{d,j}^s$ where $j \in \{1, \dots, n_d^s\}$:

   (a) Choose a topic $z_j^s \sim Mult(\theta_d)$

   (b) Choose a sentiment label $l_j \sim Mult(\pi_d)$

   © If l_j is a normal sentiment, choose a sentiment word $w_{d,j}^s$ from the distribution φ^n over words defined by the topic z_j^s and sentiment l_j. Otherwise, choose a special sentiment word $w_{d,j}^s$ from the distribution φ^s over words defined by the sentiment m_j.

In the proposed model, we divide the words into topic words and sentiment words. A sentiment lexicon and POS tagging are used to identify the sentiment words. There are two kinds of sentiments in the model-normal and special ones. The normal sentiments are topic-sensitive, where users use different words to express the same sentiment in different topics. However, the special sentiments are not topic-sensitive. According to [214], there are some patterns in approval and disapproval. Therefore, we choose the distributions of all K topics for each normal sentiment S^n, but only one distribution is chosen for each special sentiment S^s.

The topics and sentiments of the comment are affected by the comment a user interacts with, so we introduce the topics distribution θ' and sentiments distribution π' of the interacted comment to reflect this effect. Intuitively, we expect the two distributions θ and θ' are linear correlation, where $\theta = p\theta' + (1-p)\theta_{temp}$. The greater p value means a better topic consistency, which depends on the data set. We also expect $\pi = q\pi' + (1-q)\pi_{temp}$. Approximately, a larger q represents more weight on user interactions. The setting for p and q was determined empirically.

3.2.4 Online Inference

We use a stochastic EM algorithm to sequentially update the model parameters at each epoch using the newly arrived data and the parameters of the previously estimated model. At each EM iteration, we infer latent sentiment labels and topics using the collapsed Gibbs sampling and estimate the hyperparameters using maximum likelihood [215].

Model Parameters Estimation. The sampling formulas of model parameters θ^t, π^t, φ_t^o, φ_t^n and φ_t^s at epoch t given the evolutionary parameters E^t, μ^t are follows:

$$\theta_{d,k}^t = \frac{N_{d,k,t}^o + N_{d,k,t}^s + \alpha_k^t}{\sum_{k=1}^K (N_{d,k,t}^o + N_{d,k,t}^s + \alpha_k^t)} \tag{1}$$

$$\pi_{d,m}^t = \frac{N_{d,m,t}^s + \gamma_m^t}{\sum_{m=1}^M (N_{d,m,t}^s + \gamma_m^t)} \tag{2}$$

$$\varphi_{k,v,t}^o = \frac{N_{k,v,t}^o + \sum_S \mu_{k,s,v}^t \sigma_{k,s,v}^t}{\sum_{v=1}^V (N_{k,v,t}^o + \sum_S \mu_{k,s,v}^t \sigma_{k,s,v}^t)} \tag{3}$$

$$\varphi^{n}_{k,m,v,t} = \frac{N^{n}_{k,m,v,t} + \sum_{S} \mu^{t}_{k,m,s,v} \sigma^{t}_{k,m,s,v}}{\sum_{v=1}^{V} (N^{n}_{k,m,v,t} + \sum_{S} \mu^{t}_{k,m,s,v} \sigma^{t}_{k,m,s,v})} \tag{4}$$

$$\varphi^{s}_{m,v,t} = \frac{N^{s}_{m,v,t} + \sum_{S} \mu^{t}_{m,s,v} \sigma^{t}_{m,s,v}}{\sum_{v=1}^{V} (N^{s}_{m,v,t} + \sum_{S} \mu^{t}_{m,s,v} \sigma^{t}_{m,s,v})} \tag{5}$$

where $N^{o}_{d,k,t}$ is the number of topic words assigned to topic k in document d at epoch t. $N^{s}_{d,k,t}$ is the number of sentiment words assigned to topic k in document d at epoch t. $N^{s}_{d,m,t}$ is the number of sentiment words assigned to sentiment m in document d at epoch t. Other variables containing N are defined similarly.

Evolutionary Parameters Estimation. There are two sets of evolutionary parameters to be estimated, the weight parameters μ and the evolutionary matrix E. The update formulas are:

$$\left(\mu^{t}_{k,m,s}\right)^{new} \leftarrow \frac{\mu^{t}_{k,m,s} \sum_{v} \sigma^{t}_{k,m,s,v} A}{B} \tag{6}$$

where $A = \Psi\left(N^{t}_{k,m,v} + \sum_{s'} \mu^{t}_{k,m,s'} \sigma^{t}_{k,m,s',v}\right) - \Psi(\sum_{s'} \mu^{t}_{k,m,s'} \sigma^{t}_{k,m,s',v})$ and $B = \Psi\left(N^{t}_{k,m} + \sum_{s'} \mu^{t}_{k,m,s'}\right) - \Psi(\sum_{s'} \mu^{t}_{k,m,s'})$, $N^{t}_{k,m,v}$ is the number of times word v assigned to sentiment label m and topic k at epoch t, $N^{t}_{k,m} = \sum_{v} N^{t}_{k,m,v}$.

The evolutionary matrix E^{t} accounts for the historical word distributions at different time slices. The derivation of E^{t} therefore requires the estimation of each of its elements, $\sigma^{t}_{k,m,s,v}$, the word distribution in topic k and sentiment label m at time slice s, which can be calculated as follows:

$$\sigma^{t}_{k,m,s,v} = \frac{C^{t}_{k,m,s,v}}{\sum_{v} C^{t}_{k,m,s,v}} \tag{7}$$

where $C^{t}_{k,m,s,v}$ is the expected number of times word v is assigned to sentiment label m and topic k at time slice s. For the Multi-scale model, a time slice s might consist of several epochs. Therefore, $C^{t}_{k,m,s,v}$ is calculated by accumulating the count $N^{t'}_{k,m,v}$ over several epochs. The formula for computing $C^{t}_{k,m,s,v}$ is $C^{t}_{k,m,s,v} = \sum_{t'=t-2^{s-1}}^{t-1} N^{t'}_{k,m,v}$.

Distributed Model Training. To handle large scale data sets, we design a parallel training program for DTSA model on Hadoop, which is a Java-based open source distributed computing framework. Hadoop implemented the MapReduce framework proposed by Jeffrey et al. [216], and it can effectively handle a large amount of data. In Hadoop, all data are stored as key-value pairs. For our proposed model training program, the key is document id, and the value is the words and sentiments in the comment with their corresponding latent topics. The global model parameters include the Dirichlet prior α^t, γ^t, the weight parameter μ^t and the element of evolutionary matrix σ^t. Initially, a comment set is randomly split into N equal parts for N parallel executing processes. In the Map stage, every process loads the global model parameters from the last iteration, and uses them to sample the comments in its own part. The posterior distribution of hidden variables θ^t, π^t, φ_t^o, φ_t^n and φ_t^s are computed. In the Reduce stage, the posterior distribution θ^t, π^t, φ_t^o, φ_t^n and φ_t^s from all processes are aggregated to generate a new version of global model parameters.

3.3 Experimental Setup

We evaluate our proposed model on two kinds of datasets: news and twitter. For news datasets, we crawl the comments of four hot news events occurred from February 2014 to April 2014 using the Guardian Open Platform API[4]. (1) MH370 event: Malaysia airlines MH370 B777-200ER loses contact with air traffic control. (2) Crimea event: Russia dispatches troops to Crimea. (3) Sochi event: Sochi 2014 Winter Olympics are held successfully. (4) India event: India holds the largest president election ever. In order to evaluate our model's generality, we also crawl the tweets of Facebook events occurred on February 2014 from Twitter Search API[5]. Facebook event: Facebook buys WhatsApp for 19 Billion US Dollars. Each dataset contains the comments interacted with other comments by reply. Detail statistics of the datasets and sentiment distribution are shown in Table 3.2.

Table 3.2: Some statistics of the datasets and sentiment distribution.

Dataset	MH370	Crimea	Sochi	India	Facebook
Documents	351041	46722	405000	289900	101900
Tokens	2565490	382419	3518923	2359761	1026950
# of positive/negative documents	7/12	3/8	7/2	5/4	4/7

[4] http://open-platform.theguardian.com/

[5] http://apiwiki.twitter.com/

DTSA is an unsupervised model. As preprocessing, we first perform stemming and remove stopwords. Then we use Stanford POS Tagger[6] to tag the comments. In prior information, we use the sentiment lexicon SentiWordNet[7], containing 2290 positive and 4800 negative words with score over 0.6, as normal sentiment. Words contained in the sentiment lexicon are automatically labelled as sentiment words. For special sentiment, we use some seed words as prior information for approval, such as "praise", "agree", "support", and we use the discourse markers and swear words as prior information for disapproval, such as "what?", "nonsense" [214]. Other words, which are not labelled as normal/special sentiment words, are regarded as topic words. To quantitatively evaluate our model, we randomly select 500 comments from five datasets separately, and manually label each word as topic, normal and special sentiment word.

In our experiments, the unit epoch is set to daily. The number of topics K is set to be 20, the number of normal sentiments S_1 is set to be 2, the number of special sentiments S_2 is set to be 2. We set the Gibbs sampling iterations to be 5000. Following [217], we fix $\alpha = 50/K, \gamma = 50/(S_1 + S_2)$.

3.4 Experiments

In this section, we evaluate the performances of our proposed models with three experiments. In the first experiment, we show the topics and topic-specific sentiments extracted by DTSA with some qualitative analysis. The second experiment evaluates the computational time of our models. In the third experiment, we apply a document-level sentiment classification task to compare our models with several baselines.

3.4.1 Qualitative Results

In Table 3.3 we show the evolution of topics and topic-specific sentiments identified by the DTSA model with the number of time slices set to 4. Due to space limit, we only take an example of news comments on the MH370 event. For each topic, we list the top 5 topic words and the related sentiment words.

We can see that DTSA can extract topics and topic-based sentiments well. For example, the topic words are "MH370" and "disappeared" while the specific negative sentiment

[6] http://nlp.stanford.edu/software/tagger.shtml

[7] http://sentiwordnet.isti.cnr.it/

Table 3.3: News MH370 lose contact.

Time	Epoch 5	Epoch 6	Epoch 7	Epoch 8	Epoch 9	Epoch 10
Topic	MH370 Malaysia flight missing search	MH370 disappeared passenger plane terrorism	officials MH370 search scientists Australian	MH370 military sea evidence found	Boeing fuel MH370 died people	MH370 aircraft underwater Sumatra search
Senti_Positive	hopeful prospective extreme promising optimistic	optimistic hopeful cheerful confident huge	advanced optimistic sophisticated powerful support	support hopeful sustained sufficient powerful	hopeful promising likely confident high	prospective optimistic strong huge advanced
Senti_Negative	sad bad dangerous unbelievable tragic	cruel sorrily ruefully painful tearing	miserable painful tearing sorrily fierce	tragic unwilling hate cruel sorrowful	hopeless sad despairing suffering believe	hopeless ruefully painful sad misery

words are "painful" and "cruel". We also notice that the evolution of topics is well consistent with the actual news stories in real world. In addition, one improvement of the proposed model is that DTSA could automatically adjust the polarity of sentiment words. For example, in Epoch 9, the word "believe" becomes negative while it is positive in lexicon. In the comment "So what? We just believe they are alive!", "believe" should have labeled this comment positive, but the prior information "what?" makes this comment labeled as disapproval. And because this comment is a reply to a comment which approves of the news topic, we change the sentiment distribution of this comments to disapprove of the topic, which makes "believe" becomes negative words.

In Figure 3.4, we plot and compare the topic life cycle and its sentiment dynamics on MH370 event, where the strength distribution of a sentiment l in document d associated with the topic z, over the comment set C^t in each epoch t is calculated as:

$$\mathrm{P}(z,l) = \frac{1}{|C^t|}\sum_{d\in C^t} P(z|l,d)P(l|d) \qquad (8)$$

From Figure 3.4, we can see that in the first 2 days, the neutral sentiment dominates the opinions, for everyone talks about the facts during that time. However, the positive sentiment rises obviously over the next 2 days, reaching the peak at day 4, since the search and rescue operations. After that, the negative sentiment shoots up for 24 h, peaking at day 5. This is mainly because the Boeing 777 has run out of fuel and passengers have little chance of survival. All these results show that DTSA is effective to extract topics and topic-specific sentiments.

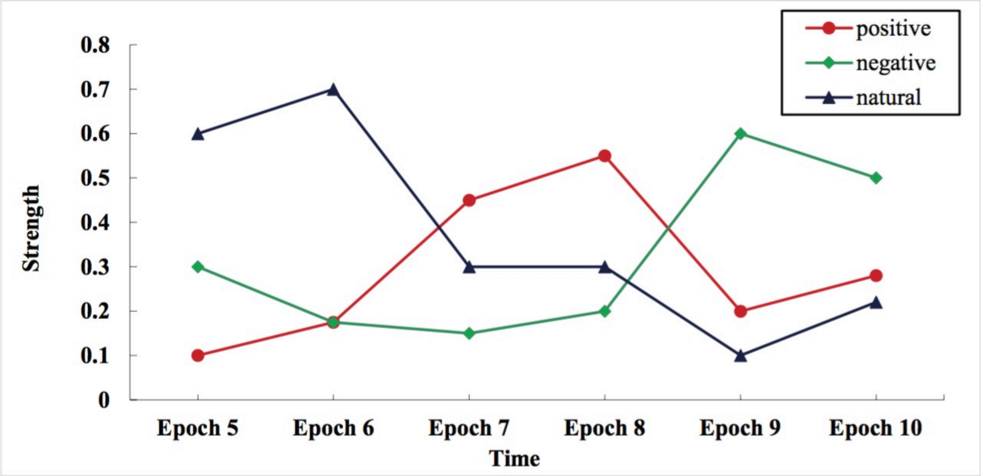


Figure 3.4: Sentiment dynamics of MH370 event.

3.4.2 Evaluation of Computational Time

In order to evaluate the effectiveness of DTSA in modelling dynamics, we compare the computational time of the DTSA model with the non-dynamic version of LDA [94] and JST [215], namely, LDA-one, JST-one, and JST-all. LDA-one and JST-one only use the training data in the current epoch whereas JST-all uses all the past data for model learning.

According to the previous work [101, 218], we also compare our proposed model with the other two different ways of setting the history influence on the generation of documents at current epoch: sliding-DTSA and skip-DTSA.
– sliding-DTSA: the current sentiment-topic-word distributions are dependent on the previous sentiment-topic specific word distributions in the last S epochs.
– skip-DTSA: we take history sentiment-topic-word distributions into account by skipping some epochs in between. For example, if S = 3, we only consider previous sentiment-topic-word distributions at epoch $t - 2^2$, $t - 2^1$, and $t - 2^0$.

Figure 3.5 shows the average training time per epoch with the increasing number of time slices. Sliding-DTSA, skip-DTSA and DTSA have similar average training time across the number of time slices. JST-one has less training time than the DTSA models. LDA-one uses least training time since it only models 3 sentiment topics while others all model a total of 20 sentiment topics. JST-all takes much more time than all the other models as it needs to use all the previous data for training.

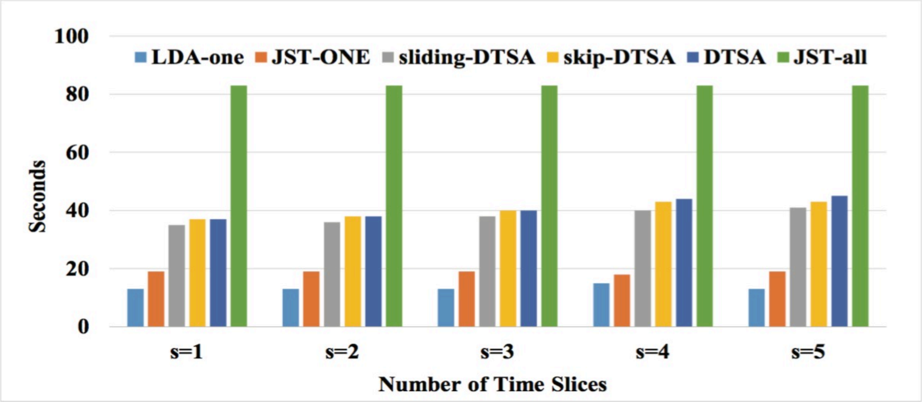


Figure 3.5: Computational time per epoch with different number of time slices.

3.4.3 Sentiment Classification

In this section, we present the results of sentiment classification with the number of time slices fixed at $S = 4$. We use the above mentioned datasets (see Table 3.2) to do the experiment. DTSA is a probabilistic model, we run 10 times for each experiment, and list the average F1-score in Table 3.4.

Table 3.4: The F1-score of sentiment classification results.

Dataset	DTSA	sliding-DTSA	skip-DTSA	JST-one	JST-one+	JST-all	JST-all+
MH370/Topic	**0.865**	0.811	0.859	0.674	0.683	0.786	0.790
MH370/Senti	**0.831**	0.792	0.803	0.613	0.679	0.715	0.767
Crimea/Topic	**0.837**	0.793	0.828	0.607	0.615	0.776	0.782
Crimea/Senti	**0.812**	0.733	0.769	0.579	0.631	0.727	0.731
Sochi/Topic	**0.896**	0.795	0.869	0.683	0.690	0.765	0.765
Sochi/Senti	**0.853**	0.763	0.783	0.602	0.668	0.734	0.760
India/Topic	**0.879**	0.815	0.853	0.657	0.662	0.790	0.803
India/Senti	**0.848**	0.778	0.793	0.613	0.659	0.716	0.765
Facebook/Topic	**0.857**	0.806	0.848	0.637	0.637	0.753	0.762
Facebook/Senti	**0.828**	0.749	0.776	0.591	0.629	0.687	0.733

We compare the performance of our model with JST-one and JST-all [213]. In order to prove the importance of user interactions, we introduce two special sentiments to JST-one and JST-all, making the new model called JST-one+ and JST-all+ which could identify approval and disapproval. For evaluating the advantage of using multiple timescale model, we also compare the DTSA model with sliding-DTSA and skip-DTSA.

As can be seen from Table 3.4, the performance of DTSA, sliding-DTSA and skip-DTSA are better than JST-one and JST-one+ method on all data sets. This is because JST-one and JST-one+ only use the data in the previous epoch for training and do not model dynamics. While our models take into account the influence of history sentiment-topic-word distributions, which can improve the sentiment classification metrics. Compared to sliding-DTSA and skip-DTSA, our model DTSA achieve the highest F1-score, which proves the effective of multiple timescale model.

In addition, we can see that JST-one+ and JST-all+ significantly improve the accuracy of sentiment classification on all data sets. This suggests that the special sentiments have a significant impact to the sentiment classification result. Furthermore, the DTSA outperforms the JST-all and JST-all+ methods on all data sets. JST-all+ could detect the user interactions, but does not use the user interactions to adjust the topic and sentiment distribution of comments, making them can not avoid the error caused by user interaction on both topic and sentiment.

We also analyze the influence of the topic number settings on the DTSA model performance. With the number of time slices fixed at $S = 4$, we vary the topic number $T \in \{1, 5, 10, 15, 20, 25\}$. Figure 3.6 shows the average sentiment classification accuracy over epochs with different number of topics. As can be seen from Figure 3.6, increasing the number of topics leads to a slight drop in accuracy. This trend is more evident on the twitter data set.

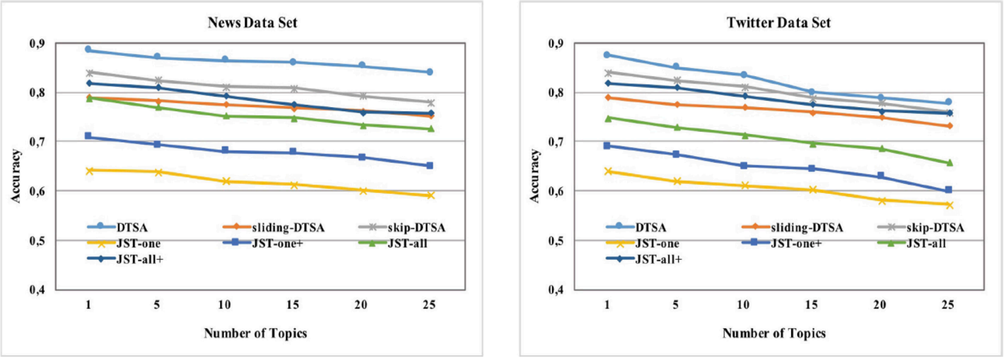


Figure 3.6: Sentiment classification accuracy with different number of topics.

3.5 Conclusion

In this chapter, a novel dynamic topic-based sentiment analysis model (DTSA) is proposed to extract topics and topic-specific sentiments from online news stories and comments. It could be used to decrease the error caused by user interactions, handle long-term and short-term dependency and automatically adjust model parameters in real time to improve the accuracy of classification based on sentiment recognition. The model is deployed on distributed online systems thus making improvements of efficiency of data process. The model has been tested on two kinds of data sets and displays promising results.

Chapter 4

Multilingual Review-Aware Recommendation via Aspect-based Sentiment Analysis

Textual reviews, which contain fine-grained user's opinions on different product features, have been regarded as valuable information sources to enhance the performance of Recommender Systems (RSs). However, with the dramatic expansion of the international market, consumers write reviews in different languages, which poses a new challenge for RSs dealing with this increasing amount of multilingual information. Recent studies that leverage deep learning techniques for review-aware RSs have demonstrated their effectiveness in modelling fine-grained user-item interactions through the aspects of reviews. However, most of these models can neither take full advantage of the contextual information from multilingual reviews nor discriminate the inherent ambiguity of words originated from the user's different tendency in writing. To this end, we propose a novel Multilingual Review-aware Deep Recommendation Model (MrRec) for rating prediction tasks. MrRec mainly consists of two parts: 1) Multilingual aspect-based sentiment analysis module (MABSA) which aims to jointly extract aligned aspects and their associated sentiments in different languages simultaneously with only requiring overall review ratings. 2) Multilingual recommendation module that learns aspect importances of both the user and item with considering different contributions of multiple languages, and estimates aspect utility via a dual interactive attention mechanism integrated with aspect-specific sentiments from MABSA. Finally, overall ratings can be inferred by a prediction layer adopting the aspect utility value and aspect importance as inputs. Extensive experiments on nine benchmark datasets from Amazon and Goodreads.com demonstrate the superior performance and interpretability of our model.

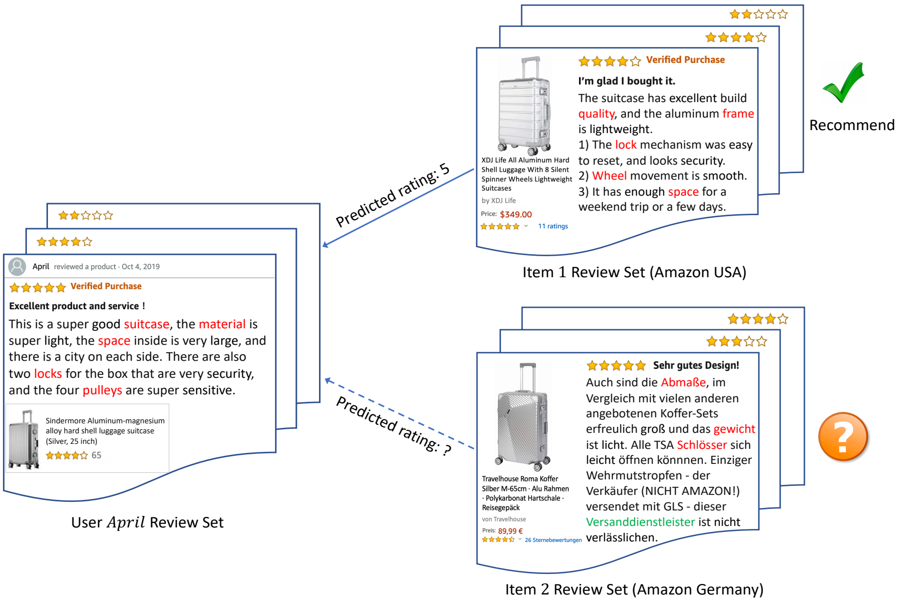
4.1 Introduction

Many e-commerce and social networking websites, such as Amazon and Goodreads, allow users to naturally write reviews along with a numerical rating to express opinions and share experiences towards their purchased items. These reviews are usually in the form of free text and play the role of carriers that reveal the reasons why users like or dislike the items or services they concerned. For example, a review may include the user's opinions on the various aspects of an item (e.g. its price, performance, quality, etc.), which are of high reference values for other users to make purchasing decisions. Therefore, in recent years, many recommender systems (RSs) [143, 219, 220, 221] have been developed by exploiting the semantic information covered in reviews to model a fine-grained user preference and alleviate the data sparsity problem for enhancing personalized recommendations.

Previous works on review-aware RSs are mainly devoted to the monolingual scenario. However, with the growth of the Web and the expansion of the international market, consumers write reviews in different languages, and online information is becoming more and more multilingual. Only addressing monolingual reviews lead to missing a lot of useful information existing in other languages. Indeed, it has been estimated that more than half of the world's population is bilingual, and nearly 45% of the websites provide content in a language different from English [222]. Besides, statistics of Amazon European market[8] show that almost 63% of users on average are non-English speakers, and Amazon provides services with different languages apart from English according to the users' geolocation. Facing the abundance of multilingual information, RSs need to evolve to effectively deal with the challenge of recommending interesting items with their review languages different from that the users adopted to express their preferences. As far as we know, this problem is very prevalent for most social media and e-commerce platforms (e.g. Amazon and Booking) but has never been explored before.

To have a deep insight into the problem of multilingual review-based recommendation, Figure 4.1 illustrates two different simplified recommendation scenarios the users often encounter when shopping on Amazon. *April* is an American user who usually buys suitcase on Amazon. When she is shopping at home in America, traditional review-based RSs could easily suggest item1 to *April* since the item features contained in its reviews match well

[8] https://orangeklik.com/optimize-listings-amazon-europe/


Predicted rating: 5
Predicted rating: ?
Recommend
User *April* Review Set
Item 1 Review Set (Amazon USA)
Item 2 Review Set (Amazon Germany)
April reviewed a product · Oct 4, 2019
Verified Purchase
Sindermore Aluminum-magnesium alloy hard shell luggage suitcase (Silver, 25 inch)
65
XDJ Life All Aluminum Hard Shell Luggage With 8 Silent Spinner Wheels Lightweight Suitcases
by XDJ Life
Price: $349.00
11 ratings
Travelhouse Roma Koffer Silber M-65cm · Alu Rahmen · Polykarbonat Hartschale · Reisegepäck
von Travelhouse
Preis: 89,99 €
26 Sternebewertungen

Excellent product and service !
This is a super good suitcase, the material is super light, the space inside is very large, and there is a city on each side. There are also two locks for the box that are very security, and the four pulleys are super sensitive.

I'm glad I bought it.
The suitcase has excellent build quality, and the aluminum frame is lightweight.
1) The lock mechanism was easy to reset, and looks security.
2) Wheel movement is smooth.
3) It has enough space for a weekend trip or a few days.

Sehr gutes Design!
Auch sind die Abmaße, im Vergleich mit vielen anderen angebotenen Koffer-Sets erfreulich groß und das gewicht ist licht. Alle TSA Schlösser sich leicht öffnen könnnen. Einziger Wehrmutstropfen - der Verkäufer (NICHT AMAZON!) versendet mit GLS - dieser Versanddienstleister ist nicht verlässlichen.

Figure 4.1: A toy example to show multilingual scenarios for RSs. Note that the red words represent aspects with positive sentiment and the green words represent aspects with negative sentiment.

with the user preference on different aspects expressed in her reviews. However, when she is travelling or studying abroad in Germany, it would be difficult for such RSs to provide a satisfying recommendation (e.g. item2) only according to the English reviews in her purchased history because most reviews of item2 are written in German. Such scenarios can also be easily found on social networking websites like Foursquare and Goodreads. This clearly motivates the need for efficient and effective recommendation techniques that cross the boundaries of languages.

So far, there have been few studies on multilingual recommendation in the literature. Existing methods [223–225] attempt to build language-independent user/item profiles by leveraging the concepts contained in external knowledge sources, such as Wikipedia and MultiWordNet. However, they are not suitable for our task due to inability to model fine-grained user-item interactions. Recently, empowered by continuous real-valued vector representations and semantic composition over contextual information, deep learning based methods have demonstrated their effectiveness in modelling user's fine-grained preferences to specific item features through the aspects extracted from reviews. The attention mechanism is mainly adopted in these works to automatically learn the aspect importance/weights for different user-item pairs. Guan et al. [220] propose an attentive aspect-based recommendation model which effectively captures the interactions between aspects extracted from reviews for rating perdition tasks. Chin et al. [221] propose to use a neural architecture incorporated with a co-attention mechanism to perform aspect-based representation learning for both users and items and estimate aspect-level importance in an end-to-end fashion.

Despite their state-of-the-art performance, they still suffer from the following limitations: (1) Most methods fail to handle multilingual reviews embodied with significant contextual information, especially when only a few reviews are provided in the monolingual scenario. (2) The users tend to exhibit different criteria when writing reviews, which leads to inherent ambiguity among words, and thus it is difficult for such approaches to precisely capture the user's intent. (3) Most existing methods neglect long-tail items when performing recommendations, which are crucial to gain the diversity of RSs and thereby improve the users' satisfaction. (4) The majority of above-mentioned algorithms take as inputs the concatenation of all words representations from every associated reviews, which makes the size of inputs considerably large, and therefore are impractical in the real-world applications.

In this chapter, to track the above limitations, we propose a novel Multilingual Review-aware Deep Recommendation Model (MrRec) which incorporates the aligned aspects and aspect-specific sentiments in different language reviews for rating prediction and interpretation. Specifically, MrRec consists of two parts: multilingual aspect-based sentiment analysis (MABSA) and multilingual recommendation module (MRM). In the first part, we utilize an unsupervised aspect-based autoencoder to learn a set of language-independent aspect embeddings. Then Multiple Instance Learning (MIL) framework integrated with hierarchical attention mechanism is designed to predict the aspect-specific sentiment distributions of review sentences, and learn aspect-aware sentence representations guided by the overall ratings. Note that the overall ratings serve both as a proxy of sentiment labels of reviews and as a bridge among languages. In the second part, a multilingual recommendation module is developed to infer the overall rating through a prediction layer with its input of the aspect utilities estimated by a dual interactive attention mechanism, and the corresponding aspect importances of both the user and item considering the different contributions of multiple languages. We applied our model to several real-world datasets and experimental results demonstrate the promising and reasonable performance of our approach.

In summary, our contributions are as follows:

– To the best of our knowledge, this is the first study that leverages multilingual reviews as potential resources to improve the interpretability and diversity of recommendation tasks. We also explore the possibility that deep learning techniques can be adopted to model language-independent user/item profiles in a fine-grained scale.

– We are the first to introduce MIL framework for multilingual aspect-based sentiment analysis which uses freely available multilingual word embeddings and only requires light supervision (user-provided ratings). It is demonstrated that the overall ratings can serve as the surrogate sentiment labels and bridges to address language barriers.

– We design a novel dual interactive attention mechanism that considers both popular and long-tail items for effectively modelling the fine-grained user-item interactions, as well as balancing between recommendation accuracy and diversity.

– Extensive experiments are conducted on nine datasets from Amazon and Goodreads to verify the effectiveness and efficiency of our model. The results show that MrRec not only outperforms state-of-the-art baselines but also interprets the recommendation results in great detail.

The remainder of the chapter is organized as follows. Section 4.2 introduces the related work. In section 4.3, we present our MrRec model in detail. We describe the data sets, experimental settings and the state-of-the-art methods we use in section 4.4. Section 4.5 shows our experiment results and analysis. Finally, we present the conclusions and future work in Section 4.6.

4.2 Relation to Other Work

Though there have been some studies on multilingual recommendation domain, this topic is still not fully investigated in the literature.

Traditional collaborative filtering is inherently multilingual since it does not rely on content information of items but solely on the user’s rating patterns. However, it encounters cold start issues when there is a rapid turnover of the recommended items. The work of [226] required users trust that is not always easy to obtain, as crucial information to overcome the gap between multiple languages. In [227], the authors proposed an LDA-based cross-lingual keyword recommendation method which can model both English and Japanese simultaneously. However, the problems lie in its inability to process more than two languages simultaneously and provide fine-grained recommendations. Some research works exploited well-known thesauri such as MultiWordNet [223, 224] and Wikipedia [225] to build language-independent user/item profiles for recommendation tasks. Narducci et al. [222] built concept-based representation of items by exploiting two knowledge sources, namely Wikipedia and BabelNet, in the multilingual recommendation. These works mainly rely on the use of ontologies and large corpora like Wikipedia, which are the key factors to determine the recommendation performance. However, they fail to consider fine-grained user preferences and sentiment information.

Specifically, in this paper, we present a novel approach for multilingual recommendations that can provide fine-grained user and item modelling based on the multilingual aspect extraction and aspect-specific sentiment analysis. The vocabularies in different languages are embedded into the same space such that synonyms and similar words project closely. Meanwhile, the contributions of multiple languages to specific user/item are learned through a neural attention mechanism.

4.3 The Proposed Model

In this section, we elaborate the proposed Multilingual Review-aware Deep Recommendation Model (MrRec) which aims to predict overall ratings based on captured multilingual user-item interactions in a fine-grained scale integrated with aspects and aspect-specific sentiments. First, we present the problem setting followed by the overview of our MrRec model. Then, we describe in detail the multilingual aspect-based sentiment analysis and the multilingual recommendation module for overall rating predictions.

4.3.1 Problem Setting

Considering a set of ratings $\mathcal{R}$ accompanied by a set of reviews $\mathcal{D}$, for item set $\mathcal{I}$ and user set $\mathcal{U}$, each user-item interaction can be represented as a tuple $(u, i,\ r_{u,i}, d_{u,i}\ , l_{u,i}\ )$ where $r_{u,i}$ is a numerical rating that can be seen as the overall sentiment the user u towards the item i, $d_{u,i}$ denotes the review text written by the user u on different aspects $a\ \in\ \mathcal{A}$ towards item i, and $l_{u,i}\ \in\ \mathcal{L}$ is the language used by u on i. In this paper, we only consider the cases that all the items are from the same category, and we assume that these items share the same set of K aspects $\mathcal{A}$. The primary goal is to predict the unknown ratings of items that the users have not reviewed yet. Before introducing our method, we would like to clarify the necessary concepts being used in our paper.

- **Overall rating**: An overall rating rated by user u on item i denoted as $r_{u,i}$ is a integer ranging from 1 to 5 stars. In our paper, we set $r_{u,i}$ as a real value within $[1, 5]$ for easy computation.
- **Aspect**: It is a high-level semantic concept denoting the attribute of items the users commented on in reviews. An aspect set $\mathcal{A} = \{a_1, \dots, a_K\}$ includes K aspects like price, screen, battery and performance for the mobile phone domain.
- **Aspect utility**: It is denoted as $y_{u,i}^{a_k} \in [-1,1]$ representing the user u's satisfaction with aspect a_k of a given item i. Aspect utility can be derived by aspect sentiment polarities with -1 being the most dissatisfied and 1 being the most satisfied with aspect ak .
- **Aspect importance**: For user u on item i, the aspect importance is represented by a K dimensional vector $\boldsymbol{\delta_u} = (\delta_{u,1}, \dots, \delta_{u,K})$, where the j -th dimension $\delta_{u,j} \in [0, 1]$ indicates the importance degree of aspect a_j of u with respect to i. Similarly, for item i on user u, the aspect importance vector is $\boldsymbol{\delta_i} = (\delta_{i,1}, \dots, \delta_{i,K})$, and $\delta_{i,k}$ indicates the importance degree of aspect a_k of i with respect to u.

4.3.2 Overview of MrRec Architecture

Figure 4.2 shows the overall architecture of our model which consists of two components responsible for aspects extraction as well as aspect-specific sentiment analysis, and overall rating prediction. Specifically, we feed the review set $\mathcal{D}$, its corresponding ratings $\mathcal{R}$ and languages $\mathcal{L}$ as the inputs to the MABSA module. Note that all inputs are from training split rather than validation or testing split. The training reviews are firstly transformed into a matrix $\mathcal{D} \in \mathbb{R}^{n \times d}$ via a multilingual embedding layer, which maps each word from the language vocabulary $\mathcal{V}$ to its corresponding d-dimensional vector initialized with pre-trained multilingual word embeddings for better semantic representations of user/item documents. n is the number of words in the reviews. Then the embedding matrix $\boldsymbol{D}$ will be used to derive a set of language-independent aspect embedding matrix $\mathcal{A} \in \mathbb{R}^{K \times d}$ through multilingual aspect extraction component. After that, aspect-based sentiment prediction part will take $\mathcal{A}$ as input and generates aspect sentiment distribution over C classes $\boldsymbol{p}_{s,a_k}^{sen} = \left(p_{sen,s,a_k}^{(1)}, \dots, p_{sen,s,a_k}^{(C)}\right), 1 \leq k \leq K$, and aspect-specific sentence representations $\boldsymbol{z}_{s,a_k}$, $1 \leq k \leq K$.

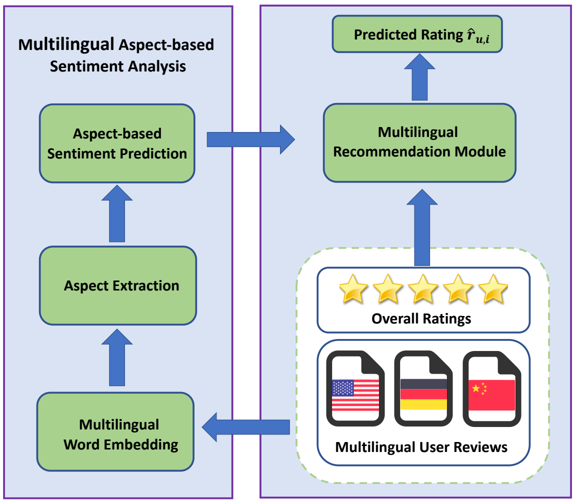

Multilingual Aspect-based Sentiment Analysis
Aspect-based Sentiment Prediction
Aspect Extraction
Multilingual Word Embedding
Predicted Rating $\hat{r}_{u,i}$
Multilingual Recommendation Module
Overall Ratings
Multilingual User Reviews

Figure 4.2: The proposed MrRec framework for rating prediction tasks.

In the second component, the inputs are document representations and document-level sentiment distributions of different aspects achieved through a weighted sum of the outputs from MABSA. Then the document representation set $\mathcal{F} = \{\boldsymbol{F}_{a_k}^{l} | 1 \le k \le K, 1 \le l \le L\}$ and document-level sentiment distribution set $\mathcal{P} = \{\boldsymbol{p}_{d,a_k}^{sen} | 1 \le k \le K, d \in \mathcal{D}\}$ are fed into MRM along with $\mathcal{R}$. $\boldsymbol{F}_{a_k}^{l} = (\boldsymbol{f}_{1,a_k}^{l}, \ldots, \boldsymbol{f}_{M_l,a_k}^{l})$ where M_l is the total number of reviews in language l, $\boldsymbol{f}_{m,a_k}^{l}$ is the realvalue vector of document representation. The output of MRM is the predicted rating $\hat{r}_{u,i}$ of user u on item i.

4.3.3 Multilingual Aspect-based Sentiment Analysis Module

The architecture of MABSA module is depicted in Figure 4.3. The module is basically composed of three parts: (a) multilingual word embedding layer, (b) aspect extraction and (c) aspect-based sentiment prediction.

Multilingual Word Embedding. For a given review $d_{u,i} \in D$, suppose there are N_s sentences in $d_{u,i}$, and the j-th sentence is composed by a sequence of words $\{w_{j1}, \ldots, w_{jN_w}\}$, where N_w is the total number of words in the sentence. For each word, we first use the multilingual word embeddings[9] [228] to represent the word in the multilingual embedding vector space with its representation denoted as $\boldsymbol{e} \in \mathbb{R}^{d_e}$. We then adopt a

[9] https://fasttext.cc/docs/en/aligned-vectors.html

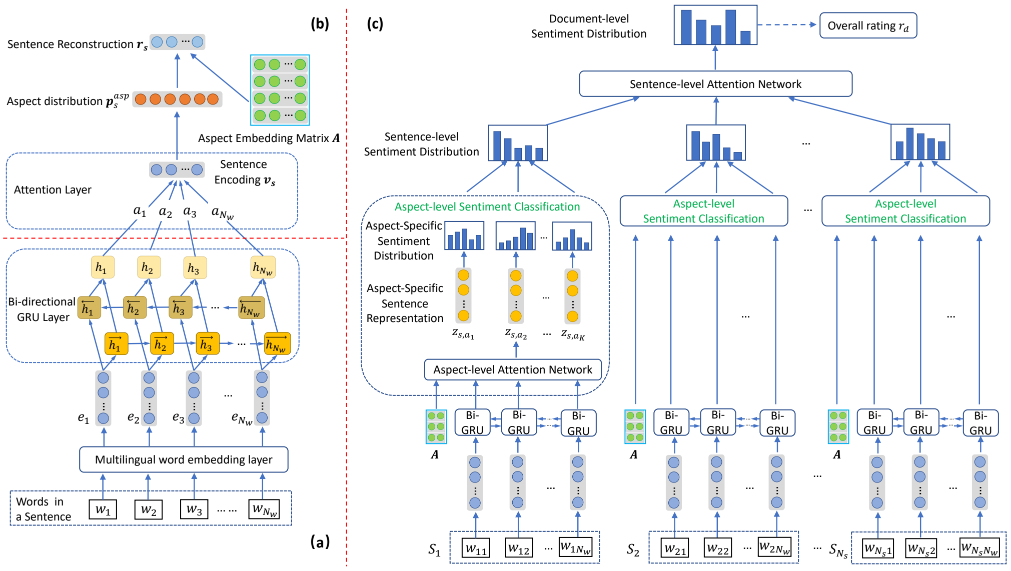

(b)
Sentence Reconstruction r_s
Aspect distribution p_s^{asp}
Aspect Embedding Matrix A
Sentence Encoding v_s
Attention Layer
a_1 a_2 a_3 a_{N_w}
h_1 h_2 h_3 h_{N_w}
Bi-directional GRU Layer
e_1 e_2 e_3 e_{N_w}
Multilingual word embedding layer
Words in a Sentence
w_1 w_2 w_3 w_{N_w}
(a)
(c)
Document-level Sentiment Distribution
Overall rating r_d
Sentence-level Attention Network
Sentence-level Sentiment Distribution
Aspect-level Sentiment Classification
Aspect-Specific Sentiment Distribution
Aspect-Specific Sentence Representation
z_{s,a_1} z_{s,a_2} z_{s,a_K}
Aspect-level Attention Network
Aspect-level Sentiment Classification
Aspect-level Sentiment Classification
A
Bi-GRU
S_1 w_{11} w_{12} w_{1N_w}
S_2 w_{21} w_{22} w_{2N_w}
S_{N_s} w_{N_s1} w_{N_s2} $w_{N_sN_w}$

Figure 4.3: Multilingual Aspect-based Sentiment Analysis Module.

bidirectional GRU [229] on $\boldsymbol{e}$ by summarizing information from both directions for word, and thus contextual information can be incorporated. Then the final word representation $\boldsymbol{h} \in \mathbb{R}^d$ can be derived through the concatenation of hidden states from both directions.

$$\boldsymbol{h} = [\overrightarrow{GRU}(\boldsymbol{e}); \overleftarrow{GRU}(\boldsymbol{e})] \tag{1}$$

Aspect Extraction. Our work builds on the basis of the research of [230], which is an analogous autoencoder called Attention-based Aspect Extraction (ABAE) model that learns aspect embedding matrix $A \in \mathbb{R}^{K\times d}$ with K aspects identified by each row by minimizing the reconstruction error.

Given the word embedding $[\boldsymbol{h}_1, \dots, \boldsymbol{h}_{N_w}]$ of sentence s, the sentence encoding $\boldsymbol{v}_s$ is computed as the weighted average of word embeddings using an attention encoder:

$$\boldsymbol{v}_s = \sum_{i=1}^{N_w} \mu_i \cdot \boldsymbol{h}_i \tag{2}$$

$$\mu_i = softmax\left(\boldsymbol{h}_i^T \cdot \boldsymbol{M}_a \cdot \boldsymbol{v}'_s\right) \tag{3}$$

where $\boldsymbol{v}'_s$ is simply the average of all word embeddings, μ_i is the attention weight on the i-th word, and $M_a \in \mathbb{R}^{d\times d}$ is an attention matrix that needs to be learned. The sentence embedding $\boldsymbol{v}_s$ is then fed into a softmax classifier to obtain a probability distribution over K aspects.

$$\boldsymbol{p}_s^{aps} = softmax(W_a \cdot \boldsymbol{v}_s + \boldsymbol{b}_a) \tag{4}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_a \in \mathbb{R}^d$ are weights and bias. $\boldsymbol{p}_s^{aps} = (p_{s,a_1}, \cdots, p_{s,a_K})$ is a K-dimensional vector with each element $p_{s,a_j}, j \in [1, K]$ representing the possibility that sentence s belongs to aspect a_j . The reconstruction of the sentence s is a linear combination of aspects $\boldsymbol{A}$:

$$\boldsymbol{r}_s = \boldsymbol{A}^T \cdot \boldsymbol{p}_s^{aps} \tag{5}$$

The model is trained by minimizing the reconstruction loss $L_r = \sum_{s\in\mathcal{D}} max\,(0, 1 - \boldsymbol{r}_s \cdot \boldsymbol{v}_s + \boldsymbol{r}_s \cdot \boldsymbol{v}_h) + \lambda\|\widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{A}}^T - \boldsymbol{I}\|$, where $\widetilde{\boldsymbol{A}}$ is $\boldsymbol{A}$ nomalized along each row, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{v}_h = argmin_{t\in\mathcal{V}_n} t \cdot \boldsymbol{v}_s$ represents the hardest one in a set of negative samples $\mathcal{V}_n$ in a minibatch.

Different from ABAE, we only focus on the hardest negative samples of different languages for computational efficiency [231]. When training on examples from different languages consecutively, it is difficult to learn a shared space that works well across languages. It is because only a subset of parameters are adjusted when training on each language, which may bias the model away to other languages. To avoid such issue, we follow the work of [232] and sample parallel sentences from different language pairs in a cyclic fashion at each training iteration. Specifically, during each iteration, the number of samples per language is equal to the mini-batch size divided by L. We randomly re-select samples to pad the vacancies for those languages which have fewer reviews.

Note that in Eq. 3, ABAE adopts word embedding $\boldsymbol{e}_i$ as input rather than $\boldsymbol{h}_i$, which makes the model originally a neural topic model. It is assumed that the sentence is composed with a bag of independent words, and thus the surrounding context among words are neglected when computing the global context of the sentence, $\boldsymbol{v}'_s$. By using the bidirectional GRU on each word embedding $\boldsymbol{e}_i$, we can summarize the information of the whole sentence centred around word w_i.

Aspect-based Sentiment Prediction. Given multilingual word embeddings $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{N_w})$ from Eq. 1, aspect matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K)$ and aspect distribution $\boldsymbol{p}_s^{asp}$ as inputs, for sentence s, aspect-based sentiment prediction module will output the document-level sentiment distribution $\boldsymbol{p}_d^{sen}$ on review d.

The idea of this module is based on Multiple Instance Learning (MIL) framework [233, 234] which deals with the problems where labels (document-level sentiment polarities in our case) are associated with groups of instances or bags (sentences), while instance labels are unseen. In our scenario, we assume that the sentiment distribution of document (overall rating) is composed as the weighted sum of the sentiments of each segment (sentence), which are the linear combinations of sentiment polarities of their associated aspects. To the best of our knowledge, we are the first that applies MIL framework to multilingual sentiment analysis.

The architecture of our module is shown in Figure 4.3(c). Particularly, we propose an aspect-level attention mechanism to fuse the information of aspects to the representations of target sentences.

$$\boldsymbol{r}_i' = \boldsymbol{W}_e \cdot [\boldsymbol{h}_i; \boldsymbol{a}_j] \tag{6}$$

$$\alpha_i = softmax(\boldsymbol{h}_c^T \cdot \tanh(\boldsymbol{W}_c \cdot [\boldsymbol{h}_i; \boldsymbol{r}_i'])) \tag{7}$$

where $\boldsymbol{r}_i' \in \mathbb{R}^d$ can be seen as aspect-based word embedding, and α_i represents the importance of the i-th word in sentence s. $\boldsymbol{W}_e \in \mathbb{R}^{d\times 2d}$ and $\boldsymbol{W}_c \in \mathbb{R}^{d_c\times 2d}$ are weight matrices. $\boldsymbol{h}_c \in \mathbb{R}^{d_c}$ is a learnable parameter. Then, the aspect-aware sentence representation can be achieved by weighted summation of all word embeddings in the sentence.

$$\boldsymbol{z}_{s,a_j} = \sum_{i=1}^{N_w} \alpha_i \cdot \boldsymbol{h}_i \tag{8}$$

The sentence representation $\boldsymbol{z}_{s,a_j}$ is fed into a softmax layer to predict the aspect-specific sentiment distribution on sentence s with respect to aspect a_j:

$$\boldsymbol{p}_{s,a_j}^{sen} = softmax(\boldsymbol{W}_s \cdot \boldsymbol{z}_{s,a_j} + \boldsymbol{b}_s) \tag{9}$$

where$\boldsymbol{W}_s$ and $\boldsymbol{b}_s$ are the parameters. $\boldsymbol{p}_{s,a_j}^{sen}$ is a real-valued vector $(p_{sen,s,a_j}^{(1)}, \cdots, p_{sen,s,a_j}^{(C)})$ with 1 and C representing the most negative and most positive polarity score respectively. For instance, supposing a 5-class scenario, C represents 5 classes and $p_{sen,s,a_j}^{(k)}, \mathrm{k} \in [1, \mathrm{C}]$ denotes the probability that the polarity score equals to k of sentence s with respect to aspect a_j. Thus the sentence-level sentiment distribution can be calculated as:

$$\boldsymbol{p}_s^{sen} = \sum_{j=1}^{K} p_{s,a_j} \cdot \boldsymbol{p}_{s,a_j}^{sen} \tag{10}$$

Each element $p_{sen,s}^{(k)}, \mathrm{k} \in [1, \mathrm{C}]$ of $\boldsymbol{p}_s^{sen}$ represents the probability that the polarity score is equal to k of sentence s. After that, the sentence representation on all aspects can be achieved by:

$$\boldsymbol{z}_s = \sum_{j=1}^{K} p_{s,a_j} \cdot \boldsymbol{z}_{s,a_j} \tag{11}$$

Similarly, to capture the context around the target sentence s, we feed $\boldsymbol{z}_s$ to the bi-directional GRU layer $\boldsymbol{h}_s = [\overrightarrow{GRU}(\boldsymbol{z}_s); \overleftarrow{GRU}(\boldsymbol{z}_s)]$. To learn different contributions of sentences in a review, we adopt a sentence-level attention network defined as follows:

$$\beta_s = softmax(\boldsymbol{h}_r^T \cdot \tanh(\boldsymbol{W}_r \cdot \boldsymbol{h}_s + \boldsymbol{b}_r)) \tag{12}$$

where $\boldsymbol{h}_r \in \mathbb{R}^{d_r}$, $\boldsymbol{W}_r \in \mathbb{R}^{d_r \times d}$ and $\boldsymbol{b}_r \in \mathbb{R}^{d_r}$ are learnable parameters. Finally, we obtain the document-level sentiment distribution as the weighted sum of sentence distributions:

$$\boldsymbol{p}_d = \sum_{s=1}^{N_s} \beta_s \cdot \boldsymbol{p}_s^{sen} \tag{13}$$

where N_s is the number of sentences in review d.

The aspect-based sentiment prediction is trained end-to-end on all training reviews guided by the overall ratings accompanied with reviews. We use the negative log-likelihood as the objective function:

$$\mathcal{L}_s = -\sum_{d \in \mathcal{D}} \log p_d^{(r_d)} \tag{14}$$

where $r_d \in [1, C]$ is the polarity score of review d.

4.3.4 Multilingual Recommendation Module

Given the review set $\mathcal{F}_u = \{\boldsymbol{F}_{u,a_k}^l | 1 \le k \le K, 1 \le l \le L\}$ written by user u and the review set $\mathcal{F}_i = \{\boldsymbol{F}_{i,a_j}^l | 1 \le j \le K, 1 \le l \le L\}$ written for item i, as input to multilingual recommendation module (MRM). $\boldsymbol{F}_{u\,/i,a_k}^l = (\boldsymbol{f}_{u/i,1,a_k}^l, \cdots, \boldsymbol{f}_{u/i,M_l,a_k}^l)$, where $\boldsymbol{f}_{u/i,m,a_k}^l \in \mathbb{R}^d$ denotes the document representation of the m-th review in language l on aspect a_k for user u or item i, and M_l is the total number of reviews in language l. To obtain it, we first learn sentence representation incorporated with contextual fusion using bi-directional GRU with input from Eq. 8: $\boldsymbol{h}_{s,a_k} = [\overrightarrow{GRU}(\boldsymbol{z}_{s,a_k}); \overleftarrow{GRU}(\boldsymbol{z}_{s,a_k})]$. Then the importance of sentence s on aspect a_k can be calculated as:

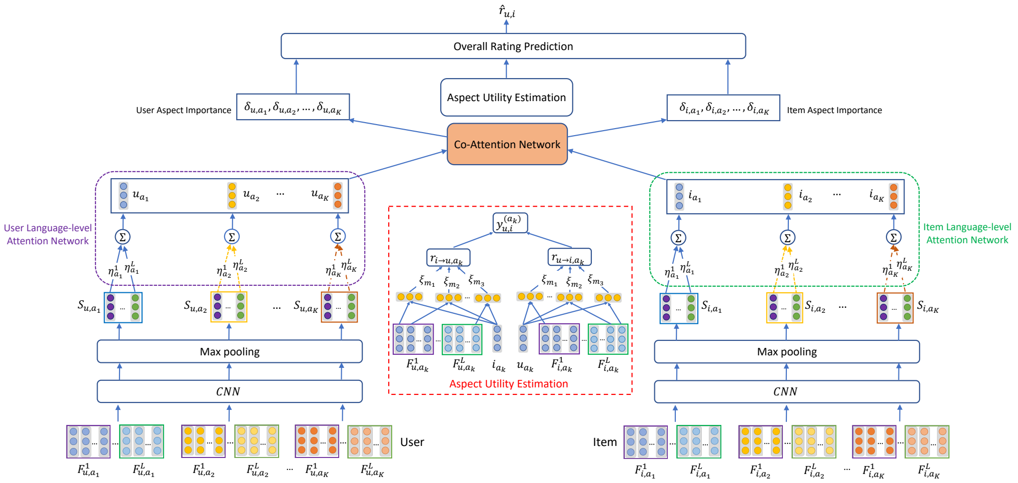

$\hat{r}_{u,i}$
Overall Rating Prediction
Aspect Utility Estimation
User Aspect Importance
$\delta_{u,a_1}, \delta_{u,a_2}, \dots, \delta_{u,a_K}$
$\delta_{i,a_1}, \delta_{i,a_2}, \dots, \delta_{i,a_K}$
Item Aspect Importance
Co-Attention Network
User Language-level Attention Network
u_{a_1}
u_{a_2}
u_{a_K}
Σ
$\eta^1_{a_1}$ $\eta^L_{a_1}$
$\eta^1_{a_2}$ $\eta^L_{a_2}$
$\eta^1_{a_K}$ $\eta^L_{a_K}$
S_{u,a_1}
S_{u,a_2}
S_{u,a_K}
Max pooling
CNN
F^1_{u,a_1} F^L_{u,a_1} F^1_{u,a_2} F^L_{u,a_2} F^1_{u,a_K} F^L_{u,a_K}
User
$y^{(a_k)}_{u,i}$
$r_{i\to u,a_k}$
$r_{u\to i,a_k}$
ξ_{m_1} ξ_{m_2} ξ_{m_3}
F^1_{u,a_k} F^L_{u,a_k} i_{a_k} u_{a_k} F^1_{i,a_k} F^L_{i,a_k}
Aspect Utility Estimation
Item Language-level Attention Network
i_{a_1}
i_{a_2}
i_{a_K}
S_{i,a_1}
S_{i,a_2}
S_{i,a_K}
Item
F^1_{i,a_1} F^L_{i,a_1} F^1_{i,a_2} F^L_{i,a_2} F^1_{i,a_K} F^L_{i,a_K}

Figure 4.4: Multilingual Recommendation Module.

$$\beta'_{s,a_k} = softmax(\boldsymbol{h}_t^T \cdot \tanh\,(\boldsymbol{W}_t \cdot \boldsymbol{h}_{s,a_k} + \boldsymbol{b}_t)) \tag{15}$$

where $\boldsymbol{h}_t \in \mathbb{R}^{d_t}$, $\boldsymbol{W}_t \in \mathbb{R}^{d_t \times d}$ and $\boldsymbol{b}_t \in \mathbb{R}^{d_t}$ are learnable parameters. The document representation can be achieved by the weighted sum of sentence representations. Likewise, document-level sentiment distribution on the aspect can also be derived through a weighted sum of aspect sentiment distributions:

$$\boldsymbol{f}^l_{u/i,M_l,a_k} = \sum_{s=1}^{N_s} \beta'_{s,a_k} \cdot \boldsymbol{h}_{s,a_k}, \quad \boldsymbol{p}^{sen}_{d,a_k} = \sum_{s=1}^{N_s} \beta'_{s,a_k} \cdot \boldsymbol{p}^{sen}_{s,a_k} \tag{16}$$

Since the modelling process for users and items are identical, we focus on illustrating the process for a given user.

The overall architecture of MRM is depicted in Figure 4.4. First, the user review set $\boldsymbol{F}^l_{u,a_k}$ is grouped by different languages and aspects, which is fed into MRM as input. To capture the semantic features of reviews, we employ a CNN network to perform convolution operations on each $\boldsymbol{F}^l_{u,a_k}$ matrix with N_f filters. Since we do not consider the orders of reviews for users and items, we set the window size to 1 to extract features from each review independently. Specifically, for review $\boldsymbol{f}^l_{u,j,a_k}$, we perform: $\hat{\boldsymbol{f}}^{l,t}_{u,j,a_k} = \sigma(\boldsymbol{W}_t * \boldsymbol{f}^l_{u,j,a_k} + b_t)$, where $*$ is the convolution operator, $\boldsymbol{W}_t$ is the t-th convolution filter, $b_t \in R$ is a bias term, and σ is a non-linear function i.e. ReLU. By applying the t-th filter on the $\boldsymbol{F}^l_{u,a_k}$ matrix, we obtain a feature map represented as $\hat{\boldsymbol{f}}^{l,t}_{u,a_k} = (\hat{f}^{l,t}_{u,1,a_k}, \dots, \hat{f}^{l,t}_{u,M_l,a_k})$. Then max-pooling is applied to find the most important feature on the subset of reviews $s_{u,a_k}^{l,t} = max(\hat{\boldsymbol{f}}_{u,a_k}^{l,t})$. After performing on all filters, we obtain the vector $\boldsymbol{s}_{u,a_k}^{l} = \left(s_{u,a_k}^{l,1}, \dots, s_{u,a_k}^{l,N_f}\right) \in \mathbb{R}^{N_f}$ which can be seen as the language-specific representation of user u on aspect a_k. The output from max-pooling layer that represent the same aspect a_k are concatenated to form a matrix $\boldsymbol{S}_{u,a_k} = (\boldsymbol{s}_{u,a_k}^{1}, \dots, \boldsymbol{s}_{u,a_k}^{L}) \in \mathbb{R}^{\mathrm{L} \times N_f}$.

Language-level Attention Network. We argue that not all languages are of equal importance to the user. For instance, if a user u's primary language is French and s/he also writes reviews in English, French should be more important than English in most cases. In other words, French contributes more than English in learning user representation. Note that when we refer to "primary language", we mean the language which is the most informative one for the user u. Therefore, inspired by the related research of self-attention network [235], we propose a language-level attention network.

Indicatively, a softmax layer is employed to determine the importance of different languages. In this case, the most informative languages are given larger weights, while the input of the softmax is a transformation of each language.

$$\eta_{a_k}^{l} = softmax(\boldsymbol{w}_l^T \cdot \boldsymbol{s}_{u,a_k}^{l}) \tag{17}$$

where $\boldsymbol{w}_l \in \mathbb{R}^{N_f}$ is a weight vector. Then a weighted combination of language-specific user representations on aspect a_k is considered as the representation of user u on aspect a_k:

$$\boldsymbol{u}_{a_k} = \sum_{l=1}^{L} \eta_{a_k}^{l} \cdot \boldsymbol{s}_{u,a_k}^{l} \tag{18}$$

The representation of user u on all aspects are denoted as $\boldsymbol{U}_u = (\boldsymbol{u}_{a_1}, \dots, \boldsymbol{u}_{a_k})$. Similarly, we learn language importance on item i's review set and obtain the item representation matrix denoted as $\boldsymbol{I}_i = (\boldsymbol{i}_{a1}, \cdots, \boldsymbol{i}_{ak})$.

Co-Attention Network. The self-attention mechanism focuses on the "static" features of users or items rather than the features of user-item interactions, and thus is suboptimal to learn the importance among aspects of user u taken specific item i into account, and vice versa. Therefore, following the work of [236], we propose to learn the aspect importance of user u or item i in a joint manner.

To incorporate item i as context when calculating the aspect importance of user u, we need to know how user u and item i matches on certain aspects:

$$\boldsymbol{E}_u = \sigma(\boldsymbol{U}_u \cdot \boldsymbol{W}_e \cdot \boldsymbol{I}_i^T) \tag{19}$$

where $\boldsymbol{W}_e \in \mathbb{R}^{N_f \times N_f}$ is a learnable parameter, and each entry of $\boldsymbol{E}_u \in \mathbb{R}^{K \times K}$ represents the similarity between the corresponding user and item pair representations on aspects. Next, the aspect-level importance of user u w.r.t. item i can be learned as:

$$\boldsymbol{H}_u = \sigma\big(\boldsymbol{U}_u \cdot \boldsymbol{W}_u + \boldsymbol{E}_u(\boldsymbol{I}_i \cdot \boldsymbol{W}_i)\big), \qquad \boldsymbol{\delta}_u = softmax(\boldsymbol{H}_u \cdot \boldsymbol{v}_u) \tag{20}$$

where $\boldsymbol{W}_u, \boldsymbol{W}_i \in \mathbb{R}^{N_f \times d_f}$, and $\boldsymbol{v}_u \in \mathbb{R}^{d_f}$ are learnable parameters. $\boldsymbol{\delta}_u = (\delta_{u,a_1}, \dots, \delta_{u,a_K})$ is a K-dimensional vector with each element representing the importance of the corresponding aspect for user u. Likewise, the aspect importance of item i can be derived as $\boldsymbol{\delta}_i = (\delta_{i,a_1}, \dots, \delta_{i,a_K})$.

Aspect Utility Estimation. When calculating user u's satisfaction with each aspect a_k of item i, for the improvement of recommendation diversity, we need to consider not only the utilities of his/her like-minded users on aspect a_k w.r.t item i, but also the user u's individual utilities assigned by user u to items that are similar to item i on aspect a_k even though the items are less popular (long-tail items). Hence, a dual interactive attention mechanism is designed to learn the aspect-level ratings of user u on item i and vice versa. Given the aspect-specific sentiment distribution on document d w.r.t aspect a_k, $\boldsymbol{p}_{d,a_k}^{sen} = (p_{sen,d,a_k}^{(1)}, \cdots, p_{sen,d,a_k}^{(C)})$, and aspect-level document representations $\{\boldsymbol{f}_{u/i,m,a_k}^{l} | 1 \leq m \leq M_{u/i}, 1 \leq k \leq K\}$, to estimate the aspect utility of user u on item i $r_{u \to i,a_k}$, and the aspect utility of item i w.r.t. user u $r_{i \to u,a_k}$, we first define a real-valued sentiment polarity vector $\boldsymbol{\omega} = (\omega^{(1)}, \cdots, \omega^{(C)})$ where $\omega^c \in [-1, 1]$ represents a weight assigned according to discrete uniform distribution so that $\omega^{(c+1)} - \omega^{(c)} = \frac{2}{C-1}$. For instance, the sentiment polarity vector of a 5-class scenario would be $\boldsymbol{\omega} = (-1, -0.5, 0, 0.5, 1)$. Thus, the document-level sentiment polarity on aspect a_k can be calculated as:

$$polarity(d)^{a_k} = \sum_{c \in [1,C]} p_{sen,d,a_k}^{(c)} \cdot \omega^{(c)} \tag{21}$$

Next, to find the like-minded users of user u, we define the element-wise product of user representation and document-level representation of item i w.r.t. aspect a_k.

$$\phi(u, i) = \boldsymbol{u}_{a_k} \odot (\boldsymbol{W}_f \cdot \boldsymbol{f}_{i,m_i,a_k}) \tag{22}$$

where $\boldsymbol{f}_{i,m_i,a_k} \in \mathcal{F}_i$ and $\boldsymbol{W}_f \in \mathbb{R}^{N_f \times d}$ is the projection matrix used to map document-level representations and user representation to the same space. The contribution of review m_i to user u can be learned by a softmax layer:

$$\xi_{m_i} = softmax\left(\boldsymbol{W}_{att}^T \cdot \phi(u,i)\right) \tag{23}$$

where $\boldsymbol{W}_{att} \in \mathbb{R}^{N_f}$ is a learnable parameter. Then we can obtain the aspect utility of user u to item i on aspect a_k:

$$r_{u \to i, a_k} = \sum_{m_i=1}^{|\mathcal{F}_i|} \xi_{m_i} \cdot polarity(d_{m_i})^{a_k} \tag{24}$$

Similarly, the aspect utility of item i w.r.t. user u can be calculated as: $r_{i \to u, a_k} = \sum_{m_u=1}^{|\mathcal{F}_u|} \xi_{m_u} \cdot polarity(d_{m_u})^{a_k}$, where $|\mathcal{F}_u|$ and $|\mathcal{F}_i|$ are total number of reviews in user u's set and item i's set. To learn user u's overall satisfaction with item i on the aspect a_k, a regression layer is stacked to the concatenation of these two aspect-level ratings:

$$y_{u,i}^{(a_k)} = W_y \cdot \begin{bmatrix} r_{u \to i, a_k} \\ r_{i \to u, a_k} \end{bmatrix} \tag{25}$$

Overall Rating Prediction. The overall rating for user-item pair can be predicted via a prediction layer with the combination of the user's satisfaction $y_{u,i}^{(a_k)}$ and the aspect importance $\delta_{u,a_k}, \delta_{i,a_k}$ as inputs:

$$\hat{r}_{u,i} = \sigma_C\left(\sum_{k=1}^{K} \delta_{u,a_k} \cdot \delta_{i,a_k} \cdot y_{u,i}^{(a_k)}\right) + b_u + b_i + b \tag{26}$$

where b_u , b_i and b are user, item and global bias. Function $\sigma_C(x) = 1 + \frac{C-1}{1+exp(-tan(\frac{\pi}{2}x))}$ is a variant of sigmoid function, producing the value within the range of $[1, C]$. Note that since $x \in [-1, 1]$ needs to be mapped to the range of $[1, C]$, we first map x to radian space which is then prolonged to $[-\frac{\pi}{2}, \frac{\pi}{2}]$. $tan(\cdot)$ function is adopted to project x to the range of $[-\infty, \infty]$. Finally, the variant of sigmoid function can be used to achieve the goal. The model parameters can be learned through backpropagation with the standard Mean Squared Error (MSE) as the loss function. The three parts of our model need to be learned separately. The performance of each part implicitly relies on the outputs from the previous component. Thus we adopt a pre-trained multilingual word embedding to improve the performance. To train the first two parts ((b) and (c) in Figure 4.3), we uniformly mix the training set with different languages. To deal with the overfitting problem existing in deep learning models, we adopt the dropout technique with parameter ρ , and $L2$ regularization term to the objective function.

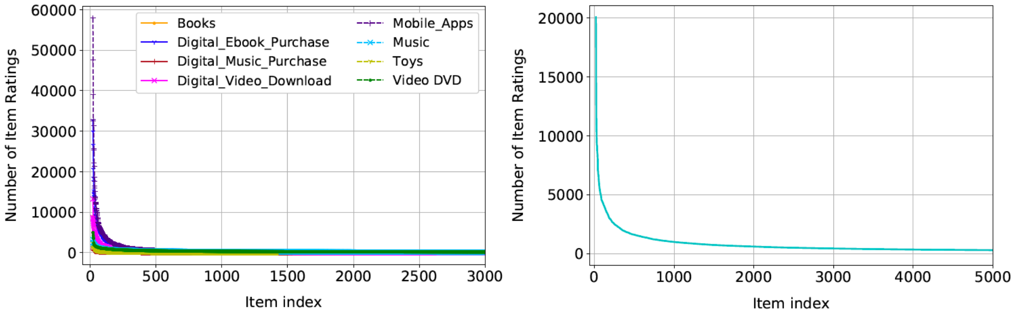
4.4 Experimental Settings

4.4.1 Datasets

We evaluate our proposed model on rating predictions against several state-of-theart baselines with real-world datasets freely available online. Specifically, we use nine datasets

Table 4.1: Statistics of the datasets for evaluating the recommendation task.

Datasets	*Books*	*Digital Ebook Purchase*	*Digital Music Purchase*	*Digital Video Download*	*Mobile Apps*	*Music*	*Toys*	*Video DVD*	*Goodreads*
# Users	847,499	1,118,718	125,381	758,052	1,161,439	753,598	96,819	1,038,981	440,817
# Items	26,642	5,392	16,310	18,674	1,327	28,540	1,408	37,365	1,901,485
# Interactions/Reviews	1,165,926	1,534,618	159,320	1,078,790	1,709,289	1,318,337	108,547	1,908,260	14,668,579
# Multilingual users	3,017	5,890	528	12,164	12,775	6,324	232	10,682	84,669
# Multilingual items	8,180	2,699	2,518	6,345	1,326	16,632	926	15,536	147,025
# Multilingual interactions	1,033,626	1,493,101	87,749	999,787	1,709,263	1,261,610	105,448	1,627,160	11,478,423
Avg. # words/ review	115.0 *(σ=184.3)*	49.7 *(σ=86.1)*	53.2 *(σ=97.6)*	34.6 *(σ=62.7)*	29.9 *(σ=35.0)*	114.3 *(σ=179.5)*	46.8 *(σ=76.5)*	101.7 *(σ=184.8)*	129.3 *(σ=181.1)*
Avg. # sentences/ review	7.066 *(σ=10.223)*	3.842 *(σ=5.091)*	4.107 *(σ=5.899)*	3.084 *(σ=3.723)*	2.741 *(σ=2.393)*	7.436 *(σ=11.240)*	3.604 *(σ=4.254)*	6.527 *(σ=9.903)*	8.415 *(σ=11.179)*
Avg. # reviews/ user	1.376 *(σ=3.810)*	1.372 *(σ=1.498)*	1.271 *(σ=1.133)*	1.423 *(σ=1.480)*	1.472 *(σ=1.638)*	1.749 *(σ=4.612)*	1.121 *(σ=0.553)*	1.837 *(σ=5.179)*	33.276 *(σ=114.7)*
Avg. # reviews/ item	43.76 *(σ=174.9)*	284.61 *(σ=1247.2)*	9.77 *(σ=31.5)*	57.77 *(σ=290.0)*	1288.09 *(σ=3702.8)*	46.19 *(σ=116.9)*	77.09 *(σ=144.2)*	51.07 *(σ=125.6)*	7.714 *(σ=70.4)*
Density	0.005%	0.025%	0.008%	0.008%	0.111%	0.006%	0.080%	0.005%	0.002%



(a) Amazon Dataset

(b) Goodreads Dataset

Figure 4.5: The popularity distribution of items in the experimental datasets.

from two sources: Amazon Customer Reviews[10] and Book Reviews[11]. The datasets cover 11 languages: Afrikaans (AF), English (EN), German (DE), French (FR), Catalan (CA), Spanish (ES), Italian (IT), Norwegian (NO), Romanian (RO), Slovenian (SL), Tagalog (TL). For Amazon Customer Reviews dataset, eight datasets from different domains are used (i.e. Books, Digital Ebook Purchase, Digital Music Purchase, Digital Video Download, Mobile Apps, Music, Toys and Video DVD). The other is from the Book Reviews dataset. Note that we determine not to apply the k-core settings [237] over these datasets, whereby there are at least k ratings/reviews for each user and item, as it trivializes the problem of data sparsity which is inevitable in real-world recommendations. The basic statistics are summarized in Table 4.1. Besides, we also plot the popularity distribution of item set on two dataset sources in Figure 4.5, from which we can see a substantial amount of long-tail items that need to be considered when providing recommendations.

Table 4.2: Statistics of the datasets for evaluating the aspect-based sentiment analysis task.

Datasets	**Trip-MAML**			**Restaurant Reviews**	
	EN	*ES*	*IT*	*EN*	*FR*
# Reviews	442	500	500	652	455
# Sentences	5799	2620	2593	3418	2427
# Opinions	5587	3416	3602	3742	3484
# Positive opinions	3344	2402	2484	2278	1605
# Negative opinions	1377	792	651	855	1646
# Neutral opinions	866	222	467	609	233

To evaluate the performance of our multilingual aspect-based sentiment prediction module, we adopt Trip-MAML[12] dataset, which consists of TripAdvisor hotel reviews in English, Italian and Spanish. Besides, we also produce a multilingual dataset which incorporates English and French reviews on restaurant domain to test our module. Specifically, we adopt English restaurant reviews[13] follow the work of [236], and French restaurant reviews[14] from [239], which are then combined to form a multilingual datasets denoted as Restaurant

[10] https://s3.amazonaws.com/amazon-reviews-pds/readme.html

[11] https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/reviews?authuser=0

[12] http://hlt.isti.cnr.it/trip-maml/

[13] http://dilab.korea.ac.kr/jmts/jmtsdataset.zip

[14] http://metashare.ilsp.gr:8080/repository/search/?q=semeval+2016

Reviews. The statistics of the datasets are presented in Table 4.2. For both datasets, each review comes with an overall rating on a discrete ordinal scale from 1 to 5 “stars”. The datasets are annotated at sentence-level with 3-values sentiment labels including Positive, Negative and Neutral/Mixed. Each sentence is manually annotated according to 12 recurrent aspects, i.e. *Rooms, Cleanliness, Value, Service, Location, Check-in, Business, Food, Building, Sleep Quality, Other* as well as *NotRelated*, and 7 recurrent aspects, i.e. *Restaurant, Food, Service, Ambience, Price, Location*, as well as *Miscellaneous*, for Trip-MAML and Restaurant Reviews respectively.

As for preprocessing, we perform the following steps: (1) set maximum length of raw documents to 300; (2) split documents into sentences which are then tokenized into words, and the words are further converted into lowercases; (3) shorten the words with redundant characters into their canonical forms (e.g., coooooool is converted to cool); (4) remove URLs and HTML tags such as <br/>; (5) remove the duplicates and records with empty or invalid content. Furthermore, we convert all rating ranges in all datasets to $[1, 5]$ and therefore the C is set to 5. For each dataset, we randomly split the training and testing set according to the ratio of 80:20. Moreover, 10% reviews in the training set are left out as a validation set for hyper-parameter selection. Note that for records in the testing set, at least one interaction for each user or item is included in the training set, and otherwise will be moved from the testing set to the training set.

4.4.2 Evaluation Metrics

Performance of rating prediction tasks is evaluated on the testing set via Mean Square Error (MSE) which is widely adopted in the recommendation domain.

Despite the importance on measuring the recommendation performance of MSE, user experience can be greatly enhanced if the systems provide diverse recommendations. To evaluate the diversity of our proposed method, we first generate top-N recommendation list $L(N)$ to the target user according to $\hat{r}_{u,i}$ in descending order. More advanced ranking algorithms are out of the scope in this paper. These N items should present various characteristics in terms of i.e. aspects. Then the following metrics are utilized in this paper as measurements:

Intra-list Similarity. This metric proposed by [240] assesses diversity on an individual level. The rationale behind this metric is that each user prefers recommendations from various categories. Assuming i and j are two different items in the recommendation list, the similarity between i and j can be measured via binary similarity calculated upon the training set, which is defined as:

$$Sim(i,j) = \frac{\#users\ that\ click\ both\ i\ and\ j}{\sqrt{\#users\ that\ click\ i} \cdot \sqrt{\#users\ that\ click\ j}} \tag{27}$$

Thus the intra-list similarity (ILS) can be defined as:

$$ILS = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \sum_{(i,j) \in L(N), R(i) < R(j)} Sim(i,j) \tag{28}$$

The lower the ILS value is, the more diverse the recommender system is.

Novelty. The novelty of a recommender system evaluates the likelihood of a recommender system to give recommendations to the user that they are not aware of, or that they have not seen before. The definition of novelty is varied in publications according to its context and purpose. In this paper, we apply the population-oriented item novelty evaluation metric introduced in [241] as expected popularity complement (EPC) to measure the ability of our recommender system to recommend items from long-tail. Its definition is shown below:

$$EPC = \frac{\sum_{u \in \mathcal{U}} \sum_{r=1}^{N} \frac{rel(u,i_r) * (1 - pop(i_r))}{log_2(r+1)}}{\sum_{u \in \mathcal{U}} \sum_{r=1}^{N} \frac{rel(u,i_r)}{log_2(r+1)}} \tag{29}$$

where i_r denotes the item ranked to the r-th place in the recommendation list. $rel(u, i_r)$ is a binary function with values of 1 or 0 representing if the user u rated the item i_r or not respectively. The popularity $pop(i)$ is calculated based on the times the item has been rated in training set, and can be defined as:

$$pop(i) = \frac{|rate(i)|}{max_{j \in \mathcal{I}} |rate(j)|} \tag{30}$$

where $rate\ (i)$ denotes the number of ratings of item i and the denominator is the maximum number of ratings obtained by an item in item set. It is desirable for a recommender system to have a high EPC value when it not only recommends items from long-tail but also ranks them highly in the recommendation list.

Besides, we adopt the precision (P), recall (R), and F1-score as evaluation metrics for multilingual aspect-based sentiment analysis.

4.4.3 Baselines

We compare MrRec with several comparative baselines:

- **MF** [17]: It characterizes users and items by vectors with implicit feedbacks inferred from item rating patterns.
- **NAIS** [242]: It learns the importance of user's historical clicking items via a neural attention network which is then integrated into the item-based collaborative filtering for rating prediction.
- **Tran-D-Attn** [243]: It models user preferences and item characteristics by CNNs with dual local and global attention mechanism for review rating prediction.
- **Tran-ALFM** [219]: It is an aspect-based recommender system with aspect discovered by an aspect-aware topic model on review texts. A weighted matrix is introduced to associate latent factors with aspects by using MF approach to predict ratings.
- **Tran-ANR** [221]: It performs aspect-based representation learning to model both user preferences and item properties. The neural co-attention mechanism is introduced to learn the aspect-level user and item importance.
- **Tran-CARP** [146]: The model predicts ratings based on sentiment-aware representations of user-item interactions, which are learned via a novel Routing by Bi-Agreement mechanism.
- **CL-Babelfy** [222]: This is a content-based recommender system aiming to generate crosslingual recommendations using knowledge-based strategies to build the bond between different languages. In [222], the authors extracted concepts from Wikipedia or BabelNet. Here, we adopt BabelNet[15] since it can lead to better recommendation performance on the two multilingual datasets.

We evaluate our multilingual aspect-based sentiment analysis module with the following comparative approaches:

- **CLJAS** [123]: It jointly performs aspect-specific sentiment analysis of two languages simultaneously by incorporating sentiment parameter into a cross-lingual topic model.
- **Tran-AT-LSTM** [244]: The attention mechanism is adopted in LSTM to generate the sentence representation. The aspect embedding is used to compute the attention weights.
- **Tran-CAN**: [245] It introduces sparse and orthogonal regularizations when performing

[15] https://babelnet.org/

aspect-specific sentiment analysis to learn sentiment distributions on the sentence level. Orthogonal regularization is designed especially for reviews with non-overlapping aspectspecific sentiments, which are unknown in two review datasets. Thus, we only adopt sparse regularization for testing.

Note that for monolingual baselines such as D-Attn, ALFM, ANR, CARP, AT-LSTM and CAN, we translate all the reviews from other languages to English using Google Translate[16], and adopt the prefix **Tran-** as an indicator.

4.4.4 Parameter Settings

We initialize our multilingual word embeddings by using the aligned word vectors pre-trained with fastText, while the word embeddings used in the translation baselines for the English language were initialized by Glove[17] [246]. We also initialized the aspect embedding matrix $\boldsymbol{A}$ with the centroids of clusters resulting from running k-means on word embeddings. The orthogonality penalty weight λ was set to 0.9. We experimented with different numbers of aspects ranging from $[2, 8]$ for all datasests and no major difference was shown with the results. For a fair comparison with other aspect-based baselines, i.e. Tran-ALFM, Tran-ANR and Tran-CARP, we set K to 5. The dimension of hidden state output from bi-directional GRU was set to 150. The number of hidden units for each direction was 75. The number of convolution filters N_f was set to 50 for MRM. The number of latent factors d_c, d_r, d_t and d_f were set to 300, 300, 300 and 100 respectively. MrRec was trained with Adam optimizer because Adam uses adaptive learning rates for parameters with different update frequencies and converges faster than vanilla stochastic gradient descent. We tested the initial learning rate of $[0.0001, 0.001, 0.01]$. For the coefficient of $L2$ regularization, $[0.0, 0.0001, 0.01, 0.1]$ was tested. To prevent overfitting, the dropout rate ρ was set to 0.7. The batch size was set as 200 for the Book Reviews datasets while others were set to 100. The model was trained for a maximum of 300 epochs with early stopping, which means that the training will stop if the performance on validation set does not improve in 10 epochs. The final performances are reported after 5 runs with the average test results.

For recommendation baseline methods, we adopted the optimization strategies reported in

[16] https://translate.google.com/

[17] https://nlp.stanford.edu/projects/glove/

their papers to tune the hyper-parameters. The number of latent factors for ALFM was set amongst $\{5, 10, 15, 20, 25\}$ for each dataset through grid search, and the number of latent topics was set to 5. We reuse the implementations such as the number and size of convolutional filters, the number of factors used for the fully connected layer and the activation functions, reported in [243] for D-Attn. For ANR model, we set the width of the local context window c, the number of latent factors h_1, h_2 to 3, 10, and 50 respectively. For CL-Babelfy, we tuned the dimension of feature vectors m, which is selected from $\{5, 10, 15, 20\}$. Other parameters were set to the same as MrRec model if not specified.

As for multilingual aspect-based sentiment analysis baselines, the aspect embedding matrix and parameters were initialized by sampling from a uniform distribution $U(-\sigma, \sigma), \sigma = 0.01$ in the AT-LSTM and CAN models. The parameter of λ was set to 0.1 in CAN. The dimension of word vectors, aspect embeddings, and the size of hidden layer were set to 300 in AT-LSTM and CAN. As for CLJAS, we employed the best parameters that are reported in the paper [123].

4.5 Experiments

4.5.1 Evaluation on Aspect Extraction and Sentiment Prediction

In this section, we conduct experiments to verify if the models are able to extract aspects and predict associated sentiments in different languages simultaneously. Given a review sentence, our MABSA module assigns one or more inferred aspect labels that correspond to the learned weights higher than a threshold τ[18] according to Eq. 4.

Table 4.3: Comparison results of the MABSA part with the baseline methods in terms of Precision, Recall and F1 score. The best results are highlighted in boldface. "*" indicates the improvements are statistically significant for p-value < 0.01 with paired t-test.

Model	**Trip-MAML**						**Restaurant Reviews**					
	Aspect Extraction			**Sentiment Prediction**			**Aspect Extraction**			**Sentiment Prediction**		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Precision*	*Recall*	*F1*
Tran-AT-LSTM	0.725	0.702	0.712	0.691	0.716	0.703	0.636	0.582	0.609	0.614	0.592	0.602
CLJAS	0.773	0.685	0.728	0.722	0.781	0.751	0.682	0.577	0.624	0.631	0.602	0.617
Tran-CAN	0.854	**0.882**	**0.867**	0.793	0.779	0.786	0.791	**0.814**	**0.803**	0.737	**0.723**	0.731
MABSA	**0.876***	0.843	0.859	**0.837***	**0.786***	**0.812***	**0.802***	0.761	0.783	**0.758***	0.714	**0.735***

[18] We set $\tau = 0.2$ for it achieves the best performance after experiments.

A summary of the results of the baselines and our MABSA module over the two datasets w.r.t. aspect extraction and sentiment prediction are reported in Table 4.3[19]. Several observations can be made. First, the values of all evaluation metrics on Trip-MAML dataset are generally higher than that on Restaurant Reviews dataset. It is probably because we have more training samples in Trip-MAML dataset, which gives the model more opportunities to fit the data well during training. Second, we can clearly observe that Tran-AT-LSTM consistently performs worst of all methods since the attention mechanism may scatter the distribution of weights across the whole sentence and thus may introduce noisy words or opinion words from other aspects. Besides, machine translation, to some extent, is unable to take into account the divergence in the expression of sentiments across different languages. Moreover, the performance gain of CLJAS baseline compared with Tran-AT-LSTM mainly benefits from the knowledge transferred from the source language, and therefore can capture more statistics characteristics. Our model outperforms Tran-AT-LSTM and CLJAS on both tasks and datasets for that the usage of bi-directional GRU helps to incorporate contextual information into word embeddings, while CLJAS captures the words co-occurrence based on the assumption of independence of each word in sentences. The utilization of the pretrained multilingual word embeddings that project languages into a shared space also contributes to the performance improvement. Different from Tran-AT-LSTM, we fuse aspect information into word representations when learning attention weights, which to some extent concentrates the importance on more meaningful words. Furthermore, the experimental results show that compared with supervised model Tran-CAN, MABSA can obtain comparable performance on aspect extraction tasks, which have convincingly validated the effectiveness of MABSA in extracting aspects. It is interesting to note that our module outperforms Tran-CAN on sentiment analysis tasks which is probably attributed to the hierarchical attention mechanism (including aspect-level and sentence-level attention nets) and aspect fusion that can learn the most indicative sentiment words associated with each aspect in both overlapping and non-overlapping muli-aspect sentences, while Tran-CAN only adopts sparse regularization term that is inadequate to extract sentiment words of non-overlapping aspects.

[19] Note that the values of P/R/F1 reported are the average over 5 runs, and thus the F1-score cannot be computed directly from corresponding P/R values.

4.5.2 Recommendation Performance Evaluation

Recommendation Effectiveness. Table 4.4 shows the performance comparison of our MrRec model with state-of-the-art methods on the same test dataset. The table is separated in three blocks showing the results on metric MSE, ILS and EPC respectively. From Table 4.4, we can make the following observations:

For the first block, it is not surprising that MF which depends solely on user-item interactions for rating prediction consistently yields worst MSE among all approaches on all datasets, which we believe validates the importance of contextual information in reviews. Though NAIS only adopts item IDs rather than textual reviews as inputs, it outperforms review-based method, CL-Babelfy, which we believe is probably credited to the powerful representation learning capacity of neural models. Among all translation based approaches, Tran-D-Attn model perform worse than others, which is because the model does not consider aspect-level features when modelling users and items and thus cannot capture the fine-grained characteristics of users/items. Whereas Tran-CARP achieves the lowest MSE among all translation-based baselines over all datasets, which shows that the aspects and aspect-specific sentiments derived from textual reviews play crucial roles in improving recommendation performance. Though both Tran-ANR and Tran-ALFM attempt to utilize aspects in their architectures, Tran-ANR outperforms Tran-ALFM, whose major drawback is that the proposed model leverages topic model to learn the statistical features of words in reviews which neglects the contextual information around the word. Our model shows comparable performance compared with Tran-CARP and even shows superior performance on most datasets that have more multilingual reviews.We believe this benefits from the language attention mechanism that can learn the different contributions of reviews in multiple languages and multilingual word embeddings that jointly mine semantic information with textual reviews written in various languages.

Diversity and novelty are measured by ISL and EPC with the results displayed in the rest two blocks, from which we can observe that our MrRec exhibits the dominating performance among all methods across 9 datasets. We argue that this is attributed to the aspect utility estimation mechanism which takes into consideration both like-minded users for target user and similar items in user's historical records with candidate item. There is no dominating winner among neural network baseline methods, but they outperform CL-Babelfy and MF on EPC, which is because they focus more on historical user preferences and thus tend to recommend items similar with items user clicked before rather than the

Table 4.4: Comparison results of the MRM part with the baseline methods in terms of Mean Square Error (MSE), Intra-list Similarity (ILS) and Novelty (EPC). The best results are highlighted in boldface. "*" indicates the improvements are statistically significant for p-value < 0.01 with paired t-test.

Measures	*Methods*	*Books*	*Digital Ebook Purchase*	*Digital Music Purchase*	*Digital Video Download*	*Mobile Apps*	*Music*	*Toys*	*Video DVD*	*Goodreads*
MSE	*MF*	2.487	2.279	2.583	2.426	2.067	2.361	2.503	2.184	2.091
	CL-Babelfy	2.361	2.183	2.504	2.337	1.972	2.254	2.396	2.068	1.995
	NAIS	2.082	1.969	2.211	2.049	1.746	2.015	2.152	1.833	1.771
	Tran-D-Attn	1.935	1.823	2.086	1.902	1.627	1.885	2.011	1.726	1.654
	Tran-ALFM	1.746	1.628	1.893	1.731	1.433	1.692	1.825	1.542	1.467
	Tran-ANR	1.633	1.527	1.781	1.618	1.326	1.587	1.714	1.425	1.348
	Tran-CARP	1.481	1.366	**1.635**	1.464	1.169	1.437	1.556	**1.232**	**1.183**
	MrRec	**1.307***	**1.253***	1.682	**1.288***	**1.036***	**1.269***	**1.392***	1.243	1.189
	Improvement (%)	11.75~47.45	8.27~45.02	-2.87~34.88	12.02~46.91	11.38~49.88	11.69~46.25	10.54~44.39	-0.89~43.09	-0.51~43.14
ILS	*MF*	8.756	7.427	10.683	8.894	7.048	8.347	9.530	7.795	7.052
	CL-Babelfy	9.877	8.352	11.565	10.032	7.917	9.233	11.158	8.937	7.426
	NAIS	10.453	8.781	11.672	10.684	7.885	9.931	11.797	9.732	8.278
	Tran-D-Attn	11.283	9.693	12.462	11.531	8.392	12.846	12.135	9.524	8.732
	Tran-ALFM	12.732	10.662	14.959	13.263	9.531	12.182	13.677	12.151	9.041
	Tran-ANR	12.345	10.236	13.501	14.025	8.825	11.473	12.972	10.604	9.727
	Tran-CARP	13.527	11.379	14.153	12.672	10.257	10.746	14.579	11.372	10.562
	MrRec	**8.283***	**7.264***	**9.565***	**8.667***	**6.371***	**7.716***	**9.372***	**7.386***	**6.558***
	Improvement (%)	5.40~38.77	2.19~36.16	10.47~36.06	2.62~38.20	9.61~37.89	7.56~39.93	1.66~35.72	5.25~39.21	7.01~37.91
EPC	*MF*	0.653	0.598	0.621	0.633	0.586	0.639	0.591	0.672	0.694
	CL-Babelfy	0.673	0.616	0.641	0.670	0.592	0.672	0.604	0.684	0.709
	NAIS	0.693	0.621	0.675	0.683	0.605	0.687	0.612	0.706	0.723
	Tran-D-Attn	0.712	0.636	0.702	0.708	0.663	0.714	0.635	0.753	0.766
	Tran-ALFM	0.748	0.719	0.713	0.750	0.648	0.742	0.661	0.826	0.825
	Tran-ANR	0.775	0.708	0.748	0.739	0.611	0.781	0.674	0.772	0.841
	Tran-CARP	0. 794	0.687	0.741	0.763	0.681	0.766	0.692	0.831	0.846
	MrRec	**0.842***	**0.775***	**0.798***	**0.817***	**0.732***	**0.826***	**0.767***	**0.859***	**0.873***
	Improvement (%)	6.05~28.94	7.79~29.60	6.68~28.50	7.08~29.07	7.49~24.91	5.76~29.26	10.84~29.78	3.37~27.83	3.19~25.79

popular items. Because of the same reason, they neglect the diversification on candidate items of the user potential interests, and therefore perform worse than CL-Babelfy and MF on ILS.

Recommendation Efficiency. Figure 4.6 illustrates the log scale training time comparisons between MRM module and Tran-CARP, the best performance with MSE among all baselines. Though our MrRec is composed of three steps, we report the training time only for rating prediction, the last step, since step 1 and step 2 that are used to extract aspects and aspect-specific sentiments can be learned offline separately. Compared with other review-based methods which usually feed the reviews with embedding vectors of all words into model, our inputs of MRM are achieved through aspect-based representations of all sentences, and thus the size of input for a specific user is changed from $O(|\mathcal{F}_u| \times N_w \times d_e)$ to $O(|\mathcal{F}_u| \times N_s \times K \times d)$. Here N_w and N_s represent the average words and average sentences per review respectively. Form Table 4.1 we can see $N_s \ll N_w$. Actually in practice, $N_s \times K$ is usually smaller than N_w. Therefore our MRM module can accelerate the training efficiency. Figure 4.6 also verify our analysis. The MRM module trains 6.5 to 8 times faster than Tran-CARP over all datasets, which benefits from the reduction of input size and decomposition of training tasks.

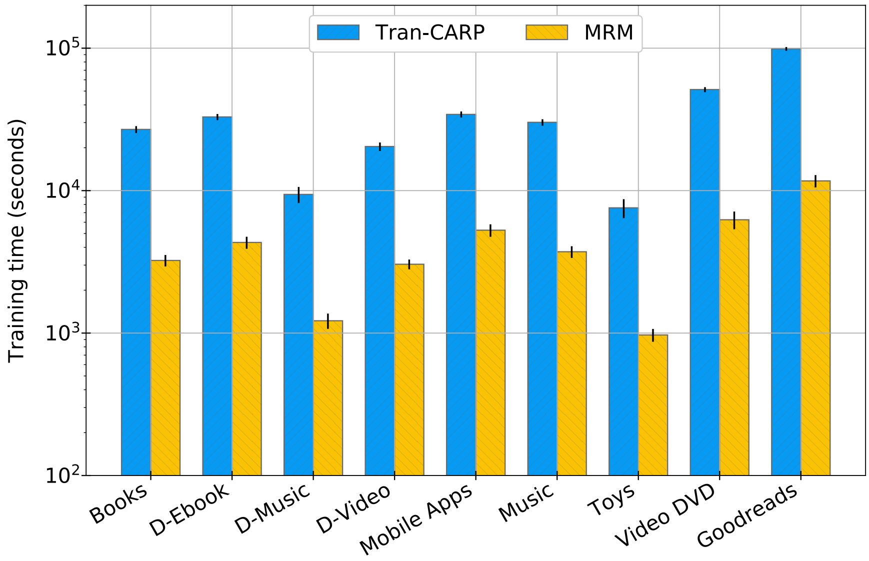


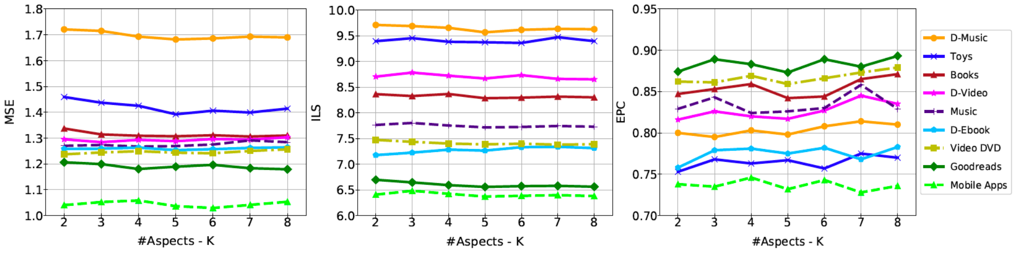
Figure 4.6: Runtime comparison (seconds) for training model on all datasets. D-Ebook, D-Music and D-Video are short for Digital Ebook Purchase, Digital Music Purchase and Digital Video Download.

4.5.3 Effects of the Hyper-Parameters

In this section, we analyze the influence of embedding size d and the number of aspects K on the final performance of MrRec. We optimize one parameter with another one fixed to see how performance will change accordingly.

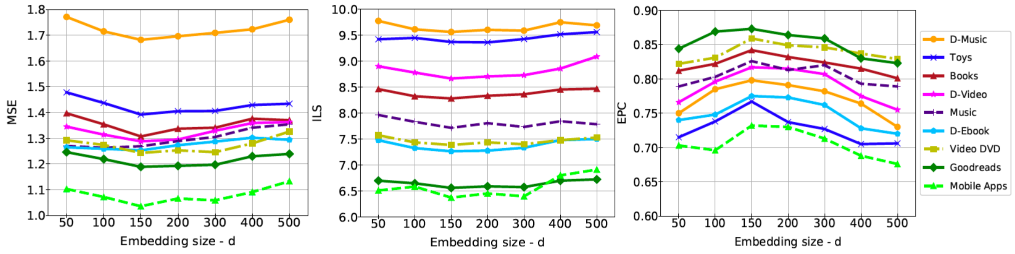
The empirical results displayed in Figure 4.7 indicate the effect of varying the number of aspects K from 2 to 8 for our model w.r.t. MSE, ILS and EPC across 9 datasets. We can observe that though the optimal value of K varies across different datasets, the overall trends are relatively stable. The comparatively good performance can be achieved with 5 aspects. We hypothesize that adjusting the number of aspects can only influence the granularity of modelling the textual reviews. As such, varying K within a reasonable range has little impact on the recommendation performance.

Figure 4.8 illustrates the effect of varying the embedding size d from 50 to 500 across multiple datasets on three metrics. As can be observed, the performance keeps improving with d ranging from 50 to 150 on most datasets. The highest performance appears with d set around 150 and remains relatively stable before d equals to 300. However, the results show the turbulent trends when d is higher than 300, which indicates that further use of larger embedding size does not show significant improvement. Thus, we set $d = 150$ in our experiments.



(a) Mean Square Error (b) Intra-list Similarity (c) Novelty

Figure 4.7: Effect of the number of aspects.

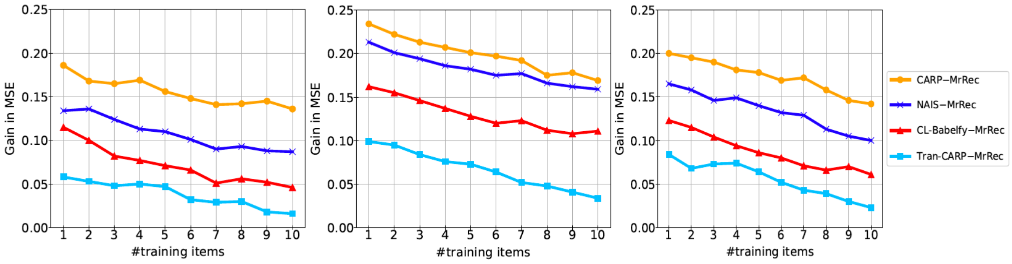


(a) Mean Square Error (b) Intra-list Similarity (c) Novelty

Figure 4.8: Effect of the Bi-GRU output dimension.

4.5.4 Cold Start Evaluation

For monolingual scenario, cold start refers to users/items with limited ratings which makes it difficult to provide satisfactory recommendations for monolingual recommendation models. MF method can easily lead to cold start issue since there are only few user-item interactions available. In contrast, review-based methods can alleviate the problem, since reviews contain rich contextual information on users' preference and item characteristics. However, we argue that such problem can further be alleviated by introducing resources from other languages, i.e. textual reviews written in different languages. To verify this assumption and demonstrate the capability of our model in dealing with multilingual user-item interactions, in this section, we conduct experiments on multilingual datasets with our MrRec model and different baselines, i.e. Tran-CARP, NAIS and CL-Babelfy. We also compare our model with the original version of Tran-CARP, namely CARP, to test that to what extent translation can help to improve multilingual recommendations.



(a) Digital Music Purchase (b) Mobile Apps (c) Goodreads

Figure 4.9: Performance on the cold start problem.

The experiments are conducted on selected three of nine datasets, Digital Music Purchase, Mobile Apps and Goodreads with the highest, lowest and middle ratio of multilingual interactions respectively. As preprocessing, we first filter out monolingual user-item interactions, and then split the datasets into training, validation, and testing set based on the number of ratings in each set.We also remove users from the testing set who have no ratings in the training set. We evaluate the performance of users who have the number of ratings from 1 to 10 in the training set. Figure 4.9 shows the **Gain in MSE** grouped by the number of user ratings. Gain in MSE can be calculated by the average MSE of baselines minus that of our model, i.e. CARP-MrRec. As can be seen, similar trends can be found across all datasets. Our MrRec model consistently outperforms other baselines on three datasets since the differences are all positive values. In particular, Tran-CARP substantially improves the rating prediction accuracy compared with CARP, which we believe verifies the importance and benefit of leveraging multilingual reviews for recommendations. Besides, our MrRec model beats the other baselines that integrate multilingual resources. This is attributed to the fact that our model is more effective in exploiting and modelling textual reviews in different languages.

4.5.5 Ablation Study

In this section, we perform an ablation study to analyze how different components in our proposed model contribute to the overall performance. The experiments are conducted among variants of MrRec and the complete model (denoted as "baseline") with hyper-parameter settings as stated in Section 4.4.4. We incorporate the following variants:

- **RandomWord Embeddings (RWE):** Instead of using pre-trained multilingual word embeddings as inputs to our model, we train our convolutional model on word embeddings initialized randomly from a uniform distribution. The word embeddings are part of the trainable parameters of the network in this model.
- **Without Bi-directional GRU Layer (Without Bi-GRU):** In order to show the effect of adopting Bi-directional GRU Layer to the word representations, we choose to remove the Bi-directional GRU Layer to test its effectiveness in the MABSA module.
- **Without Aspect-level Interactions (Without ALI):** We forgo co-attention network and aspect utility estimation component in our model. Instead, we apply a fully-connected layer upon the concatenation of $\boldsymbol{u}_{a_k}/\boldsymbol{i}_{a_k}$on all aspects to learn the user/item representation. Similar to D-Attn, the user and item representations are then adopted to derive the overall rating.
- **Uniform Language Importance (ULI):**We vieweach language as equal importance. Specifically, $\eta_{a_k}^{l}$ is set to $1/L$ in Eq. 18.
- **Without Aspect Utility Estimation (Without AUE):** We remove the aspect utility estimation component, and use aspect-based user/item representation to predict the overall ratings.
- **Uniform Aspect Importance (UAI):** In Eq. 20, each δ_{u,a_k} is replaced with $1/K$ to verify the importance of co-attention network.

The results are shown in Table 4.5 for the Mobile Apps and Goodreads datasets. As shown in the table, we can observe that the lack of aspect-level interaction component can lead to

Table 4.5: Comparison of the model variants for the Mobile Apps and Goodreads datasets. The worst and second-worst results are highlighted in boldface and underlined respectively.

Setup	**Mobile Apps**			**Goodreads**		
	MSE	*ILS*	*EPC*	*MSE*	*ILS*	*EPC*
Baseline	1.036	6.371	0.732	1.189	6.558	0.873
RWE	1.432	6.946	0.701	1.508	7.161	0.821
Without Bi-GRU	1.115	6.696	0.712	1.257	6.959	0.837
ULI	1.125	6.789	0.705	1.263	7.020	0.818
UAI	1.143	6.748	0.708	1.278	6.985	0.831
Without AUE	1.229	7.638	0.633	1.307	7.860	0.763
Without ALI	**1.519**	**8.263**	**0.627**	**1.605**	**8.496**	**0.752**

large performance degradation on both datasets over three metrics. The performance deteriorates secondly on ILS and EPC when our model is without aspect utility estimation component, which verifies that this component does improve the diversity and novelty of recommendations. Besides, we find that the pre-trained multilingual word embedding provides a crucial starting point for multiple language integration and consequently, affect the overall rating prediction. Finally, excluding either Bi-directional GRU, language attention network, or co-attention network can cause the degradation of recommendation performance to different degrees, which highlights the demanding of each component in improving the rating prediction, system's diversity as well as novelty.

4.5.6 Interpretability Visualization

In this paper, a user's preference on an item can be decomposed into the user's preference on different aspects with considering the importance of those aspects from both user and item sides, as well as the sentiment utilities exhibiting from the aspects discovered based on multilingual textual reviews. The learned aspects for the user can be expressed with their representative words which are found by looking at the nearest words from his/her reviews in the embedding space using cosine as the similarity metric. Specifically, the cosine similarity is calculated between the aspect representation $\boldsymbol{a}_k$ from Eq. 5, and the word representation $\boldsymbol{h}_i$ from Eq. 1: $sim(k, i\,) = cos\,(\boldsymbol{a}_k, \boldsymbol{h}_i)$. The higher value of $sim(k, i\,)$ is desirable for the word w_i belonging to the k-th aspect. The top 10 aspect words in each

Table 4.6: Top ten words of each aspect in English (EN) and French (FR) for a user (id 50989966) from Video DVD dataset. Each column is corresponding to an aspect attached with an "interpretation" label. $\eta^{l}_{a_k}$ denotes the contribution of language l on aspect a_k for the target user.

Film		**Style**		**Time**		**Character**		**Value**	
EN	*FR*	*EN*	*FR*	*EN*	*FR*	*EN*	*FR*	*EN*	*FR*
$(\eta^{en}_{a_1}:0.352)$	$(\eta^{fr}_{a_1}:0.648)$	$(\eta^{en}_{a_2}:0.365)$	$(\eta^{fr}_{a_2}:0.635)$	$(\eta^{en}_{a_3}:0.419)$	$(\eta^{fr}_{a_3}:0.581)$	$(\eta^{en}_{a_4}:0.337)$	$(\eta^{fr}_{a_4}:0.663)$	$(\eta^{en}_{a_5}:0.473)$	$(\eta^{fr}_{a_5}:0.527)$
story	*téléfilm* $(TVmovie)^{\star}$	*fiction*	*genre* *(kind)*	*august*	*année* *(year)*	*artists*	*associée* *(partner)*	*sale*	*affaires* *(business)*
theatre	*épisodes* *(episodes)*	*documentary*	*comédie* *(comedy)*	*october*	*mars* *(March)*	*referee*	*épouse* *(wife)*	*profit*	*score* *(score)*
movie	*éditions* *(editions)*	*comedy*	*fiction* *(fiction)*	*months*	*septembre* *(September)*	*individuals*	*police* *(police)*	*free*	*dollars* *(dollars)*
episode	*personnages* *(characters)*	*historical*	*musical* *(musical)*	*medieval*	*vie* *(life)*	*children*	*artistes* *(artists)*	*money*	*champion* *(champion)*
actors	*scénariste* *(scriptwriter)*	*album*	*historique* *(historical)*	*hours*	*présentent* *(present)*	*man*	*juifs* *(Jews)*	*million*	*libre* *(free)*
actress	*actrice* *(actress)*	*musical*	*documentaire* *(documentary)*	*life*	*actuel* *(current)*	*director*	*chanteuse* *(singer)*	*industry*	*millions* *(million)*
character	*studio* *(studio)*	*philosophy*	*exposition* *(exhibition)*	*diff*	*évoluant* *(evolving)*	*chief*	*chiffre* *(figure)*	*economic*	*moins* *(minus)*
description	*fin* *(end)*	*military*	*action* *(action)*	*throughout*	*quand* *(when)*	*winner*	*infanterie* *(infantry)*	*material*	*meilleurs* *(best)*
families	*séries* *(series)*	*social*	*images* *(images)*	*further*	*finale* *(final)*	*brothers*	*parisien* *(Parisian)*	*commercial*	*haute* *(high)*
sports	*écrivain* *(writer)*	*criminal*	*sociale* *(social)*	*november*	*parfois* *(sometimes)*	*citizens*	*filles* *(girls)*	*trade*	*mesure* *(measure)*

Table 4.7: Interpretation for why the "user 50989966" rated "item1" and "item 2" with 4 and 2, respectively, from Video DVD dataset.

Aspects	*Film*	*Style*	*Time*	*Character*	*Value*
Importance for User (1)	0.214	0.097	0.014	0.653	0.022
Importance for Item (1)	0.103	0.015	0.007	0.861	0.014
Aspect Utility (1)	0.871	0.526	0.145	0.922	-0.263
Importance for User (2)	0.203	0.018	0.003	0.712	0.064
Importance for Item (2)	0.129	0.073	0.006	0.784	0.008
Aspect Utility (2)	-0.576	-0.691	0.193	-0.879	-0.154

language of user u from Video DVD dataset are shown in Table 4.6. The contributions of different languages, i.e. English and French user u adopted in total, are listed under the name of each aspect. For instance, $\eta_{a_1}^{en}$: 0.352 represents the contribution of English for aspect a_1 is less than that of French. As shown in Table 4.6, the five aspects can be semantically interpreted to Film, Style, Time, Character, and Value. The top aspect words of candidate items can also be achieved in the same way, but here we only illustrate on the user side. Then in Table 4.7, we demonstrate how to interpret the high and low ratings the user u giving to items on the same dataset. From the table, we can see the aspect importance $\boldsymbol{\delta}_u$ for user and $\boldsymbol{\delta}_i$ for item from Eq. 20, as well as aspect utility $y_{u,i}^{a_k}$ from Eq. 25 w.r.t. "item 1" and "item 2". As can be observed, the user pays more attention to Character and Film aspects on both items. Similarly, "item 1" and "item 2" put more importance on Character, and Film. However, the user is more satisfied with Character and Film on "item 1" than that on "item 2". As a result, according to Eq. 26, the overall rating of "item 1" should be apparently higher than that of "item 2", which is 4 to 2 respectively. From the illustration, we can see that our model could capture to what extent the user likes or dislikes an item on an aspect and interpret the recommendation results at a fine level of granularity.

4.6 Conclusion

In this paper, we have proposed for the first time a multilingual review-aware deep recommendation model (MrRec) for overall rating prediction and item recommendation. The model requires neither external translation tools nor knowledge bases to analyze multilingual reviews. Particularly, instead of labelled datasets, MrRec extracts aspects and analyzes aspect-specific sentiments requiring merely overall ratings which are leveraged as user sentiments to remove the possible ambiguity contained in the textual reviews. Besides, our model is able to estimate aspect importance for each user-item pair by utilizing co-attention network on the learned aspect-based user/item representations with considering the different contributions of multiple languages. Furthermore, user satisfaction is embodied by the aspect utility derived from a dual interactive attention mechanism with considering both like-minded users to the target user and similar items with the candidate item. Finally, the overall rating is predicted by adopting a prediction layer on the combination of learned aspect utility and aspect importance. We have compared the MrRec with state-of-the-art baselines on 9 real-world datasets and experimental results demonstrate the effectiveness and efficiency of our model on recommendation accuracy, as well as recommendation diversity, especially for cold start users/items in the monolingual scenario but with extra reviews written in other languages.

Part III

Analyzing Network Structure for Social Recommendation

Chapter 5
Network Representation Learning for Social Recommendation

With the rapid proliferation of online social networks, personalized social recommendation has become an important means to help people discover their potential friends or interested items in real-time. However, the cold-start issue and the special properties of social networks, such as rich temporal dynamics, heterogeneous and complex structures, render the most commonly used recommendation approaches (e.g. Collaborative Filtering) inefficient. In this chapter, we propose a novel dynamic graph-based embedding (DGE) model for social recommendation which is capable of recommending relevant users and interested items. In order to support real-time recommendation, we construct a heterogeneous user-item (HUI) network and incrementally maintain it as the social network evolves. DGE jointly captures the temporal semantic effects, social relationships and user behavior sequential patterns in a unified way by embedding the HUI network into a shared low dimensional space. Then, with simple search methods or similarity calculations, we can use the encoded representation of temporal contexts to generate recommendations. We conduct extensive experiments to evaluate the performance of our model on two real large-scale datasets, and the experimental results show its advantages over other state-of-the-art methods.

5.1 Introduction

With the rapid development of Web 2.0 and smart mobile devices, online social networks have proliferated and are still promptly growing. According to Twitter statistics, the number of users is estimated to have surpassed 300 million generating more than 200 million tweets per day[20]. Faced with the abundance of user generated content, a key issue

[20] https://blog.twitter.com/2011/200-million-tweets-per-day

of social networking services is how to help users find their potential friends or interested items that match the users' preference as much as possible, by making use of both semantic information and social relationships. This is the problem of personalized social recommendation.

Generally, different techniques used in building personalized recom- mender systems are mainly divided into three categories: collaborative filtering, content-based filtering and hybrid system [14]. Although previous techniques have been shown to be effective to some extent, there still exist two major challenges in front of online social networks. First, the complex structures in the social network need to be properly mined and exploited by algorithms. Second, these networks contain millions or even billions of edges making the problem very difficult computationally. For example, the efficiency of classic item-based k nearest neighbor (KNN) recommendation algorithms is largely limited by the construction of the KNN graph [247]. Matrix factorization involves eigen-decomposition of the data matrix which is expensive and usually with approximation calculation [248]. Therefore, it is crucial to handle large-scale heterogeneous networks for social recommender system.

In recent years, there have been numerous studies exploiting different types of relationships in heterogeneous networks [1, 2, 174] to improve the quality of recommendations. However, considering the dynamic nature of social network, almost all existing social recommendation methods are incapable of supporting real-time recommendation principally, and they would suffer from the following three drawbacks: 1) Delay on model updates caused by the expensive time cost of re-running the recommender model. 2) Disability to track changing user preferences due to the fact that latest entries used for updating recommendation models are often overwhelmed by the large data of the past. 3) Cold start problem becomes even more severe in online social networks as the new users and new items will join in the recommender system constantly over time. Some online learning algorithms address this problem by keeping a representative sample of the data set in a reservoir to retrain the model [209], which however is not appropriate for large streaming data set. To avoid this problem, some other online algorithms propose to update the model based solely on the current observation [249], at the cost of reducing the quality of recommendations.

In this work, our goal for social recommendation is to provide real-time and accurate recommendation services for users in large-scale heterogeneous networks. Specifically, it demonstrates three requirements. First of all, the recommender system needs to produce accurate recommendations for users. Second, the model should be updated in real-time to

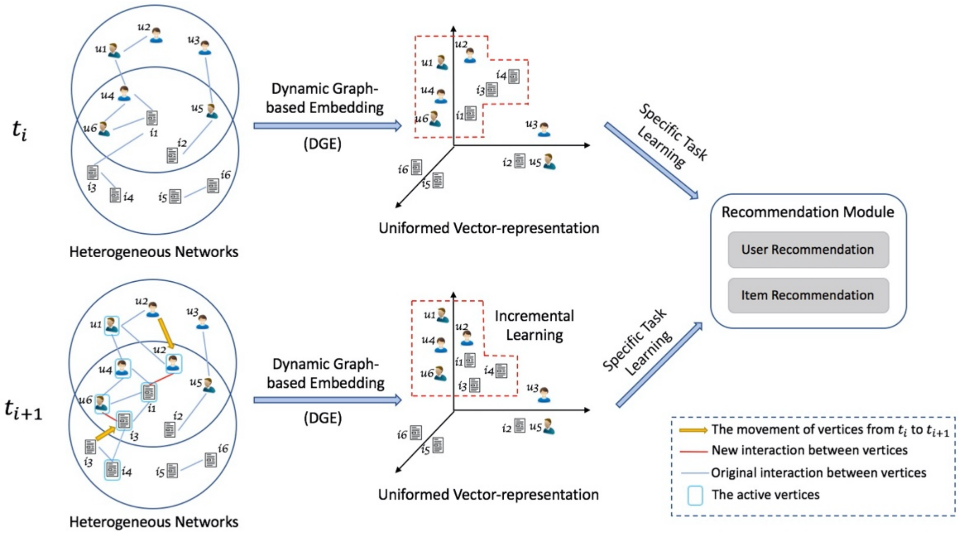


Figure 5.1: The flowchart of dynamic graph-based embedding framework.

capture users' instant interests and social network evolution in very short delay. Third, the processing needs to be executed in parallel, i.e., scalable to handle large amounts of computations.

To fulfill the aforementioned goals, we propose a novel dynamic graph-based embedding (DGE) model which can effectively recommend relevant users and interested items in real-time. Inspired by recent progress in network representation learning and deep learning [41, 43, 250], we propose to use the distributed representation method for modeling online social networks. Specifically, we construct a heterogeneous user-item (HUI) network, in which the two types of vertices represent users and various items and the three types of edges respectively characterize the semantic effects, social relationships and user behavior sequential patterns. Based on the differential behaviour represented among continuous time slots, the HUI network is incrementally maintained as the social network evolves. Then, an incremental learning algorithm is applied to embed the HUI network into low-dimensional vector spaces, in which the proximity information of each vertex is encoded into its learned vector representation. Afterwards, we use the learned representations of vertices with some simple search methods or similarity calculations to conduct the task of social recommendation. Figure 5.1 illustrates the idea of dynamic graph-based embedding framework. To summarize, this chapter makes the following contributions:

– We propose a dynamic graph-based embedding model that integrates the temporal semantic effects, social relationships and user behavior sequential patterns into the process of network embedding. To the best of our knowledge, this work is the first to address real-time social recommendation by a network representation learning approach.

– We devise a transition probability matrix P for the complex HUI network to capture the semantic effect of different edge types. Based on this, an asynchronous parallel stochastic gradient descent method is proposed to allow horizontally scaling the algorithm for large-scale social networks and improve the efficiency of the inference.

– To speed up the process of producing top-k recommendations from large-scale social media streams, we develop an efficient query processing technique by extending the Threshold Algorithm (TA) [251].

– We conduct extensive experiments to evaluate the performance of our model on two real large-scale datasets. The results show the advantages of our method for social recommendation in comparison with state-of-the-art techniques.

The remainder of the chapter is organized as follows. In Section 5.2, we formally define our problem and give the definition of each source for the heterogeneous network. Section 5.3 presents our new model. We describe the data sets, comparative approaches and the evaluation criteria we use in Section 5.4. Section 5.5 shows our experiment results. Finally, we present the conclusions and future work in Section 5.6.

5.2 Problem Formulation

In this section, we first introduce the key data structures and the definition of each source for the heterogeneous network. Then, the problem statement of this study is presented. Table 5.1 summarizes the notations of frequently used variables.

Definition 1. Item Profile An item is defined as a uniquely post (e.g., a tweet or a news article). In our model, an item can be denoted as a five tuple $(iId, \mathcal{M}, \mathcal{H}, \mathcal{W}, \rho)$, representing itemID, named entity, hashtag/category, content, create time respectively.

Definition 2. User Profile For each user u, we create the user profile as a three tuple $(uId, \mathcal{L}, \mathcal{D})$, which indicates userID, user social links and a set of items associated with u.

Definition 3. User-user Relationship Network A user-user relationship network can be

Table 5.1: Notations used in the chapter.

Symbol	Description
$\mathcal{U}, \mathcal{P}$	the set of users and items
$\mathcal{M}, \mathcal{H}, \mathcal{W}, \mathcal{L}$	the set of named entities, hashtags/categories, content words and social links
G_{mix}	heterogeneous user-item (HUI) network
t	the timestamp of heterogeneous user-item network
$\mathbb{R}^d$	d dimensional latent space
$\vec{v}, \vec{p}$	embeddings of user u and item p, respectively
Δt	the time interval
α, β, γ	model parameters controlling the relative importance of user behavior sequential patterns, social relationships and semantic effects
$\mathcal{R}_i$	the social links of user u_i
P	the transition probability matrix of heterogeneous user-item network
$\widetilde{\mathcal{V}}_t$	the active nodes at timestamp t

represented by $G_{uu} = (\mathcal{U}, \varepsilon_{uu})$, where $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ is the set of users, and ε_{uu} is the set of edges. Each $e_{ij} \in \varepsilon_{uu}$ is a social link, such as following or friends, between user i and user j.

Definition 4. Item-item Relationship Network An item-item relationship network can be represented by $G_{pp} = (\mathcal{P}, \varepsilon_{pp})$, where $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ is the set of items, and ε_{pp} denotes the set of edges. If item p_i and item p_j have a semantic link such as Named Entity or Hashtag, there will be an edge $e_{ij} \in \varepsilon_{pp}$ between them, otherwise none.

Definition 5. User-item Interaction Network A user-item interaction network can be represented by $G_{up} = (\mathcal{U} \cup \mathcal{P}, \varepsilon_{up})$,, where $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$ is the set of users, $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$ is the set of items, and ε_{up} denotes the set of edges. If item p_j is of interest to user u_i (based on user activities such as 'clicked', 'retweet', etc), there will be an edge $e_{ij} \in \varepsilon_{up}$ between them, otherwise none.

Definition 6. Heterogeneous User-Item (HUI) Network A heterogeneous user-item network can be represented by $G_{mix} = G_{uu} \cup G_{pp} \cup G_{up}$, which consists of the user-user relationship network G_{uu} , the item-item relationship network G_{pp} and the user-item interaction network G_{up}. The same sets of users and items are shared in G_{mix}.

The heterogeneous user-item network can well capture social relationship influence, semantic effect and user behavior sequential patterns simultaneously. Take the semantic effect as an example, we can interpret it as following: if a user u_i is visiting an item p_j at time slot t and item p_k is more similar with p_j than other items, then u_i is most likely to visit p_k. Our goal is to embed the heterogeneous user-item network into a shared low dimensional space $\mathbb{R}^d$ where d is the dimension. Then, we can get the vector representations of users $\vec{v}$ and items $\vec{p}$.

Finally, we formally define the problem investigated in our work. Given a time-stamped heterogeneous user-item network, we aim to provide real-time social recommendations stated as follows.

Problem 1 (Real-time Social Recommendation). Given a heterogeneous user-item network G_{mix} at timestamp t and a querying user $u \in \mathcal{U}$, the task is to generate a ranked list of user or item recommendations that u would be interested in.

5.3 DGE: Dynamic Graph-based Embedding Model

In this section, we propose a novel dynamic graph-based embedding (DGE) model for real-time social recommendation. Firstly, the construction of the HUI network as well as its update process are described in details. Then, we introduce the dynamic graph embedding approach which involves the edge sampling and an incremental learning algorithm. Finally, a list of top-k recommendations can be generated by evaluating the similarities between the learned representations of different vertices.

5.3.1 Heterogeneous User-Item (HUI) Network

HUI Network construction. For notational simplicity, we ignore the time-subscript in this subsection. Assume that we are given a set of users $\mathcal{U} = \{u_1 , u_2, \dots , u_m \}$ and a set of items $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$. To integrate the semantic effects, social relationships and the user behavior sequential patterns simultaneously, we construct a heterogeneous user-item network comprising two types of nodes and three types of edges, as shown in Figure 5.2. The two types of nodes which consist of user and item nodes are formed by projecting the user set and item set respectively. The three types of edges are defined as follows:

1) Each user node u_i and each item node p_j are connected if user u_i shows an interest on item p_j. In the HUI network, such an edge is indicated by yellow solid lines. The associated item nodes of the user node u_i are denoted as $\mathcal{I}_p(u_i)$, the associated user nodes of the item node p_j are denoted as $\mathcal{I}_u(p_j)$.

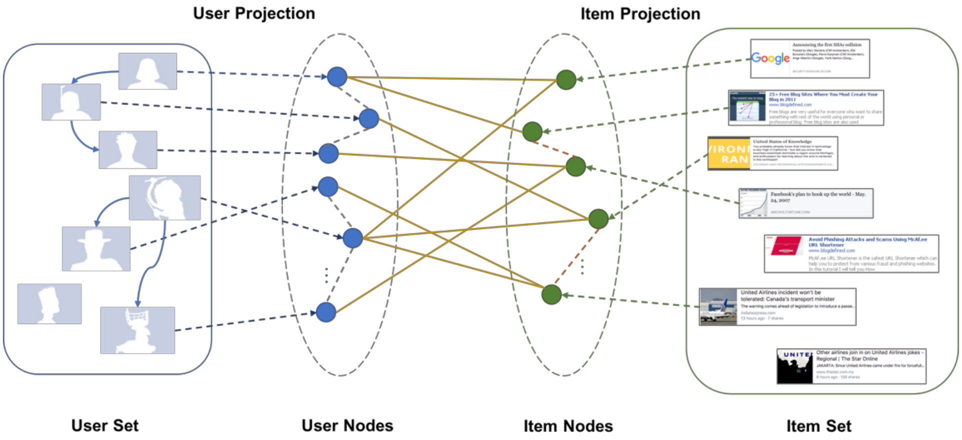


Figure 5.2: The heterogeneous user-item (HUI) network.

2) Two user nodes u_i and u_j are connected with the property of user similarity $sim_u(u_i, u_j)$ if user u_i and u_j have a social link, such as following or friends. In the HUI network, such edge is indicated by grey dash lines. The adjacent user nodes of the user node u_i are denoted as $\mathcal{A}_u(u_i)$.
3) Two item nodes p_i and p_j are connected with the property of item similarity $sim_p(p_i, p_j)$ if item p_i and p_j have a semantic link such as Named Entity or Hashtag. In the HUI network, such edge is indicated by orange dash lines. The adjacent item nodes of the item node p_i are denoted as $\mathcal{A}_p(p_i)$.

We assume that $\mathcal{R}_i$ is a r-dimensional vector representing the social links of user u_i, where r is the total number of users, and the k-th dimension of vector $\mathcal{R}_i$ equals 1 only if there is an edge between u_i and u_k, otherwise 0. The user similarity $sim_u(u_i,\ u_j)$ between user u_i and user u_j can be defined as the cosine similarity between the two vectors,

$$sim_u\left(u_i, u_j\right) = \frac{\mathcal{R}_i^T \cdot \mathcal{R}_j}{\sqrt{\mathcal{R}_i^T \cdot \mathcal{R}_i} \cdot \sqrt{\mathcal{R}_j^T \cdot \mathcal{R}_j}} \tag{1}$$

Likewise, the item similarity $sim_p(p_i, p_j)$ between two item nodes p_i and p_j is also defined as the cosine similarity between the two corresponding feature vectors, which contain named entity, hashtag/category and the occurrence frequency of words in the item content.

Corresponding to the three types of edges with different characteristics, there are three types of random walk modes, which are between user nodes, between item nodes as well as between user and item nodes. Directly applying random walk to the HUI network does not work due to different edge types, leading to a challenging problem. To this end, we propose a novel way to capture the different edge type characteristic into the transition probability matrix P, where three parameters α, β, γ with $\alpha + \beta + \gamma = 1$ are used to respectively control the relative importance of user behavior sequential patterns, social relationships and semantic effects.

Definition 7. A transition probability matrix $P \in \mathbb{R}^{(m+n)\times(m+n)}$ is constructed for the HUI network,

$$P = \begin{pmatrix} p_u & p_{up} \\ p_{pu} & p_p \end{pmatrix} \tag{2}$$

which comprises four matrix blocks $P_u \in \mathbb{R}^{m\times m}$, $P_{up} \in \mathbb{R}^{m\times n}$, $P_{pu} \in \mathbb{R}^{n\times m}$ and $P_p \in \mathbb{R}^{n\times n}$ respectively representing the transition probabilities of random walks between user nodes, from user nodes to item nodes, from item nodes to user nodes and between item nodes. That is

$$\begin{aligned} P_{i,j} &= Prob(\, u_j | u_i \,), \qquad i \, < \, m, j \, < \, m \\ &= \begin{cases} 0 & u_j \notin \mathcal{A}_u(u_i) \\ \frac{\beta}{\alpha+\beta} \times \frac{sim_u(u_i, u_j)}{\sum_{u_k \in \mathcal{A}_u(u_i)} sim_u(u_i, u_k)} & u_j \in \mathcal{A}_u(u_i) \end{cases} \end{aligned} \tag{3}$$

$$\begin{aligned} P_{i,m+j} &= Prob(\, p_j | u_i \,), \qquad i \, < \, m, j \, < \, n \\ &= \begin{cases} 0 & p_j \notin \mathcal{I}_p(u_i) \\ \frac{\alpha}{\alpha+\beta} \times \frac{1}{|\mathcal{I}_p(u_i)|} & p_j \in \mathcal{I}_p(u_i) \end{cases} \end{aligned} \tag{4}$$

$$\begin{aligned} P_{m+i,j} &= Prob(\, u_j | p_i \,), \qquad i \, < \, n, j \, < \, m \\ &= \begin{cases} 0 & u_j \notin \mathcal{I}_u(p_i) \\ \frac{\alpha}{\alpha+\gamma} \times \frac{1}{|\mathcal{I}_u(p_i)|} & u_j \in \mathcal{I}_u(p_i) \end{cases} \end{aligned} \tag{5}$$

$$\begin{aligned} P_{m+i,m+j} &= Prob(\, u_j | u_i \,), \qquad i \, < \, n, j \, < \, n \\ &= \begin{cases} 0 & p_j \notin \mathcal{A}_p(p_i) \\ \frac{\gamma}{\alpha+\gamma} \times \frac{sim_p(p_i, p_j)}{\sum_{p_k \in \mathcal{A}_p(p_i)} sim_p(p_i, p_k)} & p_j \in \mathcal{A}_p(p_i) \end{cases} \end{aligned} \tag{6}$$

In the above definition, we do not use the same similarity measurement to quantify the user-item connection since the user content and item content adopt independent lexicons and different representation schemes, which means that it is difficult to compute their similarities (e.g., cosine similarity). Besides, selecting different values for parameters α, β and γ corresponds to assign different importance degrees to semantic effects, social relationships and user behavior sequential patterns, which depends on the datasets. In the experiments, we will show that setting the same values for the three parameters, i.e., $\alpha = \beta = \gamma = 1 / 3$, can lead to the best recommendation results on the two testing datasets.

HUI Network update. Assume at timestamp t, the current HUI network $G_{mix,t} = (\mathcal{V}_t, \varepsilon_t) = (\mathcal{U}_t, \varepsilon_{uu,t}, \mathcal{P}_t, \varepsilon_{pp,t}, \varepsilon_{up,t})$ contains the user node set $\mathcal{U}_t$, item node set $\mathcal{P}_t$ and their related edge sets $\varepsilon_{uu,t}$, $\varepsilon_{pp,t}$ and $\varepsilon_{up,t}$. Due to the evolving of the network, $\mathcal{U}_t$ and $\mathcal{P}_t$ will contain the sets of the newly attached nodes, denoted as $\Delta\mathcal{U}_t$ and $\Delta\mathcal{P}_t$ respectively, while there exists another subsets of $\mathcal{U}_t$ and $\mathcal{P}_t$ containing the nodes that have changed at the current timestamp, which are denoted as $\Theta\mathcal{U}_t$ and $\Theta\mathcal{P}_t$. Similarly, subsets of $\varepsilon_{uu,t}$, $\varepsilon_{pp,t}$ and $\varepsilon_{up,t}$ contain the newly attached edges, separately denoted as $\Delta\varepsilon_{uu,t}$, $\Delta\varepsilon_{pp,t}$ and $\Delta\varepsilon_{up,t}$, while the subsets of changed edges within $\varepsilon_{uu,t}$, $\varepsilon_{pp,t}$ and $\varepsilon_{up,t}$ at current timestamp are denoted as $\Theta\varepsilon_{uu,t}$, $\Theta\varepsilon_{pp,t}$ and $\Theta\varepsilon_{up,t}$ separately.

It is necessary to update the HUI network from timestamp $t-1$ to timestamp t according to the evolving nodes $(\Delta\mathcal{U}_t \cup \Theta\mathcal{U}_t,\ \Delta\mathcal{P}_t \cup \Theta\mathcal{P}_t)$ and edges $(\Delta\varepsilon_{uu,t} \cup \Theta\varepsilon_{uu,t},\ \Delta\varepsilon_{pp,t} \cup \Theta\varepsilon_{pp,t},\ \Delta\varepsilon_{up,t} \cup \Theta\varepsilon_{up,t})$. This can be easily achieved by updating the two types of nodes and three types of edges in HUI network. For instance, u new user nodes and e new user-user edges are added to the HUI network and their similarities of user social links are computed among related nodes. Accordingly, the transition probability matrix P can be easily updated.

The active nodes at timestamp t (denoted as $\tilde{\mathcal{V}}_t$) are defined as the union of the evolving nodes $(\Delta\mathcal{U}_t \cup \Theta\mathcal{U}_t,\ \Delta\mathcal{P}_t \cup \Theta\mathcal{P}_t)$ and the nodes incident upon the evolving edges $(\Delta\varepsilon_{uu,t} \cup \Theta\varepsilon_{uu,t},\ \Delta\varepsilon_{pp,t} \cup \Theta\varepsilon_{pp,t},\ \Delta\varepsilon_{up,t} \cup \Theta\varepsilon_{up,t})$. That is

$$\begin{aligned}\tilde{\mathcal{V}}_t = {} & \Delta\mathcal{U}_t \cup \Theta\mathcal{U}_t \cup \Delta\mathcal{P}_t \cup \Theta\mathcal{P}_t \\ & \cup \{u_i | \exists e_u \in \Delta\varepsilon_{\mathrm{uu,t}} \cup \Theta\varepsilon_{\mathrm{uu,t}}, e_u = (u_i, u_j)\} \\ & \cup \{p_i | \exists e_p \in \Delta\varepsilon_{\mathrm{pp,t}} \cup \Theta\varepsilon_{\mathrm{pp,t}}, e_p = (p_i, p_j)\} \\ & \cup \{u_k, p_f | \exists e_{up} \in \Delta\varepsilon_{\mathrm{up,t}} \cup \Theta\varepsilon_{\mathrm{up,t}}, e_{up} = (u_k, p_f)\}\end{aligned} \tag{7}$$

The underlying principle of the network constructing and updating process can be analogous to the case of adopting sliding window schema to manage continuous data streams. The construction process of HUI network is based on the historical records, and the updating course of the network can be conducted only within several timestamps like a certain length sliding window. The worst case happens only when all nodes $\{v_i | v_i \in \mathcal{V}_t\}$ have changed within timestamp t. In such case, the retraining process of the whole HUI network is inevitable.

5.3.2 Heterogeneous User-Item Network Embedding

Inspired by DeepWalk [41] and the idea of modelling document [252, 253] in natural language processing, our model contains two main stages, heterogeneous random walk and model learning process. In this section, we will illustrate each stage in details.

Heterogeneous random walk. According to the previous work [254], random walk can be used to define proximity, but it is only limited to the network with one type of nodes and links. In order to extend random walk into heterogeneous networks with multiple nodes and various types of edges, the transition probability matrix P defined in Section 5.3.1 is introduced to treat different kinds of nodes and edges equally.

Given the length of random walk as h and the total number of random walks as l, the starting step will be performed at each of the active node $\tilde{\mathcal{V}}_t$ at timestamp t. Based on the updated transition probability matrix P, the heterogeneous random walk will generate possible route sequesces for active nodes, denoted as $S = \{s_1, s_2, \dots, s_{|\tilde{\mathcal{V}}_t|}\}$. The detailed procedure is proceeded as follows.

1) When the walker is in the user node u_i, it will jump to either one of its associated item nodes $p_j \in \mathcal{I}_p(u_i)$ or one of its adjacent user nodes $u_j \in \mathcal{A}_u(u_i)$, with probabilities accessed from the transition probability matrix P.
2) When the walker is in the item node p_i, it will jump to either one of its associated user nodes $u_j \in \mathcal{I}_u(p_i)$ or one of its adjacent item nodes $p_j \in \mathcal{A}_p(p_i)$, with probabilities accessed from the transition probability matrix P.

Such hop process is repeated until finishing h hops, which is taken as a single random walk. And since the total number of l random walks are performed, the whole procedure generates $l \times h$ hops. The encountered combination of nodes for node v_i during these hops is denoted as the possible route sequence of v_i. In random walk, the jump cannot go directly back to the previous node, for example, v_i to v_j back to v_i is not allowed, which in order to avoid getting stuck in some hops with high probabilities in P. Algorithm 1 summarizes the procedure of the heterogeneous random walk for the active nodes at each timestamp.

Algorithm 1: Heterogeneous Random Walk.

```
Input: Transition probability $P$, active node set $\widetilde{V}_t$, number of
       random walks $l$, length of random walk $h$.
Output: The set $S$ of possible route sequence for each node in
        active node set $\widetilde{V}_t$
for $\forall v_i \in \widetilde{V}_t$ do
    for $i = 1$ to $l$ do
        Perform an h-hop random walk starting at $v_i$ using the
        transition probability matrix $P$
    end
    Possibile route sequence $s_i$ for node $v_i$
end
```

Incremental network embedding learning. During the model learning process, the heterogeneous random walk will be performed on the initial HUI network $G_{mix} = (\mathcal{V}, \varepsilon)$ firstly, and it results in a set of possible route sequences $S = \{s_1, s_2, \dots, s_{|\tilde{V}_t|}\}$, where each sequence can be denoted as $s = \{v_1, v_2, \dots, v_{|s|}\}$. DeepWalk treats each route sequence s as a word sequence by regarding nodes as words. Then by introducing Skip-Gram, a widely used word representation learning algorithm, DeepWalk is able to learn node representations from the sequence set S. Similarly, our model also adopts Skip-Gram to learn the representation of each node. More specifically, when given a node route sequence $s = \{v_1, v_2, \dots, v_{|s|}\}$, each node v_i has $\{v_{\mathrm{i}-\mathrm{T}}, \dots, v_{i+T}\} \setminus \{v_i\}$, as its local context nodes. Thus, DGE model learns node representations by maximizing the average log probability of predicting context nodes:

$$\mathcal{L}(s) = \frac{1}{|s|}\sum_{i=1}^{|s|}\sum_{i-|T|\le j\le i+|T|} \log \Pr\left(v_j \middle| v_i\right) \tag{8}$$

where v_j is the context node of the node v_i, and the probability $\Pr\left(v_j \middle| v_i\right)$ is defined using the softmax function:

$$\Pr\left(v_j \middle| v_i\right) = \frac{\exp\left(\boldsymbol{v}'_j \cdot \boldsymbol{v}_i\right)}{\sum_{v' \in \mathcal{V}} \exp\left(\boldsymbol{v}' \cdot \boldsymbol{v}_i\right)} \tag{9}$$

where $\boldsymbol{v}_i$ is the representation of the center node v_i and $\boldsymbol{v}_j'$ is the context representation of its context node v_j. Then subsequently, during incremental learning process at each timestamp $t > 1$, the heterogeneous random walk procedure and Skip-Gram will be proceeded on active node set $\tilde{\mathcal{V}}_t$ and their related edges.

Given that calculating Eq. (9) directly is not feasible and will lead expensive computing cost in practical implementation. Therefore, a computational efficient approximation of the full softmax called hierarchical softmax [250], is introduced to solve this problem. The hierarchical softmax uses a binary tree representation for every context node $v_j \in \mathcal{V}$ as its leaves, and each tree node is explicitly associated with an embedding vector θ for computing the relative probability to take the branch. Each leave can be reached by an appropriate path from the root of the tree. In this way, instead of evaluating all the $|\mathcal{V}|$ nodes, it needs to evaluate only about $\log(|\mathcal{V}|)$ nodes to obtain the probability distribution. More precisely, given the representation $\boldsymbol{v}_i$ of node v_i for target context v_j, let $L(v_j)$ be the length of its corresponding path, and let $b_n^{v_j} = 0$ when the path to v_j takes the left branch at the n-th layer and $b_n^{v_j} = 1$ otherwise. Then, the hierarchical softmax defines $\Pr(v_j|v_i)$ as follows:

$$\Pr(v_j|v_i) = \prod_{n=2}^{L(v_j)}([\sigma(\boldsymbol{v}_i^T\theta_{n-1}^{v_j})]^{1-b_n^{v_j}} \cdot [1-\sigma(\boldsymbol{v}_i^T\theta_{n-1}^{v_j})]^{b_n^{v_j}}) \tag{10}$$

where $\sigma(z) = \frac{1}{1+exp(-z)}$. All parameters are trained by using the Stochastic Gradient Descent method. During the training, the algorithm iterates over the nodes through all possible route sequences, and at each time, a target node v_j with its context window is used for update. After computing the hierarchical softmax according to Eq. (10) , the error gradient is obtained via backpropagation and we use the gradient to update the parameters in our model. To derive how θ is updated at each time step, the gradient for $\theta_{n-1}^{v_j}$ is computed as follows:

$$\frac{\partial\mathcal{L}(v_j,n)}{\partial\theta_{n-1}^{v_j}} = [1 - b_n^{v_j} - \sigma(\boldsymbol{v}_i^T\theta_{n-1}^{v_j})]\boldsymbol{v}_i \tag{11}$$

In this way, $\theta_{n-1}^{v_j}$ can be updated as:

$$\theta_{n-1}^{v_j} \leftarrow \theta_{n-1}^{v_j} + \eta[1 - b_n^{v_j} - \sigma(\boldsymbol{v}_i^T\theta_{n-1}^{v_j})]\boldsymbol{v}_i \tag{12}$$

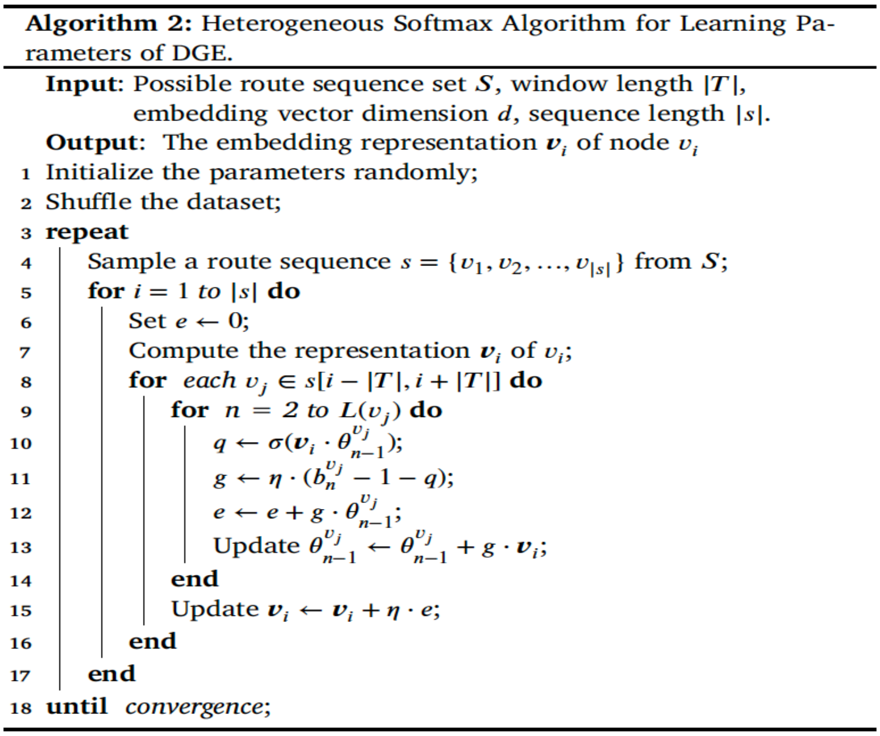
where η denotes the learning rate. To derive how the representation of the center node is updated, the fradient for $\boldsymbol{v}_i$ is computed as follows:

$$\frac{\partial \mathcal{L}(v_j,n)}{\partial \boldsymbol{v}_i} = [1 - b_n^{v_j} - \sigma(\boldsymbol{v}_i^T \theta_{n-1}^{v_j})]\theta_{n-1}^{v_j} \tag{13}$$

With this derivative, an embedding vector $\boldsymbol{v}_i$ in the context of node v_j can be updated as follows:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \eta \sum_{n=2}^{L(v_j)} \frac{\partial \mathcal{L}(v_j,n)}{\partial \boldsymbol{v}_i} \tag{14}$$

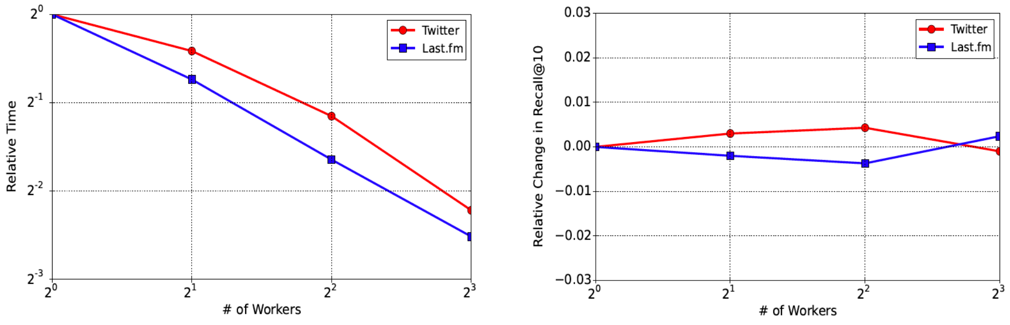
In Algorithm 2, we summarize the learning process using hierarchical softmax for proposed DGE model. The algorithm iterates through all possible route sequences and updates the embedding vectors until the procedure converges. In each iteration, given a current node, the algorithm first obtains its embedding vectors and computes its context embedding vector. Based on the derivative above, the binary tree in hierarchical sampling is updated followed by the embedding vector. Given the vector size of d, the leaf nodes number $|V|$, the sequence length $|s|$ within one iteration and window length $|T|$, then the time complexity for an iteration is $O(d \cdot |T| \cdot |s| \cdot log\ (|V|))$.



Algorithm 2: Heterogeneous Softmax Algorithm for Learning Parameters of DGE.

```
Input: Possible route sequence set $S$, window length $|T|$,
       embedding vector dimension $d$, sequence length $|s|$.
Output: The embedding representation $\boldsymbol{v}_i$ of node $v_i$
Initialize the parameters randomly;
Shuffle the dataset;
repeat
    Sample a route sequence $s = \{v_1, v_2, \ldots, v_{|s|}\}$ from $S$;
    for $i = 1$ to $|s|$ do
        Set $e \leftarrow 0$;
        Compute the representation $\boldsymbol{v}_i$ of $v_i$;
        for each $v_j \in s[i - |T|, i + |T|]$ do
            for $n = 2$ to $L(v_j)$ do
                $q \leftarrow \sigma(\boldsymbol{v}_i \cdot \theta_{n-1}^{v_j})$;
                $g \leftarrow \eta \cdot (b_n^{v_j} - 1 - q)$;
                $e \leftarrow e + g \cdot \theta_{n-1}^{v_j}$;
                Update $\theta_{n-1}^{v_j} \leftarrow \theta_{n-1}^{v_j} + g \cdot \boldsymbol{v}_i$;
            end
            Update $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \eta \cdot e$;
        end
    end
until convergence;
```

Parallelizability. For real-world social networks, the frequency distribution of vertices in random walks follows a power law which results in a long tail of infrequent vertices [41]. Therefore, the updates of vertices' representation will be sparse in nature. Based on this, we adopt the lock-free solutions in the work [255] to parallelize asynchronous stochastic gradient descent (ASGD). Given that our updates are sparse and we do not acquire a lock to access the model shared parameters, ASGD will achieve an optimal rate of convergence. Figure 5.3 presents the effects of parallelizing DGE model with multiple threads. It shows the speed up in processing Twitter and Last.fm datasets is consistent as we increase the number of workers to 8 (Figure 5.3(a)). It also shows that there is no loss of predictive performance relative to the running DGE serially (Figure 5.3(b)).



(a) Running Time

(b) Performance

Figure 5.3: Effects of parallelizing DGE model.

5.3.3 Recommendation Using DGE

Once we have learnt the model parameters, recommendations can be made by utilizing the embeddings for each vertex in the social network. In this section, we propose the top-K recommendation algorithms for a user to select potential friends and interested items respectively.

Recommending top-k friends. This task is to recommend top-k friends that a user u would like to follow in the social network. More precisely, given a target user $u_i \in \mathcal{U}$ with the query time t, for each user node u_j who has not been connected with u_i, we compute its ranking score as in Eq. (15) , and then select the k ones with the highest ranking scores as recommendations.

$$S(u_i, u_j, t) = \sum_{k=1}^{D} x_{ik} \cdot y_{jk} \tag{15}$$

where $u_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $u_j = (y_{j1}, y_{j2}, \dots, y_{jD})$, D is the dimension of the representation vector.

The straightforward method of generating the top-k friends needs to compute the ranking scores for almost all users according to Eq. (15) , which is computationally inefficient, especially when the number of users becomes large. To speed up the process of producing recommendations, we extend the Threshold-based Algorithm (TA) [251], which is capable of finding the top-k results by examining the minimum number of users.

We first pre-compute the ordered lists of users, where each list corresponds to a dimension of the user's representation vector $u_w = (y_{w1}, y_{w2}, \dots, y_{wD})$. So, we could get D lists of sorted users, $L_n, n \in \{1, 2, \dots, D\}$, where users in each list L_n are sorted according to y_{wn}. Given a query $q = (u_i, t)$, we run Algorithm 3 to compute the top-k users from the D sorted lists and return them in the priority list L. As shown in Algorithm 3, we first maintain a priority list PL for the D lists where the priority of a list L_n is the ranking score $S(u_i, u_w, t)$ of the first user w in L_n (Lines 2–6). In each iteration, we select the most promising user (i.e., the first user) from the list that has the highest priority in PL and add it to the resulting list L (Lines 9–16). When the size of L is no less than k, we will examine the k-th user in the resulting list L. If the ranking score of the k-th user is higher than the threshold score T_s, the algorithm terminates early without checking any subsequent users (Lines 18 – 20). Otherwise, the k-th user w' in L is replaced by the current user w if w's ranking score is higher than that of w' (Lines 21 – 24). At the end of each iteration, we update the priority of the current list as well as the threshold score (lines 27–32).

Eq. (16) illustrates the computation of the threshold score T_s, which is obtained by aggregating the maximum y_{wn} represented by the first user in each list L_n. Consequently, it is the maximum possible ranking score that can be achieved by the remaining unexamined items. Hence, if the ranking score of the k-th user in the resulting list L is higher than the threshold score, L can be returned immediately because no remaining user will have a higher ranking score than the k-th user.

$$T_S = \sum_{n=1}^{D} x_{in} \cdot max_{w \in L_n} y_{wn} \tag{16}$$

Recommending top-K items. This task is to recommend top-k items that a user u would like to be interested in. From the fresh-based perspective, the most recent items that a user shows an interest on could better reflect his/her current preference and they should contribute more in the computation of the recommendations [256]. Thus, we use the

Algorithm 3: Threshold-based algorithm.

```
Input: A query $q = (u_i, t)$, ranked lists $(L_1, \dots, L_D)$.
Output: List L with all the k highest ranked users.
Initialize priority lists PL, L and the threshold score $T_s$;
for $n = 1$ to $D$ do
    $w = L_n.getfirst()$;
    Compute $S(u_i, u_w, t)$ according to Eq. (15);
    $PL.insert(n, S(u_i, u_w, t))$;
end
Compute $T_s$ according to Eq. (16);
while true do
    $nextListToCheck = PL.getfirst()$;
    $PL.removefirst()$;
    $w = L_{nextListToCheck}.getfirst()$;
    $L_{nextListToCheck}.removefirst()$;
    if $w \notin L$ then
        if $L.size() < k$ then
            $L.insert(w, S(u_i, u_w, t))$;
        else
            $w' = L.get(k)$;
            if $S(u_i, u'_w, t) > T_s$ then
                break;
            end
            if $S(u_i, u'_w, t) < S(u_i, u_w, t)$ then
                $L.remove(k)$;
                $L.insert(w, S(u_i, u_w, t))$;
            end
        end
    end
    if $L_{nextListToCheck}.hasMore()$ then
        $w = L_{nextListToCheck}.getfirst()$;
        Compute $S(u_i, u_w, t)$ according to Eq. (15);
        $PL.insert(nextListToCheck, S(u_i, u_w, t))$;
        Compute $T_s$ according to Eq. (16);
    else
        break;
    end
end
```

exponential function $f(t_1, k) = e^{-k(t-t_1)}$, where t_1 is the timestamp of item and k is employed to adjust the decay rate, to reflect the freshness of items. According to this, the ranking score of recommendations can be computed as follows:

$$S(u_i, p_j, t) = f(t_1, k) \sum_{n=1}^{D} x_{in} \cdot z_{jn} \tag{17}$$

where $u_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ is the representation of target user and $p_j = (z_{j1}, z_{j2}, \ldots, z_{jD})$ is the representation of a candidate item. Once the newly arrived items have been settled in ordered candidate item lists, as time goes on, the decay value of all items freshness will be the same as $e^{-k \cdot \Delta t}$, Δt is the time interval, without influencing the order of them. Thus this allows us to leverage TA algorithm for retrieving and recommending items the same as friends recommendation.

5.3.4 Framework Extensibility

Here we discuss the extendability of our proposed framework, which we believe may be of interest.

Multiple social networks. Intuitively, our DGE model could be extended to multiple sources for a user who is affiliated to them. For instance, if a user has an account in Facebook and also in Twitter, then both kinds of social sources can bring valuable and multiple information which could assist to improve the recommendation performance. Nowadays, some approaches have been designed to apply the recommendation strategies to support users operating in multiple social sites [257]. Other researches concentrate on the construction of a global user profile with the integration of multiple information from different sources [258]. Zhang et al. [259] proposed an unsupervised network alignment framework (UNICOAT) to address the partial co-alignment problem and discover the potential links between users and between locations for multiple sources. In Jia et al. [260], the authors introduced a deep learning-based approach integrating fusing social networks to predict volunteerism tendency. More specifically, it learns predictive models independently for multiple sources with the input of the hyper representations of features extracted from multiple sites separately, and then weighted sum of these models into the final predictive model.

Inspired by Jia et al. [260], we can build our heterogeneous network separately for multiple sources, and learn the node representations separately. Then, a weighted average can be used to form the final representation of each node. Specifically, let $\boldsymbol{v}_i^k$ be the learned representation of user/item node v_i in the k-th source, and thus the final representation of node v_i can be derived as

$$\boldsymbol{v}_i = \sum_{k \in K} \lambda^k \boldsymbol{v}_i^k \tag{18}$$

where K represents the source set, and λ^k represents the weight of different sources in k. Thus the objective function of Eq. (8) can be redefined as

$$\mathcal{L}(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-|T| \leq j \leq i+|T|} \log \Pr(v_j | v_i) + \eta R \tag{19}$$

where R is the regularization term defined as

$$R = -\sum_{i \in |s|} \sum_{k=1}^{K} \lambda_i^k \left\| \boldsymbol{v}_i^k - \boldsymbol{v}_i \right\|_2^2 \tag{20}$$

and η is a parameter used to control the weight of the regularization term. By maximizing this objective function, different multiple sources can be collaborated to learn the robust node representations.

5.4 Experimental Setup

5.4.1 Dataset Description

For experimental study, we evaluate the proposed DGE model on two kinds of real-world datasets: Twitter and Last.fm. We downloaded the Twitter dataset from Twitter API[21], which includes users and their posts. We collected the Last.fm dataset through Last.fm API[22], which contains users and artists. The statistics of each dataset is summarized in Table 5.2. For both datasets, the user-user links are constructed from bi-directional friendships between social network users, user-item links are constructed from the user listening or posting behaviour, and item-item link are constructed if the two artists share the same tag or the two posts have the same hashtag.

5.4.2 Comparative Approaches

We compared the proposed approach with four state-of-the-art methods:

[21] https://dev.twitter.com/docs

[22] http://www.last.fm/api/

Table 5.2: Some statistics of the datasets.

Datasets	#Users	#Items	#User-user links	#User-item links	#Item-item links
Twitter	87,287	7,855,830	537,251	86,414,130	38,493,509
Last.fm	50,000	11,215	291,805	14,358,010	2,264,562

- **Popular in Neighborhood (PN).** Given a personalized graph G_u, the Popular in Neighbourhood (PN) algorithm ranks the candidate items based on the number of actions all users performed on them within the context of G_u. Hence, the algorithm ranks the candidate entities that appear in G_u based on how popular they are.
- **Weighted Regularized Matrix Factorization (WRMF).** This is a state-of-the-art offline matrix factorization model for item prediction introduced by Hu et al. [261]. Their method outperforms neighbourhood based (item-item) models in the task of item prediction for implicit feedback datasets. The model is computed in batch mode, assuming that the whole stream is stored and available for training.
- **Stream Ranking Matrix Factorization (RMFX).** It is proposed in Ernesto's recent work [209], which can achieve partly online and much quicker updates of matrix factorization for item prediction.
- **DeepWalk** [41]. It uses local information obtained from truncated random walks to learn latent representations of nodes in a graph. It is an extended application of word2vec-based model.
- **LINE-2nd.** We also adopt the LINE [43] second-order (2nd) version in order to make the comparison to our proposed context embedding model. According to Xie et al. [262] , the author builds multiple graphs incorporating geographical influence, temporal cyclic effect and semantic effect. Similarly, in this paper, we build multiple graphs integrating user relationships, user-item interactions and semantic influence into one model.

WRMF setup is as follows: $\lambda_{WRMF} = 0.015, C = 1, epochs = 15$, which corresponds to a regularization parameter, a confidence weight that is put on positive observations, and to the number of passes over observed data, respectively. For RMFX, we set regularization constants $\lambda_{\mathrm{RMFX}} = 0.1$, learning rate $\eta_0 = 0.1$, and a learning rate schedule $\alpha = 1$, and find that the setting gives good performance. Moreover, the number of iterations is set to the size of the reservoir. For all the embedding algorithms (DeepWalk, LINE, and our model), the embedding dimensionality is set to 128. We tried dimensionalities in the range [16, 248] and found that 128 generally gives the best results. Context window length is set to 8, walk length is set to 40, walks per vertex is set to 30.

5.4.3 Evaluation Criteria

Given a dataset $\mathcal{D}$ which includes user profile and item profile, we first rank them according to their tweets timestamp in Twitter dataset or listening timestamp in Last.fm dataset. Then we use the 80-th percentile as the cut-off point so that user-item interaction behaviors before this point will be used for training and the rest are for testing. In the training dataset, we choose the last 10% records as the validation data to tune the model hyper-parameters such as the dimension of the latent space. According to the above dividing strategies, we split the dataset $\mathcal{D}$ into the training set $\mathcal{D}_{train}$ and the test set $\mathcal{D}_{test}$.

Since we are interested in measuring top-k recommendation instead of rating prediction, we measure the quality by looking at the $Recall@K$ metric, which is widely used for evaluating top-k recommender systems [245, 261]. We show the performance when $k = \{1, 5, 10\}$, as a greater value of k is usually ignored for a typical top-k recommendation task [263].

In our recommender system setting, the recall metric is defined as follows:

- $Recall@K$ (also known as hit rate) is the proportion of relevant items/users found in the top-k recommendations. A larger recall value indicates that the system is able to recommend more satisfactory items or users, leading to a better performance. Formally, we define $hit@K$ for a single test case as either the value 1, if the test item or user appears in the top-k results, or the value 0, if otherwise. The overall $Recall@K$ is computed by averaging over all test cases:

$$Recall@K := \frac{\#hit@K}{|\mathcal{D}_{test}|} \tag{21}$$

where $\#hit@K$ denotes the number of hits in the whole test set.

Another evaluation metric we used is average reciprocal hit-rank (ARHR) [247] which in our setting we define as follows:

- ARHR (average reciprocal hit-rank) is a weighted version of hit rate that rewards each hit based on where it occurs in the top-k list. If we take top-k item recommendation as an example, for a target user u, let h be the number of the true interested items in the recommended top-k list, and n_{u} be the number of u's true interested items in the test dataset. The ARHR is to measure the effectiveness of ranking for each target user. When $p_1, p_2, \dots, p_h$ are the positions of the true interested items in the recommended top-k items, the ARHR for u can be defined as

$$ARHR := \frac{1}{n_u} \sum_{i=1}^{h} \frac{1}{p_i} \tag{22}$$

As the true interested items appear with high ranks in the recommended items, this measure becomes larger.

In top-k friend recommendations, let h represents the number of the true friends in the recommended top-k list, and n_{u} represents the number of u's true friends in the test dataset, for each target user u, the average reciprocal hit-rank can be computed similarly.

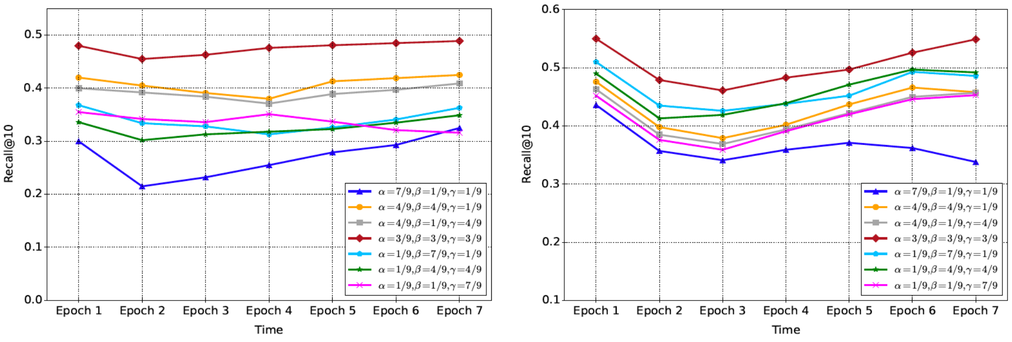
5.5 Experimental Results

5.5.1 Sentivity to Parameters

In this section, we first analyze the performance sensitivity to the three trade-off parameters, which control the relative importance of user behavior sequential patterns, social relationships and semantic effects respectively. Additionally, the performance sensitivity to the random walk parameters is also analyzed.

Trade-off parameters. To investigate how the performance of the DGE model is effected by the relative importance of user behaviour sequential patterns, social relationship and semantic effect, we run the method using various trade-off parameters. The performance is evaluated in terms of the $Recall@10$ value obtained at each timestamp. Figure 5.4 plots the $Recall$ score on the two datasets using different trade-off parameters. From Figure 5.4, we can see that, on the two testing datasets, the worst results are obtained when the relative importance of user behaviour sequential patterns is set very large while the relative importance of social relationship and semantic effects are set very small, i.e. $\alpha = 7 / 9, \beta = 1 / 9, \gamma = 1 / 9$. This is because in the heterogeneous social network, the content information of items and the influence among user-relationships encoded in the edges is the essential motivation to attract the user to click related items. Especially, in Last.fm dataset, this kind of settings causes a noticeable drop of the Recall value at the last three timestamps. The main reason may be that during that time period the users constructed friendship more widely than before.

On the other hand, when setting the relative importance of user behaviour sequence patterns, social relationships and semantic effects to the same level as $\alpha = 3 / 9, \beta = 3 / 9$ and $\gamma = 3 / 9$, the best Recall value is obtained with at least 0.1 improvement, which is very significant. Therefore, in all experiments, except stated otherwise, the relative importance of these three aspects are set to the same level. The performance sensitivity to the trade-off parameters provides a strong evidence of integrating multiple information in the recommender system.



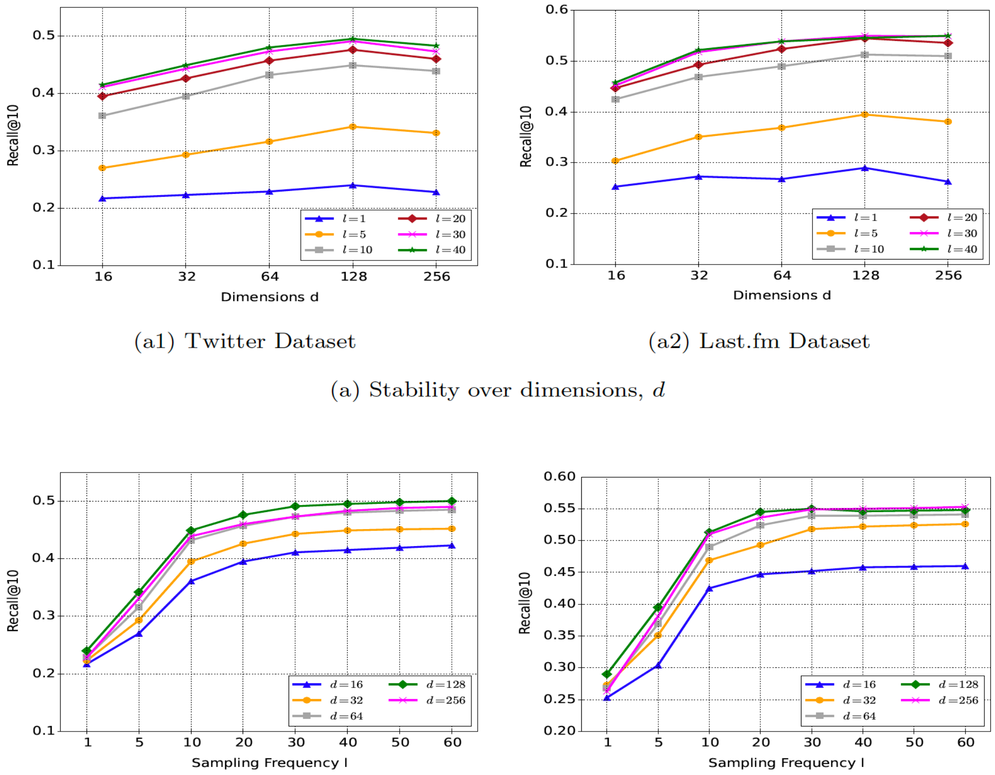
(a) Twitter

(b) Last.fm

Figure 5.4: Sensitivity to trade-off parameters.

Effect of dimensionality and sampling frequency. Tuning model parameters is critical to the performance of the proposed model. In this experiment, we study the influence of the embedding dimension d and the number of samples l by fixing the window size $|T| = 8$ and the random walk length $h = 40$. We then vary the number of dimensions d and number of walks started per node l to determine their impact on the recommendation performance. The results are shown in Figure 5.5.

From Figure 5.5, similar observations can be made on both datasets. It can be observed that recommendation $Recall$ value of DEG model is not highly sensitive to the dimension d, but still presents a tendency that its recommendation accuracy increases with the increasing number of dimension d holistically, and then it reaches peak when d is around 128. However, DGE is sensitive to the number of samples l, the $Recall$ score varies a lot. First, the performance of DEG increases quickly with the increasing number of l, this is because the model has not achieved convergence. Then, it does not change significantly when the number of samples becomes large enough, since the model DGE has converged. Thus, to achieved a satisfying trade off between effectiveness and efficiency of model training, we set $l = 30$ and $d = 128$ on both datasets.



(a1) Twitter Dataset

(a2) Last.fm Dataset

(a) Stability over dimensions, d

(b1) Twitter Dataset

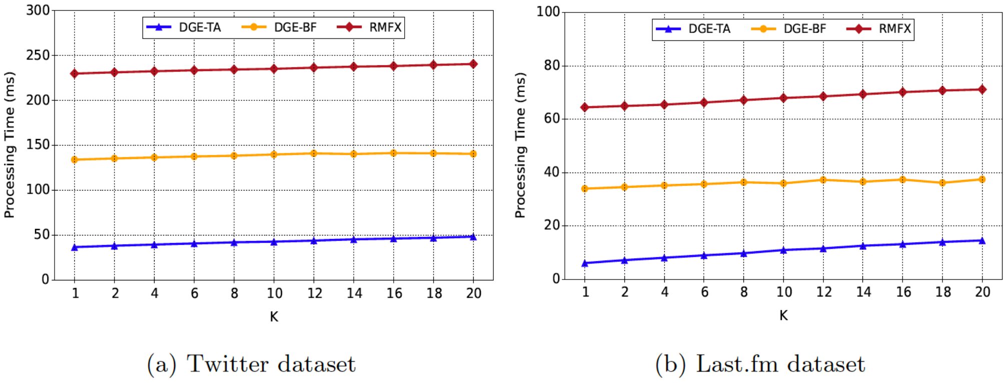
(b2) Last.fm Dataset

(b) Stability over number of samples, l

Figure 5.5: Effect of dimensionality and sampling frequency.

5.5.2 Online Recommendation Efficiency

In this section, we evaluate the online recommendation efficiency with Twitter and Last.fm datasets. We test the average time cost of top-k recommendations on two methods, DGE-TA and DGE-BF which utilize the knowledge learnt by DGE to produce recommendations. DGE-TA uses the proposed TA-based query processing technology described in Section 5.3.3 to produce top-k recommendation results. In DGE-BF, we adopt a brute-based algorithm by scanning all recommendation candidates and computing their ranking scores, to produce top-k recommendations with k highest ranking scores. In addition, we also use the state-of-the-art online recommendation algorithm RMFX to produce top-k recommendations because the other baselines are offline methods and they need to retrain the model during each updating period before recommendation task. Thus here we mainly focus on the online algorithms efficiency comparison.



(a) Twitter dataset

(b) Last.fm dataset

Figure 5.6: Online recommendation efficiency.

Figure 5.6 presents the results of different methods with varying number of k from 1 to 20 for Twitter and Last.fm datasets. On average for k equals to 10, our DGE-TA produces top-10 recommendations for Twitter dataset in 43 ms, and for Last.fm dataset in 11 ms. From the figures we conclude several observations that: 1) DGE-TA outperforms the other two methods significantly in both datasets, which verify that the benefits gained by proposed TA-based query processing techinique; 2) DGE-BF ranks the second in top-k recommendation efficiency in both datasets for RMFX generates ranking scores using large amount of matrix operations while DGE uses inner product of two vectors as shown in Eq. (15) and (17) ; 3) the time cost of DGE-TA method grows with the increasing number of k for DGE-TA needs to scan more recommendation candidates to find top-k recommendations, but DGE-TA is still much efficient than the other two recommendation algorithms since the value of k is normally constrained in a small range; 4) The time cost of each algorithm in Twitter is more expensive than that in Last.fm, showing that if a dataset contains more data, it requires more processing time to produce top-k recommendations.

5.5.3 Recommendation Effectiveness

Table 5.3 and 5.4 summarize the item and friend recommendation performance for the state-of-the-art methods and the DGE model. Generally speaking, it can be shown from these two tables that the $Recall@K$ value grows gradually along with the increasing

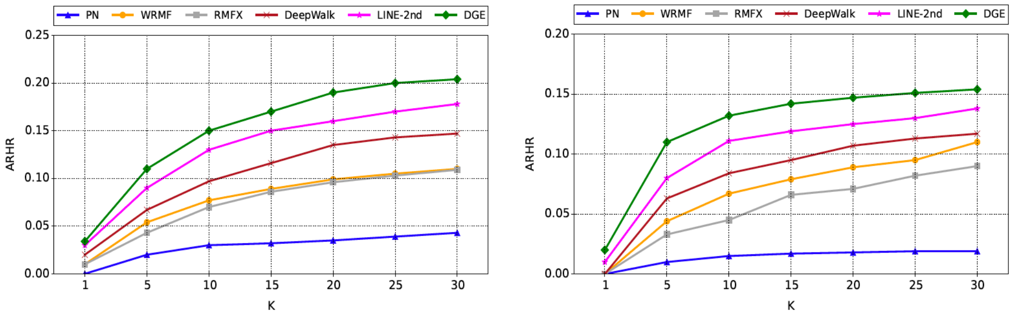
Table 5.3: Top-k items recommendation effectiveness.

Methods	Twitter			Last.fm		
	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10
PN	0.068	0.109	0.189	0.143	0.203	0.261
WRMF	0.156	0.235	0.314	0.227	0.293	0.385
RMFX	0.125	0.201	0.283	0.196	0.278	0.355
DeepWalk	0.194	0.278	0.351	0.265	0.341	0.423
LINE-2nd	0.223	0.297	0.382	0.271	0.357	0.441
DGE	**0.315**	**0.397**	**0.481**	**0.392**	**0.462**	**0.552**

Table 5.4: Top-k friends recommendation effectiveness.

Methods	Twitter			Last.fm		
	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10
PN	0.052	0.085	0.161	0.107	0.157	0.223
WRMF	0.115	0.188	0.275	0.176	0.256	0.329
RMFX	0.104	0.158	0.219	0.139	0.232	0.304
DeepWalk	0.149	0.221	0.310	0.216	0.289	0.367
LINE-2nd	0.176	0.232	0.329	0.216	0.311	0.385
DGE	**0.246**	**0.303**	**0.385**	**0.293**	**0.352**	**0.409**

number of K, and the performance of item recommendation is better than friend recommendation. Besides, we can also observe on both datasets that: Firstly, embedding-based algorithms (DeepWalk, LINE-2nd and DGE) consistently perform better than non-embedding based benchmarks (PN, WRMF and RMFX). For instance, if we consider item recommendation with $Recall@10$ in the Twitter dataset, as shown in Table 5.3, DeepWalk correctly predicts 35.1% of items, while the best performance of non-embedding based algorithms WRMF correctly predicts 31.4% of items. It is because embedding-based algorithms can fully explore the network structure of the given information, which alleviates the issues of sparse and noisy signals. Secondly, among embedding-based algorithms, DeepWalk is only applicable to homogeneous networks, while LINE-2nd and DGE are capable of handling heterogeneous networks, and thus LINE-2nd and DGE perform better than DeepWalk algorithm. Besides, through considering the semantic effects, social relationships and user behaviour sequential patterns as well as their potential relations simultaneously, DGE encapsulates more contextual information, leading to more informative updates and robustness. Therefore, DGE performs better than LINE-2nd method. The incremental training ability makes DGE update process more efficiently and timely, which also contributes to the better performance of the model.



(a) Top-k items recommendation (b) Top-k friends recommendation

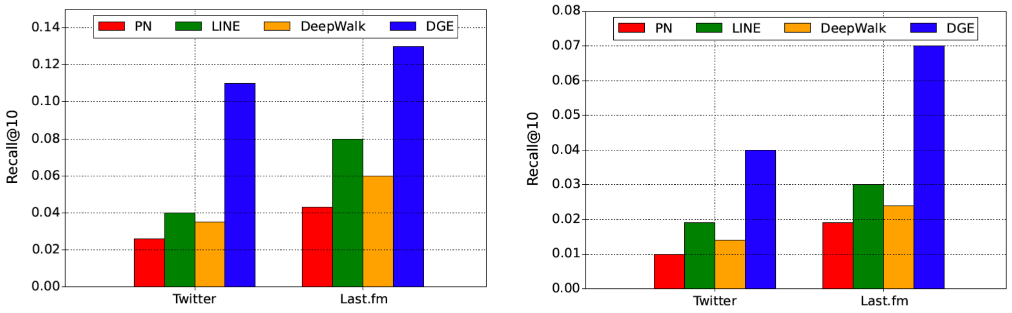
Figure 5.7: Recommendation performance with regard to ARHR.

Figure 5.7 compares the performance of alternative approaches taking average reciprocal hit-rank (ARHR) as metric. During experiments, we vary the number of recommendations K from 1 to 30. As expected, our DGE model performs better with ARHR as well, and LINE-2nd ranks the second place, whsich shows the same orders in Table 5.3 and 5.4 . As can be seen from Figure 5.7(a), when we recommend more items, since we have more chances to answer the true interested items correctly, ARHR grows gradually with increasing number K. The same trends appeared in the friend recommendation task.

5.5.4 Cold Start Problem

In this experiment, we conduct experiments to study the effectiveness of different recommendation algorithms in addressing cold-start issues on the two datasets. For evaluating the top-k items and friends recommendations, the target users who have less than 20 available items and social link information in total are selected. As there are not many interaction records between users and items available for cold-start cases, WRMF and RMFX model which are based on collaborative filtering, are not suitable for cold-start experiments. Thus, we compare our DGE model with other three recommender models that are able to leverage semantic effects and user relationships to recommend cold-start cases.

The experimental results are shown in Figure 5.8, from which we have the following observations: 1) our proposed DGE model still performs best consistently in recommending



(a) Top-k items recommendation

(b) Top-k friends recommendation

Figure 5.8: Recommendations for cold-start cases.

cold-start cases; 2) by comparing the recommendation results in Table 5.3 and 5.4 , the *Recall* value of all recommendation algorithms decreases. For instance, the *Recall* value of DeepWalk rapidly drops from 35.1% to less than 4% for twitter item recommendation, while our model deteriorate slightly. This is because DeepWalk recommends items according to their content information such as hashtags and labels, while our method also considers the content similarities when training models and the potential relationships among all effects. This is to say, our model leverages not only user-item interactions, semantic effects and user relationships, but also the potential links between the features, when recommending cold-start items/users.

5.6 Conclusion

In this paper, a novel dynamic graph-based embedding model, DGE is proposed for real-time social recommendations. DGE jointly captures the temporal semantic effects, social relationships and user behavior sequential patterns in a unified way by embedding the heterogeneous user-item network into a shared low dimensional space for addressing the issues of temporal dynamics, cold start and context awareness in the social recommender system. To capture the semantic effect of different edge types, a transition probability matrix is devised and updated as the social network evolves. For efficiently handling large-scale social media streams, a parallel incremental learning algorithm and an efficient query processing technique are developed to generate top-k recommendations. Our recommendation process is based on the proximity of the related users and items while considering the freshness of the items. Evaluation on two different real-world datasets demonstrated the effectiveness of the proposed approach.

Chapter 6
Community-aware Social Recommendation

In the previous chapter, we presented a dynamic graph-based embedding model (DGE) which considers the semantic effects, social relationships and user behavior sequential patterns for addressing the cold start issue in social recommendations. Despite effectiveness, this model primarily preserves the local structure and content, such as the first- and second-order proximities of nodes. In this chapter, we extend our DGE model to incorporate the global context, namely community information derived from network structure, into graph embedding model for recommendation. Specifically, we explore the possibility of learning global community context and local context among users and/or items in a joint manner, as well as tracking the evolution of network structures over time. We also look into the overlapping communities in heterogeneous networks. Experimental results on several real large-scale datasets show its advantages over other state-of-the-art methods.

6.1 Introduction

Social recommender systems have become a promising research direction with the rapid development of Web 2.0 and smart mobile devices. They are able to cope with the information overload and to assist users in finding information matching their individual preferences. Various recommendation mechanisms are developed by virtue of both semantic information and social relationships.

Among them, collaborative filtering (CF) has been shown to be an effective approach to recommender systems. It makes predictions about user's interests based on preferences of other users. However, CF is generally designed for bipartite graphs which model interactions between users and items and thus cannot be easily applied over complex heterogeneous social networks. Besides, cold start issue becomes even more severe in online settings as the new users and new items will join in constantly over time. Many approaches [67, 176] have been proposed to alleviate this problem, but they are not designed specifically for online environment.

Recently, network representation learning (NRL) has attracted a considerable amount of interest from various domains, with recommender systems being no exception [51, 264]. The popularization of NRL in recommendation can be mainly attributed to the network embedding techniques which learn low-dimensional vertex representation by modelling vertex co-occurrence in individual user's interaction records, thus capturing the semantic relationships among vertices and boosting recommendation accuracy [264]. Cold start issues can be alleviated through mining the structure and relations among existing and newly arrived nodes. Despite these positive results, we argue that NRL for social recommendations still suffers from the following four challenges: (1) Different from widely used homogeneous networks, heterogeneous network which includes different-typed objects and links, is seldom studied but more commonly seen in real world. Besides, online networks often incorporate millions even billions of nodes and edges in real world, which brings more obstacles in dealing with them. (2) Most real-world networks are intrinsically dynamic with addition/deletion of edges and nodes. Meanwhile, similar as network structure, node attributes also change as new content patterns may emerge and outdated content patterns will fade. (3) So far, most previous network representation methods primarily preserve the local structure and content, such as the first- and second-order proximities of nodes, the global community structure, which is one of the most prominent features, is largely ignored. (4) Considering the online environment and frequently changing velocity of social networks, the scalability and updating complexity of learning algorithms should also play a pivotal role and be seriously reckoned. Recent researches only pay attention to several of the abovementioned challenges while still neglect one or more of them [41, 43, 47, 48, 49].

To address the problems raised above, we propose a novel multi-granularity dynamic network embedding (m-DNE) model for online social recommendation. Specifically, we firstly construct a heterogeneous user-item (HUI) network which is incrementally maintained as the social network evolves. Then, a low complexity incremental learning algorithm is applied to embed HUI into low-dimensional representation space. Meanwhile, multi-granularity proximities which include the second-order proximity and the community-aware high-order proximity of nodes, are introduced to learn more informative and robust network representations. Afterwards, an efficient search method and a time-decay mechanism are adopted to conduct recommendation tasks. To the best of our knowledge, it is the first attempt to improve the representation learning by incorporating temporal community structures derived from heterogeneous networks into the dynamic network embedding method. Our experiments show that the proposed approach is superior to all baselines and state-of-the-art methods in social recommendation tasks.

In this chapter, Section 6.2 introduces the related work. In Section 6.3, we define the key concepts and our problem. Sections 6.4, 6.5 and 6.6 present our model. We describe the experimental setup and results in Section 6.7 and Section 6.8 concludes the study.

6.2 Related Work

We have reviewed the literature on Network Representation Learning (NRL) in Chapter 2, Section 2.1.2. Here we summarize the differences of our m-DNE model with some representative NRL models concerning factors dependent on the online social network in Table 6.1.

Table 6.1: Comparison with related work on Network Representation Learning.

	Node Embed	Heterogeneous Network	Community Aware	Dynamic Environment	Model Complexity
DeepWalk [41]	√				$O(\lvert V\rvert \log \lvert V\rvert)$
LINE [43]	√				$O(a\lvert E\rvert)$
Node2Vec [44]	√				$O(\lvert V\rvert \log\lvert V\rvert + \lvert V\rvert a^2)$
Metapath2vec [45]	√	√			$O(a\lvert E\rvert\lvert V\rvert)$
PTE [46]	√	√			$O(a\lvert E\rvert)$
ComE [47]	√		√		$O(\lvert \mathrm{V}\rvert + \lvert E\rvert)$
DANE [48]	√			√	$O(\lvert V\rvert)$
M-NMF [49]	√		√		$O(\lvert V\rvert^2)$
M-DNE (our model)	√	√	√	√	$O(\lvert V\rvert \log \lvert V\rvert)$

6.3 Problem Formulation

In this section, we define the key concepts and present the problem statement of this study before the detailed description of our m-DNE model.

Definition 1 Heterogeneous User-Item (HUI) Network. A heterogeneous user-item network can be represented by $G_{mix} = G_{uu} \cup G_{pp} \cup G_{up}$, which consists of the user-user relationship network $G_{uu} = (\mathcal{U}, \varepsilon_{uu})$, the item-item relationship network $G_{pp} = (\mathcal{P}, \varepsilon_{pp})$ and the user-item interaction network $G_{up} = (\mathcal{U} \cup \mathcal{P}, \varepsilon_{up})$. Among this, $\mathcal{U} = \{u_1, u_2,$

$\ldots, u_m\}$ is the set of users, where u_i is the user profile represented with a three tuple $(uId, \mathcal{L}, \mathcal{D})$, which indicates userID, user social links and a set of items associated with u_i. $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ is the set of items, where p_i is the item profile with a five tuple $(iId, \mathcal{M}, \mathcal{H}, \mathcal{W}, \rho)$, representing itemID, named entity, hashtag/category, content, create time respectively. ε_{uu}, ε_{pp} and ε_{up} are the sets of edges, which indicate different relation types.

Definition 2 Community. A community c is a group of vertices, including both users and items, in G_{mix}, and all vertices can be grouped into $\mathcal{K}$ communities $\mathcal{C} = \{c_1, c_2, \ldots, c_{\mathcal{K}}\}$. The communities can be overlapping, which is to say each vertex $v \in \mathcal{U} \cup \mathcal{P}$, can belong to different c to different degree.

Finally, we formally define the problem investigated in our work. Given a time-stamped heterogeneous user-item network, we aim to provide online social recommendations stated as follows.

Problem 1 (Online Social Recommendation). Given a heterogeneous user-item network G_{mix} at timestamp t and a querying user $u \in \mathcal{U}$, the task is to generate a ranked list of user or item recommendations that u would be interested in.

6.4 Heterogeneous User-Item Network

We adopt the same construction method and updating strategy for heterogeneous user-item network compared with our DGE model in the chapter 5. However, to incorporate user bias in our algorithm, we introduce w_{ij} which denotes the rating score that the user u_i assigns to item p_j, and the matrix blocks P_{up} and P_{pu} in the transition probability matrix P can be defined as follows:

$$P_{i,m+j} = Prob\left(p_j \middle| u_i \right), \qquad i < m, j < n$$

$$= \begin{cases} 0 & p_j \notin \mathcal{I}_p(u_i) \\ \frac{\alpha}{\alpha+\beta} \times \frac{w_{ij}}{|\mathcal{I}_p(u_i)|} & p_j \in \mathcal{I}_p(u_i) \end{cases} \tag{1}$$

$$P_{m+i,j} = Prob\left(u_j \middle| p_i \right), \qquad i < n, j < m$$

$$= \begin{cases} 0 & u_j \notin \mathcal{I}_u(p_i) \\ \frac{\alpha}{\alpha+\gamma} \times \frac{w_{ji}}{|\mathcal{I}_u(p_i)|} & u_j \in \mathcal{I}_u(p_i) \end{cases} \tag{2}$$

Note that w_{ij} has different rating scales. For example, in the movie recommendation case, w_{ij} might correspond to an explicit rating given by user u_i to movie p_j or, in the case of twitter/music recommendation, w_{ij} is implicitly derived from user's interaction patterns, e.g., how many times user u_i has clicked/listened item p_j.

6.5 Multi-granularity Dynamic Network Embedding

Inspired by DeepWalk [41] and the idea of modelling document [252] in natural language processing, our model contains three main stages as shown in Figure 6.1: heterogeneous random walk, community integration and model learning process, based on which, vertex representations will evolve after incremental learning. Given the length of random walk as h and the total number of random walks as l, the starting step will be performed at each of the active node $\tilde{\mathcal{V}}_t$ at timestamp t. Based on the updated transition probability matrix P, the random walk with restart on heterogeneous network proposed by [265] is employed to generate possible route sequences for active nodes, denoted as $S = \{s_1, s_2, \dots, s_{|\tilde{\mathcal{V}}_t|}\}$. In the rest part of this section, we will illustrate the last two stages.

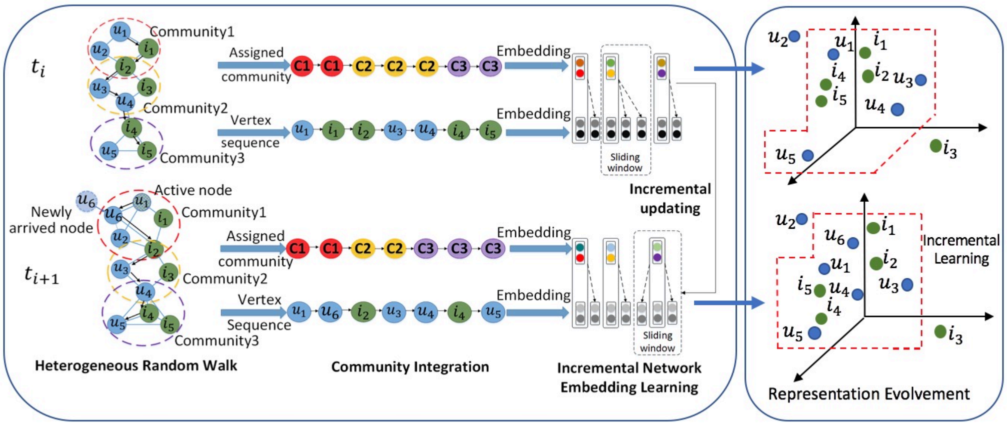


Figure 6.1: The m-DNE model.

6.5.1 Community Integration

As the analogy between words in the text and vertices in walk sequences, we introduce the idea of processing streaming data in topic models to detect overlapping communities in heterogeneous dynamic networks. Before the introduction of community integration procedure, we make two assumptions on heterogeneous random walk sequences, graph vertices and communities as follows: (1) Each vertex in the HUI network can belong to multiple communities with different preferences of $Pr(c|v)$, and each vertex sequence also owns its community distribution. (2) A vertex in a specific sequence belongs to a distinct community, and the community is determined by the community's distribution over sequences $Pr(c|s)$ and the vertex's distribution over communities $Pr(v|c)$.

With the above assumptions and heterogeneous random walk sequences, we can assign community labels to vertices in particular sequence. More specifically, for a vertex v in a sequence s, we compute the conditional probability of a community c with the following equation:

$$Pr(c|v,s) = \frac{Pr(c,v,s)}{\Pr(v,s)} \propto Pr(c,v,s) \tag{3}$$

According to our assumptions, the joint distribution of $Pr(c,v,x)$ can be formalized as

$$Pr(c,v,s) = \Pr(s)Pr(c|s)Pr(v|c) \tag{4}$$

where $Pr(v|c)$ represents the role of v in community c, and $Pr(c|s)$ represents the community distribution in sequence s. From the above 2 equations, we have

$$Pr(c|v,s) \propto Pr(v|c)Pr(c|s) \tag{5}$$

An ordinary way to estimate $Pr(v|c)$ and $Pr(c|s)$ is to use Gibbs Sampling. But it is not suitable for our updating progress. Thus, instead, we extend the Streaming Gibbs Sampling method proposed in [266] to achieve the conditional probability in our environment.

Given a Bayesian model $Pr(x|\zeta)$ with prior $Pr(\zeta)$, and incoming data mini-batches $X^1, X^2, \ldots, X^t$ represented as $X^{1:t}$. Bayesian streaming learning is the process of getting a series of posterior distributions $Pr(\zeta|X^{1:t})$ by the recurrence relation:

$$Pr(\zeta|X^{1:t}) \propto Pr(\zeta|X^{1:t-1})Pr(X^t|\zeta) \tag{6}$$

Therefore, if we fix the community distribution $C^{1:t-1}$ of the previous arrived sequences, then $C^{1:t}$ of the current timestamp can be achieved with $C^{1:t-1}$, and normal Gibbs Sampling on C^t. Therefore, the conditional distributions of $Pr(v|c)$ and $Pr(c|s)$ can be estimated as follows:

$$Pr(v|c) = \frac{N^t(v,c)+\beta_l}{\sum_{v' \in \mathcal{V}} N^t(v',c)+|\mathcal{V}|\beta_l}, \quad Pr(c|s) = \frac{N^t(c,s)+\alpha_l}{\sum_{c' \in \mathcal{C}} N^t(c',s)+\mathcal{K}\alpha_l} \tag{7}$$

where $N^t(v,c)$ is the number of times the vertex v assigned to community c at timestamp

t and $N^t(c, s)$ is the number of vertices in sequence s are assigned to community c at t. Both $N^t(v, c)$ and $N^t(c, s)$ will be updated dynamically as community assignments change, and for different timestamps. β_l and α_l are smoothing factors in Latent Dirichlet Allocation [94]. With estimated $Pr(v|c)$ and $Pr(c|s)$, we assign a discrete community label c for each vertex v in sequence s.

6.5.2 Incremental Network Embedding Learning

To initialize the learning process on HUI network $G_{mix} = (\mathcal{V}, \varepsilon)$, given a certain vertex sequence $s = \{v_1, v_2, \dots, v_{|s|}\}$, for each vertex v_i and its assigned community c_i, we will learn the representations of both vertices and communities by maximizing the average log probability of predicting context vertices using both v_i and c_i as formalized below:

$$\mathcal{L}(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-|W| \le j \le i+|W|} \log \Pr(v_j | v_i, c_i) \tag{8}$$

where v_j is the context node of the node v_i, and the probability $Pr(v_j | v_i, c_i)$ is defined using the softmax function:

$$Pr(v_j | v_i, c_i) = \frac{\exp(\boldsymbol{v}'_j \cdot \overline{\boldsymbol{v}}_i)}{\sum_{v' \in \mathcal{V}} \exp(\boldsymbol{v}' \cdot \overline{\boldsymbol{v}}_i)} \tag{9}$$

where $\boldsymbol{v}'_j$ is the context representation of its context node v_j. $\overline{\boldsymbol{v}}_i$ is the average vector representation of the center node v_i and community label c_i defined as $\overline{\boldsymbol{v}}_i = 1/2(\boldsymbol{v}_i + \boldsymbol{c}_i)$. In such case, the local context and the global community structure can be incorporated to enhance vertex representation learning. Then subsequently, during incremental learning process at each timestamp $t > 1$, the heterogeneous random walk procedure will start with active node set $\tilde{\mathcal{V}}_t$ to obtain possible route sequence set S.

To improve the computational efficiency of Eq. (9), in practical environment, we adopt hierarchical softmax[23], a computational efficient approximation of the full softmax [250]. More precisely, given the average vector representation $\overline{\boldsymbol{v}}_i$ of v_i and c_i for target context v_j, let $L(v_j)$ be the length of its corresponding path, and let $b_n^{v_j} = 0$ when the path to v_j

[23] The hierarchical softmax needs to evaluate only about $log(|V|)$ nodes instead of all the $|V|$ nodes to obtain the probability distribution.

takes the left branch at the n-th layer and $b_n^{v_j} = 1$ otherwise. Then, the hierarchical softmax defines $Pr(v_j|v_i, c_i)$ as follows:

$$\Pr(v_j|v_i, c_i) = \prod_{n=2}^{L(v_j)}([\sigma(\bar{\boldsymbol{v}}_i^T \theta_{n-1}^{v_j})]^{1-b_n^{v_j}} \cdot [1-\sigma(\bar{\boldsymbol{v}}_i^T \theta_{n-1}^{v_j})]^{b_n^{v_j}}) \tag{10}$$

where $\sigma(z) = \frac{1}{1+ exp(-z)}$. All parameters are trained by using the Stochastic Gradient Descent method. To derive how θ is updated at each time step, the gradient for $\theta_{n-1}^{v_j}$ is computed as follows:

$$\frac{\partial \mathcal{L}(v_j, n)}{\partial \theta_{n-1}^{v_j}} = [1 - b_n^{v_j} - \sigma(\bar{\boldsymbol{v}}_i^T \theta_{n-1}^{v_j})]\bar{\boldsymbol{v}}_i \tag{11}$$

To derive how the context embedding vectors are updated, the gradient for $\bar{\boldsymbol{v}}_i$ is computed as follows:

$$\frac{\partial \mathcal{L}(v_j, n)}{\partial \bar{\boldsymbol{v}}_i} = [1 - b_n^{v_j} - \sigma(\bar{\boldsymbol{v}}_i^T \theta_{n-1}^{v_j})]\theta_{n-1}^{v_j} \tag{12}$$

With this derivative, an embedding vector $\boldsymbol{v}_i$ and $\boldsymbol{c}_i$ in the context of node v_j can be updated as follows:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \eta \sum_{n=2}^{L(v_j)} \frac{\partial \mathcal{L}(v_j, n)}{\partial \bar{\boldsymbol{v}}_i}, \qquad \boldsymbol{c}_i \leftarrow \boldsymbol{c}_i + \eta \sum_{n=2}^{L(v_j)} \frac{\partial \mathcal{L}(v_j, n)}{\partial \bar{\boldsymbol{v}}_i} \tag{13}$$

where η denotes the learning rate.

6.6 Recommendation Using m-DNE

Recommendation procedure can be performed after obtaining the embeddings for each vertex. To recommend top-k friends to a user $u_i \in \mathcal{U}$ with D dimensional representation vector of $\vec{u}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and query time t, we compute the ranking score for user node u_j which does not have a direct link with u_i through the inner product of $\vec{u}_i$ and $\vec{u}_j$. Similar procedure can be found when recommending top-k items. Except that, to consider the freshness of the items such as tweets, we bring in the time decay function defined as $f(t_j, \lambda) = e^{-\lambda(t-t_j)}$, where t_j is the publication timestamp of item p_j and λ is employed to adjust the decay rate. Thus, the ranking score of item p_j can be obtained as follows: $S(u_i, p_j, t) = \vec{u}_i \cdot \vec{p}_j = f(t_j, \lambda) \sum_{n=1}^{D} x_{in} \cdot z_{jn}$. For computational efficiency, we adopt the Threshold-based Algorithm (TA) [251], which is capable of finding the top-k results by examining the minimum number of users/items.

6.7 Experiments

6.7.1 Experimental Setup

Dataset Description. For experimental study, we evaluate the proposed m-DNE model on three real-world datasets: Twitter, Last.fm and Flickr. We collected Twitter dataset from January to March 2017 with Twitter API[24], which includes users and their posts, and the Last.fm dataset for 1 month through Last.fm API[25], which contains users and artists. We also adopted Flickr dataset[26] released online with friend relationships, images, and the activities of user comment image. In order to enrich the information about user and image, we extracted the timestamp of user comments, image uploading timestamp and image description with Flickr API[27]. For all datasets, the user-user links are constructed from bi-directional friendships between social network users, user-item links are constructed from the different activities of users (e.g., posting, listening or commenting items), and item-item links are constructed if the two artists/images share the same tag or the two posts have the same hashtag. We observe that there are some stray nodes with few links in networks which will damage the community detection procedure, so we treat them as noisy nodes. Thus, we preprocess these datasets by deleting stray nodes. After that, we get three high-quality networks whose properties are shown in Table 6.2.

Table 6.2: Some statistics of the datasets.

Datasets	#Users	#Items	#User-user links	#User-item links	#Item-item links
Twitter	69,830	6,284,665	429,836	69,131,820	30,795,807
Last.fm	41,258	10,361	235,417	11,486,510	1,820,649
Flickr	2,037,538	1,262,978	219,098,660	14,913,164	97,549,330

Baselines. We compared our model with five state-of-the-art methods:

- **Weighted Regularized Matrix Factorization (WRMF).** A state-of-the-art offline matrix factorization model introduced by [261] is computed in batch mode, assuming the whole stream is stored and available for training.
- **Stream Ranking Matrix Factorization (RMFX).** It achieves partly online and much

[24] https://dev.twitter.com/docs

[25] http://www.last.fm/api/

[26] http://arnetminer.org/lab-datasets/flickr/flickr.rar

[27] https://www.flickr.com/services/api/

quicker updates of matrix factorization introduced in [209].

- **Metapath2vec (M2V)** [45]. It uses metapath-based random walks on heterogeneous graphs to obtain node representations. Following [45], we employ 5 meaningful meta-paths whose lengths are not longer than 4, “UIU”, “UUIU”, “UIIU”, “UIUIU” and “UUIIU”, since long meta-paths are likely to introduce noisy semantics. Here, ‘U’ = $User$ and ‘I’ = $Item$.
- **PTE** [46]. We build three bipartite heterogeneous networks: user-user, user-item and item-item, and retrain it as an unsupervised embedding methods.
- **M-NMF** [49]. It jointly models node and community embedding using non-negative matrix factorization.

Moreover, we also select three community detection baselines:

- **MetaFac** [267]. It performs tensor factorization on multi-relational network for overlapping community discovery.
- **Link Clustering (LC)** [268]. It aims to find link communities rather than nodes.
- **BigCLAM** [269]. proposed a typical non-negative matrix factorization based model for large-scale network.

Parameter Settings. WRMF setup is as follows: $\lambda_{\text{WRMF}} = 0.015, C = 1, epochs = 15$ for all datasets, which corresponds to a regularization parameter, a confidence weight that is put on positive observations, and the number of passes over observed data, respectively [261]. For RMFX, we set regularization constants λ_{RMFX}, learning rate η_0, and a learning rate schedule α equal to $0.1, 0.1, 1$ for Twitter, $0.15, 0.05, 1.5$ for Last.fm and $0.1, 0.15$, 1 for Flickr using grid-search on stream data with cross validation [209]. Moreover, the number of iterations is set to the size of the reservoir. For all the embedding algorithms (metapath2vec, PTE, M-NMF and our model), the embedding dimensionality is set to 128, context window length is set to 8, walk length is set to 40, walks per vertex is set to 30, the neighborhood size is equal to 7 and the size of negative samples is equal to 5 for all datasets. For M-NMF, we followed the same tuning procedure in [49], and we found out that $\alpha = 0.1$ and $\beta = 5$ works at best for Twitter and Last.fm, while $\alpha = 10$ and $\beta = 5$ for Flickr. As for our m-DNE model for three datasets, we also set the dimension of community representation as 128. Following [94], the smoothing factors α_l and β_l are set to 2 and 0.5 respectively. We set decay rate $\lambda = 0.2$ for Twitter and 0.1 for Last.fm and Flickr. The number of communities $\mathcal{K}$ is set to 20 for m-DNE and M-NMF model [49]. We run experiments on Linux machines with eight 3.50 GHz Intel Xeon(R) CPUs and 16 GB memory.

Evaluation Criteria. Given a dataset $\mathcal{D}$ ordered according to time, including user and item profiles, we use the first 50% of $\mathcal{D}$ as historical data pool to train the models, while the rest half data mimics the streaming input called "candidate set". For evaluation, we first randomly select a reference time as "current time" in candidate set. Then, we test our recommendations for the following week starting from reference time, while the data before reference time in candidate set are used to tune the hyper-parameters. However, WRMF and RMFX cannot explicitly handle new user/item introduction during the testing phase. For a fair comparison, all testing sets only cover users/items existing in training set. During evaluation phase, all experimental results are averaged over 10 different runs for reliability, and there is no temporal overlapping between any testing set.

Since we are interested in measuring top-k recommendation instead of rating prediction, we measure the quality by looking at the $Recall@K$ [263] and Average Reciprocal Hit-Rank (ARHR) [247], which are widely used for evaluating top-k recommender systems. We show the performance when $k = \{1,5,10\}$, as a larger value of k is usually ignored for a typical top-k recommendation [263].

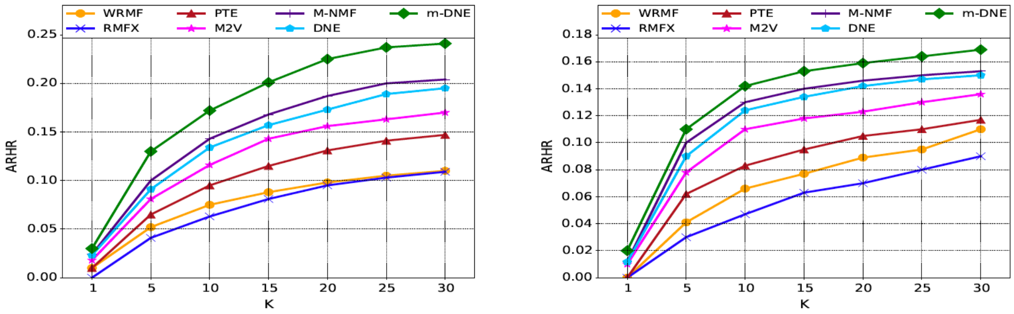
6.7.2 Results

Recommendation Effectiveness. Table 6.3 summarizes the item and friend recommendation performance between our model and baselines. Besides, we also test our model without community attribute integration represented as DNE. From the results, we can observe that the $Recall@K$ value grows gradually along with the increasing number of K, and the performance of item recommendation is better than friend recommendation. Besides, we can also observe on all datasets that: (1) Embedding-based algorithms (PTE, M2V, M-NMF, DNE and m-DNE) consistently perform better than non-embedding based benchmarks (WRMF, RMFX). It is because embedding-based algorithms can fully explore the network structure of the given information, which alleviates the issues of sparse and noisy signals. (2) The significant improvements show the promising benefit of the community integration and our incremental learning approach, which lead to the better performance of m-DNE than the other listed embedding methods.

Figure 6.2 compares the performance of alternative approaches taking ARHR as metric. During experiments, we vary the number of recommendations K from 1 to 30. As expected, our m-DNE model performs better with ARHR as well, and M-NMF ranks the second place followed by DNE, which shows the same orders in Table 6.3. In Figure 6.2(a), as we recommend more items, since we have more chance to answer the true interested items correctly, ARHR grows gradually with increasing number K. The same trends appear in the friend recommendation task.

Table 6.3: Top-k items and friends recommendation w.r.t. $Recall@K$ ($K = 1, 5, 10$).

Method	Twitter			Last.fm			Flickr		
	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10	Recall@1	Recall@5	Recall@10
Top-k items recommendation									
WRMF	0.152	0.229	0.301	0.226	0.293	0.387	0.204	0.261	0.356
RMFX	0.115	0.194	0.273	0.197	0.276	0.358	0.171	0.252	0.334
PTE	0.219	0.292	0.379	0.276	0.352	0.433	0.246	0.327	0.394
M2V	0.236	0.307	0.392	0.291	0.374	0.467	0.263	0.341	0.435
M-NMF	0.264	0.328	0.426	0.342	0.407	0.506	0.311	0.386	0.479
DNE	0.251	0.324	0.417	0.331	0.403	0.498	0.306	0.374	0.470
m-DNE	**0.309**	**0.385**	**0.472**	**0.395**	**0.471**	**0.557**	**0.368**	**0.449**	**0.531**
Top-k friends recommendation									
WRMF	0.113	0.175	0.266	0.172	0.247	0.314	0.148	0.220	0.281
RMFX	0.097	0.146	0.204	0.136	0.225	0.290	0.118	0.196	0.267
PTE	0.152	0.226	0.313	0.224	0.276	0.347	0.198	0.255	0.329
M2V	0.176	0.235	0.327	0.234	0.292	0.348	0.203	0.267	0.334
M-NMF	0.226	0.267	0.339	0.262	0.321	0.378	0.237	0.304	0.351
DNE	0.213	0.256	0.332	0.254	0.317	0.363	0.228	0.296	0.344
m-DNE	**0.243**	**0.294**	**0.371**	**0.298**	**0.352**	**0.406**	**0.275**	**0.329**	**0.390**



(a) Top-k items recommendation

(b) Top-k friends recommendation

Figure 6.2: Recommendation performance w.r.t. ARHR.

To evaluate the efficiency of our model, we compare our m-DNE with other baselines on Twitter. As all baselines are not designed to handle dynamics except RMFX, we compare their cumulative running time over all time steps and plot it in a log scale. Each time step represents one day period. As can be seen in Figure 6.3, m-DNE is much faster than the baselines which need to retrain and still show advantages compared with RMFX.

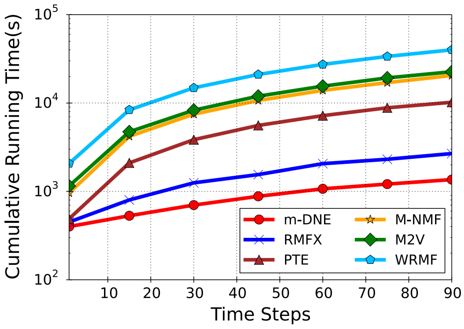
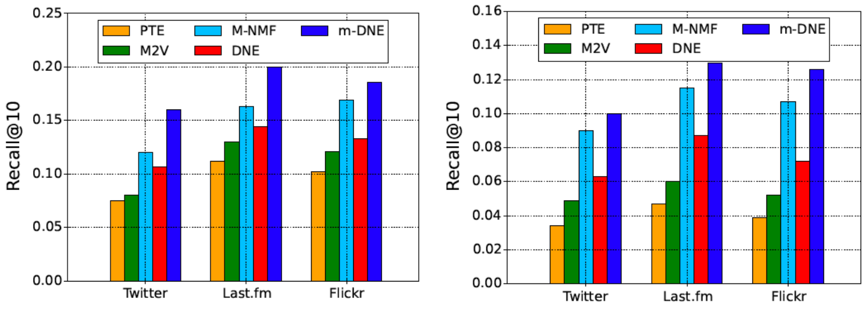


Figure 6.3: Cumulative running time comparison.

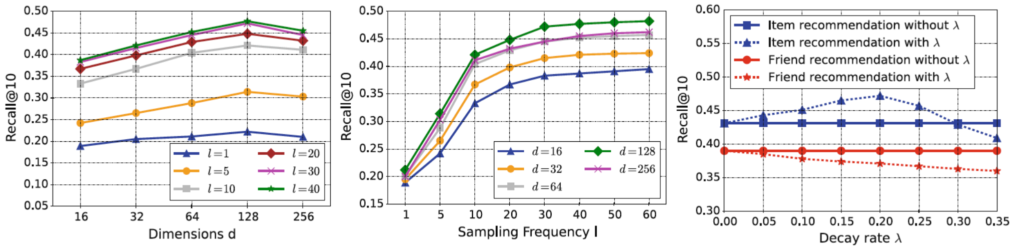
Test for Cold Start Problem. We also conduct experiments to study the effectiveness of different algorithms in addressing cold-start issues. As pre-processing, the target users who have less than 20 available items and social links in total are selected. As there are not many interaction records between users and items available for cold-start cases, WRMF and RMFX which are based on collaborative filtering, are not suitable for cold-start experiments. Thus, we compare m-DNE with the baselines which can leverage social information to recommend cold-start cases. The experimental results are shown in Figure 6.4, from which we have the following observations: (1) m-DNE model still performs best consistently in recommending cold-start cases; (2) by comparing with Table 6.3, the *Recall* value of all algorithms decreases. For instance, the Recall value of M-NMF rapidly drops from 42.6% to 12% for twitter item recommendation but still better than DNE model, while m-DNE deteriorate slightly, which validates that community-aware high order proximity and the ability to capture the dynamic properties of the network are key factors affecting the recommendation performance.



(a) Item recommendation (b) Friend recommendation

Figure 6.4: Recommendations for cold-start cases.

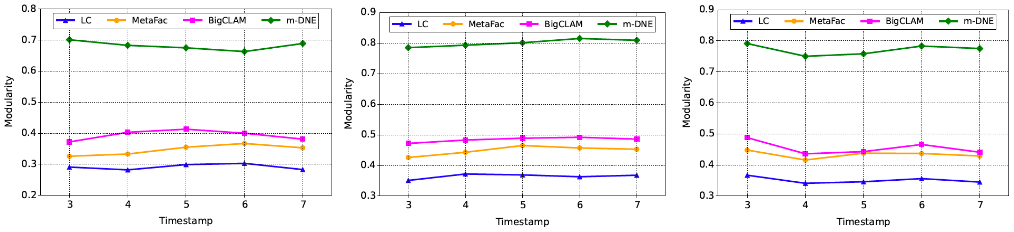
Sensitivity to Parameters. In this experiment, we study the influence of the embedding dimension d, the number of samples l and time decay rate λ by fixing the window size $|W| = 8$ and the random walk length $h = 40$. We vary one parameter each time to test the impact on recommendation performance with other parameters fixed. Because of the page limit, we only show the results on Twitter in Figure 6.5. But similar observations can be made on other datasets. Recommendation $Recall$ value of m-DNE model is not highly sensitive to the dimension d, but still presents a tendency that its recommendation accuracy increases with the increasing number of d holistically, and it reaches peak when d is around 128. However, m-DNE is sensitive to l with the $Recall$ score varying a lot. First, the performance of m-DNE increases quickly with the increasing number of l, this is because the model has not achieved convergence. Then, it does not change significantly when the number of samples becomes large enough, since m-DNE has converged. Thus, to achieve a satisfying trade off between effectiveness and efficiency of model training, we set $l = 30$ and $d = 128$ on all datasets. In Figure 6.5(c), λ shows different influence on item/user recommendation tasks. For item recommendation, the performance reaches the peak when $\lambda = 0.2$ but drops significantly afterwards. However, for user recommendation, the performance constantly decreases with the increasing value of λ. These phenomena show that in our case, items are more sensitive to time compared with users, and a suitable value of λ can help to improve the recommendation performance.



(a) dimensions, d (b) number of samples, l (c) time decay rate, λ

Figure 6.5: Effect of different parameters on performance.

Community Detection. We adopt modified modularity [270] to evaluate the quality of community detection methods without groud-truth label. As the results shown in Figure 6.6, we can observe that our m-DNE outperforms other state-of-the-art community detection methods on both datasets, which states the effectiveness of our model on community detections as well.



(a) Twitter dataset (b) Last.fm dataset (c) Flickr dataset

Figure 6.6: Community detection in terms of modularity.

Table 6.4 lists the top 4 tweets and top 3 users that are most likely appear in the community at $t = 2$ and 3. In both time intervals, all tweets and users are correlated with “technology” topics. At $t = 3$, the new popped tweets reflect users’ attention to the significant events of “Trump’s refugee ban” which began on January 28, 2017. Besides, we also discover that “technology” related community is tightly connected with “political” related community. For instance, at $t = 3$, we detect that the second tweet also appears in “political” community with different probability, and therefore it shows our community detection method can accurately identify the boundary vertices and balance the weights of the communities they belong to.

Table 6.4: Illustration of the community evaluation on Twitter dataset.

Node Type	T=2	T=3
Items	“The science community is rallying together for a march on Washington” https://twitter.com/i/moments/824382457570996224fi	#Techcrunch, Google planning AI tools for Pi makers this year: Google is intending to expand the dev toolsfi http://dlvr.it/NBz04x
	A scientistsfi march on DC is already in the works! Looks like early stages but watch http://www.scientistsmarchonwashington.com/ & follow @ScienceMarchDC.	Tech firms recall employees to U.S., denounce Trump’s ban on refugees from Muslim countries http://wpo.st/lAWX2 #Google #LinkedIn.
	We all age. But what does aging truly mean from a scientific perspective? http://ow.ly/XAwE307u3PR.	Google denounces Trump’s ban on refugees, recalls employees from Muslim countries http://wpo.st/emgW2 #MuslimBan.
	Google planning AI tools for Pi makers this year http://tcrn.ch/2krRVXG by @riptari.	Science brains unite! Stop the war on intelligence. Facts are facts. Truth matters. #ScienceMarch @ScienceMarchDC
Users	@jack; @TechCrunch; @alexisohanian ...	@cridge17; @jack; @demishassabis ...

6.8 Conclusion

In this paper, we propose m-DNE, an efficient model which learns the embedding of heterogeneous social network by jointly modelling the temporal semantic effects, social relationships and user behavior sequential patterns in a unified way. Community-aware high-order proximity is applied to optimize the node representations. Besides, a parallel incremental learning algorithm and an efficient query processing technique are employed for recommendation efficiency. The experimental results show the effectiveness of our m-DNE on social recommendations.

Part IV

Utilizing Visual Context for Social Recommendation

Chapter 7

Explainable Social Recommendation with Textual and Visual Fusion

Explainable recommendation, which provides explanations about why an item is recommended, has attracted growing attention in both research and industry communities. However, most existing explainable recommendation methods cannot provide multi-model explanations consisting of both textual and visual modalities or adaptive explanations tailored for the user's dynamic preference, potentially leading to the degradation of customers' satisfaction, confidence and trust for the recommender system. On the technical side, Recurrent Neural Network (RNN) has become the most prevalent technique to model dynamic user preferences. Benefit from the natural characteristics of RNN, the hidden state is a combination of long-term dependency and short-term interest to some degrees. But it works like a black-box and the monotonic temporal dependency of RNN is not sufficient to capture the user's short-term interest.

In this hapter, to deal with the above issues, we propose a novel Attentive Recurrent Neural Network (Ante-RNN) with textual and visual fusion for the dynamic explainable recommendation. Specifically, our model jointly learns image representations with textual alignment and text representations with topical attention mechanism in a parallel way. Then a novel dynamic contextual attention mechanism is incorporated into Ante-RNN for modelling the complicated correlations among recent items and strengthening the user's short-term interests. By combining the full latent visual-semantic alignments and a hybrid attention mechanism including topical and contextual attentions, Ante-RNN makes the recommendation process more transparent and explainable. Extensive experimental results on two real world datasets demonstrate the superior performance and explainability of our model.

7.1 Introduction

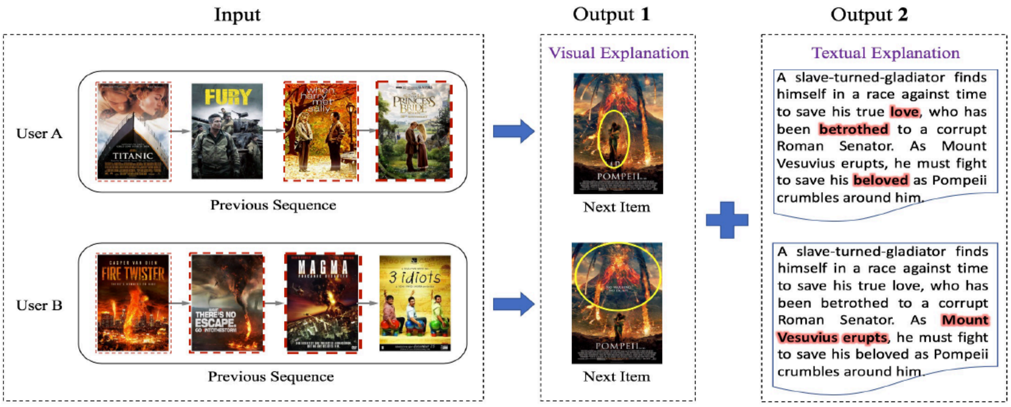
In recent years, explainable recommendation has become an active research topic in many online customer-oriented applications, such as social media, e-commerce and content-sharing websites. By explaining how the system works and/or why an item is recommended, the system becomes more transparent and has the potential to allow users to tell when the system is wrong (scrutability), help users make better (effectiveness) and faster (efficiency) decisions, convince users to try or buy (persuasiveness), or increase the ease of the user enjoyment (satisfaction) [271]. Current explainable models usually interpret the recommendations based on user reviews. For instance, Zhang et al. [272] proposed an Explicit Factor Model (EFM) to learn user cared features from the review information and fill them into pre-defined templates regarded as explanations. Chen et al. [135] and Wang et al. [273] extended EFM for more accurate user-item-feature explanations by leveraging tensor factorization techniques. Chen et al. [274] used attention mechanism to extract valuable item reviews for explaining the rating prediction.

Despite effectiveness, these explainable recommendation methods still suffer from some inherent issues: (1) Most of them model the item's characteristics by only leveraging their textual features, which leads to the limited recommendation performance and explanatory capability. In fact, for some types of items (e.g., clothing), their visual appearances play an important role in their properties, which can greatly bias the user's preference towards them. For example, users can easily determine whether they watch a movie based on the movie poster images. Thus, the visual features of items are also important complementary information for the explainable recommendation. (2) Most methods assume that user preferences are invariant and generate static explanations. However, in real scenarios, a user's preference is always dynamic, and s/he may be interested in different topics at different states. The static assumption can easily lead to incorrect matches between the explanation and user dynamic preference, thus impairing the recommendation performance and degrading customers' satisfaction, confidence and trust for the recommender system.

Previous works that leverage the visual information for personalized recommendation usually transform images into embedding vectors, which are then incorporated with collaborative filtering (CF) for improving the performance. For example, McAuley et al. [200] adopted neural networks to transform images into feature vectors, and used the vectors for product style analysis and recommendation; He et al. [192] further extended the approach to pair-wise learning to rank for recommendation; Zhang et al. [202] adopted image features for recommendation in a social network setting; Wang et al. [203] extracted image features with neural network for point-of-interest recommendation. Though the recommendation performance has been improved by incorporating image representation extracted with (convolutional) neural networks, the related works have largely ignored an important advantage of leveraging images for recommendation – its ability to provide intuitive visual explanations. This is because by transforming the whole image into a fixed latent vector, the images become hardly understandable for users, which makes it difficult for the model to generate visual explanations to accompany certain recommendations.

On the other hand, recent approaches that leverage Recurrent Neural Network (RNN) for recommendation have demonstrated their effectiveness in modelling the temporal dynamics of user preferences. RNN based methods adopt the last hidden state as the user's final representation to make recommendations. With the help of gated activation function like long-short term memory or gated recurrent unit [275], RNN can better capture the long-term dependency. However, it works like a black-box, for which the reasons underlying a prediction cannot be explicitly presented. Besides, due to the recurrent structure and fixed transition matrices, RNN holds an assumption that temporal dependency has a monotonic change with the input time steps [276]. It assumes that the current item or hidden state is more significant than the previous one. This monotonic assumption would restrict the modelling of user's short-term interests and can not well distinguish the importance of several recent factors. For example, a user is looking for interesting movies on the Internet. During browsing, s/he tends to click on some movies with the "disaster" topic which is treated as the user's short-term interest, meanwhile s/he might click a comedy movie by accident or due to curiosity. In this case, small weight should be provided for the comedy movie. So the short-term interest should be carefully examined and needs to be integrated with the long-term dependency.

In this chapter, we focus on the problem of simultaneously multi-model explanation generation and dynamic user preference modelling in the context of explainable recommendation. The problem setup is illustrated in Figure 7.1. We propose a novel Attentive Recurrent Neural Network (Ante-RNN) to address this problem. More specifically, we first learn image representations with the latent semantic alignments between image regions and the corresponding words in text. Meanwhile, in order to capture the user's long-term preference, a topical attention mechanism which can model the interactions between the words and the user's interested topics is adopted to learn text representations. After that, the representations from image and text sources are integrated to obtain a joint representation of visual and textual features for each item by investigating different modality fusion strategies, which is used as the input of Ante-RNN. Then a novel dynamic contextual attention mechanism is incorporated into our model for modelling the complicated correlations among recent items and strengthening the user's short-term interests. By combining the full latent visual-semantic alignments and the attention weights learned from topical attention network and contextual attention network, Ante-RNN makes the recommendation process more transparent and explainable. Compared with existing methods, our model not only improves the recommendation performance, but also generates textual and visual explanations for the recommended items.



A slave-turned-gladiator finds himself in a race against time to save his true love, who has been betrothed to a corrupt Roman Senator. As Mount Vesuvius erupts, he must fight to save his beloved as Pompeii crumbles around him.

A slave-turned-gladiator finds himself in a race against time to save his true love, who has been betrothed to a corrupt Roman Senator. As Mount Vesuvius erupts, he must fight to save his beloved as Pompeii crumbles around him.

Figure 7.1: Problem Setup. Given the users' clicking sequence of items, different parts of the images are marked in rectangle to provide intuitive explanations for the next recommended item. Meanwhile, their textual explanations are also provided by highlighting the topic-related words. Besides, the item in the red dashed is more relevant to the current user's intention. And the red line is thicker when the item is more important.

To summarize, this chapter makes the following contributions:

– We propose an Attentive Recurrent Neural Network (Ante-RNN) for the dynamic explainable recommendation which could provide multi-model explanations according to the user dynamic preference. To the best of our knowledge, it is the first time to jointly explore multi-modal and adaptive explanations in a unified framework for the personalized recommendation.

– In order to alleviate the issues caused by the monotonic assumption of RNN, a hybrid attention mechanism is developed to capture the user's long-term dynamic interest over different topics and strengthen the short-term interest simultaneously. More importantly, our proposed dynamic contextual attention scheme incorporates diverse temporal factors of the user's clicking sequence of items (e.g. time interval and the time of week) to further improve the recommendation performance.

– We analyze and study a variety of fusion strategies for mutual association learning across modalities, and find that the attention-based fusion robustly achieves the best results.

– We conduct extensive experiments on two real large-scale datasets. The results show that Ante-RNN outperforms state-of-the-art baselines in terms of Recall and NDCG on both datasets.

The remainder of the chapter is organized as follows. Section 7.2 introduces the related work. In Section 7.3, we formally define the problem and our new model. We describe the datasets, comparative approaches, the evaluation criteria we use and experimental results in Section 7.4. Finally, we present the conclusions and future work in Section 7.5.

7.2 Related Work

7.2.1 Explainable Recommendation

Researchers have shown that providing appropriate explanations could improve user acceptance of the recommended items [277], as well as benefit user experience in various other aspects, including system transparency, user trust, effectiveness, efficiency, satisfaction and scrutability [271]. However, the underlying algorithm may influence the types of explanations that can be constructed. In general, the computational complex algorithms within various latent factor models make the explanations difficult to be generated automatically [271]. Many meticulously designed strategies have been investigated to tackle the problem. For instance, the authors in [132] aligned user/item latent factors in matrix factorization (MF) with topical distribution in latent dirichlet allocation (LDA) for joint parameter optimization under the supervision of both score ratings and textual reviews, and thus the user preferences are explained by the learned topical distributions. Ling et al. [278] applied topic modelling techniques with mixture of Gaussians on the reviews and generated interpretable topics. Bao et al. [279] further extended Ling's work and proposed a novel topical matrix factorization model (TopicMF) to extract topics from each review. To explain finer-grained user preference, some approaches [135, 272] combined matrix factorization (MF) and sentiment analysis (SA) to generate explanations at the feature-level. More specifically, they extracted feature-opinion-sentiment triplets from the user review information, and infused them into MF for collective user preference modelling. The explanations were provided by filling the predicted user cared features into pre-defined templates. Despite effectiveness, the final results of these methods can be easily affected by the accuracy of the review preprocessing tools, and the complex process for extracting triplets usually render them inefficient.

Recently, with the rapid development of deep learning technology, there has been a surge of interest in leveraging attention mechanisms to explain a recommendation. A common approach employed by these works involves two steps. First, they merge the raw user review information related to a user (or an item) into a document and attentively discovered valuable information in the document. The next step is to provide the explanations by highlighting the words with the highest attention weights. In particularly, Seo et al. [243] and Chen et al. [274] automatically learned the importances of different review sentences under the supervision of user-item rating information. To provide explanations tailored for different target items, Tay et al. [280] adopted "co-attention" mechanism to capture the correlations between users and items. Apart from user-review explanations, Ai et al. [281] conducted explainable recommendation by reasoning over knowledge graph embeddings, where explanation paths between users and items were constructed to generate knowledge-enhanced explanations. Hu et al. [282] built a multilevel personal filter to calculate users' attractiveness on textual information of items and provided interpretable recommendations upon them.

Although these methods have achieved promising results, they failed to model user dynamic preference, and the provided explanations were usually static and unimodal, which may weaken the persuasiveness of the explanations as mentioned before.

7.2.2 Sequence-aware Recommendation

Recently, a number of research works have demonstrated that the sequential information (e.g., user sequential behaviors), which are regarded as the important information source for understanding user dynamic preferences, can be utilized to improve personalized recommendations at the right time. In specific, early methods care more about transition properties between two successive behaviors. For instance, the factorized personalized Markov chains (FPMC) [237] combined matrix factorization with one-order Markov chain to capture the influence of the last behavior towards the next one. The hierarchical representation model (HRM) [283] generalized FPMC into a representation learning framework, and significantly improved the recommendation performance. The major limitation of these methods lies in the ignoring of long-term preference dependency.

To solve this problem, many models were proposed to capture user multi-step behaviors based on the recurrent neural network (RNN). Yu et al. [53] represented a basket acquired by pooling operation as the input layer of RNN, which outperforms the state-of-the-art methods for next basket recommendation. Song et al. [54] proposed a multi-rate Long Short-Term Memory (LSTM) with considering temporal user preferences for commercial news recommendation. Hidasi et al. [55] utilized RNNs for session-based online recommendation. Furthermore, with the ability to express, store and manipulate the records explicitly, dynamically and effectively, external memory networks (EMN) [57] have shown their promising performance for many sequential prediction tasks, such as question answering (QA) [284], natural language transduction [285], and recommender system [58]. Chen et al. [58] proposed a novel framework integrating recommender system with external User Memory Networks which could store and update users' historical records explicitly. Huang et al. [59] proposed to extend the RNN-based sequential recommender by incorporating the knowledge-enhanced Key-Value Memory Network (KV-MN) for enhancing the representation of user preference. Despite these models achieve some degree of improvements, one of the important features - the temporal context of user sequential behaviors - has been totally ignored. Recently, Zhu et al. [286] designed a model called Time-LSTM to demonstrate the importance of time interval information for user dynamic preference modelling. However, their proposed model was designed for a particular type of contextual information (i.e. time intervals) and is not flexible to incorporate other types of context (e.g. the time of week). What's more, the Time-LSTM model cannot automatically select important interaction records in the user-item interaction history when recommending items.

To model the different impacts of a user's diverse historical interests on current candidate item, Wang et al. [287] designed an attention module to dynamically calculate a user's aggregated historical representation. Pei et al. [288] extended recurrent networks for modelling user and item dynamics with a novel gating mechanism, which adopts the attention model to measure the relevance of individual time steps of user and item history for recommendation. Li et al. [289] explored a hybrid encoder with an attention model to capture both the user's sequential behavior and main purpose in the current session.

Specifically, they involved an item-level attention mechanism which allowed the decoder to dynamically select and linearly combine different parts of the input sequence. Different from existing works, we propose a hybrid attention mechanism which takes into account the user's long-term interested topics and short-term contextual surroundings at the same time. And more importantly, the proposed dynamic contextual attention scheme enables our model to selectively concentrate on critical parts of the sequential information and is fairly flexible, which can easily add other types of contextual information when available. To illustrate more clearly, Table 7.1 summarizes the differences among related works with sequential information.

Table 7.1: Summary of related studies about the sequence-aware recommendation. Fields without information in the related study are marked with a hyphen.

		Features			
Reference	**Model**	**Short-term Behaviors**	**Long-term Behaviors**	**Relevance of Historical Behaviors**	**Temporal Context**
Rendle et al. 2010	Markov chain	✓	-	-	-
Wang et al. 2015	Markov chain	✓	-	-	-
Hidasi et al. 2015	RNN	✓	✓	-	-
Yu et al. 2016	RNN	✓	✓	-	-
Song et al. 2016	RNN	✓	✓	-	-
Zhu et al. 2017	RNN	✓	✓	-	✓
Chen et al. 2018b	RNN + Memory network + Attention mechanism	✓	✓	✓	-
Huang et al. 2018	RNN + Memory network + Attention mechanism	✓	✓	✓	-
Wang et al. 2018b	DNN + Attention mechanism	✓	✓	✓	-
Pei et al. 2017	RNN + Attention mechanism	✓	✓	✓	-
Li et al. 2017	RNN + Attention mechanism	✓	✓	✓	-
Ante-RNN	RNN + Attention mechanism	✓	✓	✓	✓

7.2.3 Multi-modality Fusion

Multi-modality fusion enables us to leverage complementary information presented in multimodal data, thus discovering the dependency of information on multiple modalities. There exist two commonly used fusion strategies in previous research: feature-level fusion and decision-level fusion. Specifically, feature-level fusion aims to directly combine feature vectors by concatenation [290] or kernel methods [291, 292]. Poria et al. [292] used a multiple kernel learning strategy to fuse the modality data on the feature-level. Zadeh et al. [293] proposed a tensor fusion technique to fuse audio, visual and textual features at feature level. Decision-level fusion builds separate models for each modality and then integrates the outputs together using a method such as majority voting or weighted averaging [294, 295]. For instance, Wöllmer et al. [296] combined the results of the text and audio-visual modalities by a threshold score vector on the decision-level. Deep neural network fusion was proposed in a recent study to fuse the extracted modality-specific features [297, 298]. More recent approaches introduced LSTM structures to fuse the features at each time step [296, 300].

In recommender systems, previous works often adopt the strategy of combining image-, rating- and review-based features for boosting recommendation performance. The most frequently used fusion methods are concatenation [301, 290], addition [302, 303] and element-wise product [304]. Recently, Zhang et al. [204] integrated images with reviews and ratings in a multimodal deep learning framework for top-n recommendation. Cui et al. [9] proposed a multi-modal Marginalized Denoising AutoEncoder (3mDAE) to learn fusion features by reconstructing the original multi-modal data. However, only few works consider the sophisticated interactions between different modalities in the recommendation. For instance, Cheng et al. [305] adopted a fully-connected neural layer directly after the addition fusion step to get better fusion features in the rating prediction. Lian et al. [306] proposed a multi-channel deep fusion model which leverages an attention mechanism to merge latent representations learnt from different domains in the personalized news recommendation. In this work, we explore several fusion techniques for mutual association learning across modalities (mainly based on the textual and visual modalities) in the context of explainable recommendation.

7.3 Proposed Ante-RNN Model

In this section, we describe the proposed Attentive Recurrent Neural Network (Ante-RNN) for the dynamic explainable recommendation in detail. The basic idea of Ante-RNN is to build a unified representation of the user's interacted items, and then generate predictions along with explanations based on it. The representation should take into account various potential factors that influence user's next decision. As shown in Figure 7.2, our model firstly learns text embedding with topical attention network fused with image embedding with the according textual alignment in the same D-dimensional space to represent item. Then our dynamic contextual attention mechanism learns attentive weights by considering the contextual influence of current interacting (e.g. clicking/reading) item to strengthen the representation before GRU network, and thus to improve the recommendation performance. Furthermore, the attention weights learned from topical attention network and contextual attention network, can in turn help to explain the recommendation results by the descriptive snippets learned from images and texts.

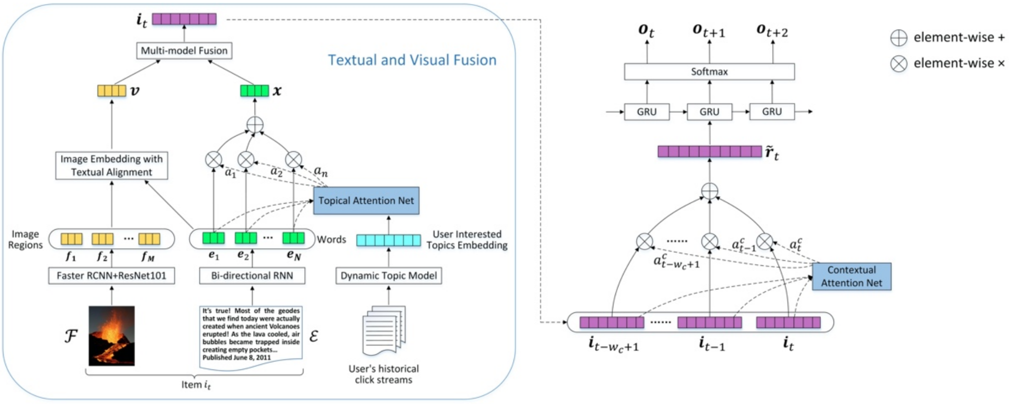

i_t
Multi-model Fusion
Textual and Visual Fusion
v
x
Image Embedding with Textual Alignment
a_1
a_2
a_n
Topical Attention Net
Image Regions
f_1 f_2 f_M
e_1 e_2 e_N
Words
User Interested Topics Embedding
Faster RCNN+ResNet101
Bi-directional RNN
Dynamic Topic Model
$\mathcal{F}$
It's true! Most of the geodes that we find today were actually created when ancient Volcanoes erupted! As the lava cooled, air bubbles became trapped inside creating empty pockets... Published June 8, 2011
$\mathcal{E}$
Item i_t
User's historical click streams
o_t o_{t+1} o_{t+2}
Softmax
GRU
GRU
GRU
$\tilde{r}_t$
$a^c_{t-w_c+1}$
a^c_{t-1}
a^c_t
Contextual Attention Net
i_{t-w_c+1} i_{t-1} i_t
element-wise +
element-wise ×

Figure 7.2: The proposed Ante-RNN framework for explainable recommendation. (1) In blue square, the model learns image representation $\boldsymbol{v}$ with textual alignment and text representation $\boldsymbol{x}$ with topical attention net of item at timestamp t, then the fused representation $\boldsymbol{i}_t$ can be achieved through textual and visual fusion component to denote the item embedding at timestamp t. (2) The contextual attention net takes the fused representation $\boldsymbol{i}_t$ as input to learn user's dynamic interest representation $\tilde{\boldsymbol{r}}_t$. Finally, the model outputs the probability score of the next possible item the user interacted with.

In the rest of this section, we first define relevant notations used in this chapter and formulate the recommendation problem. Then, we present the image embedding with textual alignment and topic-based text embedding in Section 7.3.2 and Section 7.3.3 respectively. In Section 7.3.4, several multi-model fusion strategies are explained in detail. We introduce the dynamic contextual attention mechanism in Section 7.3.5 and finally in Section 7.3.6, the whole objective of our model and its training procedure will be described.

7.3.1 Problem Formulation

Throughout this chapter, all vectors are column vectors and are denoted by bold lower case letters (e.g. $\boldsymbol{x}$ and $\boldsymbol{y}$), while matrices are represented by bold upper case letters (e.g., $\boldsymbol{X}$ and $\boldsymbol{Y}$). We use calligraphic letters to represent sets (e.g., $\mathcal{U}$ and $\mathcal{I}$). Lower case letters (e.g. x and y) represent as scalar parameters. Table 7.2 summarizes the notations of frequently used variables.

Let $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}$ and $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$ represent the sets of users and items respectively. For each user, we chronologically organize his/her historical behaviors as a sequence of tuples $\mathcal{O}^u = \{(i_1^u, t_1^u), (i_2^u, t_2^u), \dots, (i_{l_u}^u, t_{l_u}^u)\}$ with the length l_u, where $t_1^u \leq t_2^u \leq \cdots \leq t_{l_u}^u$ and the s-th element (i_s^u, t_s^u) means that user u interacted with (i.e. clicked/viewed) item $i_s^u \in \mathcal{J}$ at time t_s^u. Additionally, an image and a text description are available for each item $i \in \mathcal{J}$. Our task of explainable recommendation with user dynamic preference is to learn a model such that for any given user's historical interacted item set $\mathcal{O}^u$, it generates a list of top-k personalized items as recommendations for user u. And further, its internal parameters or intermediate outputs should provide explanations on both textual and visual modalities for these recommended items according to the user's preference at time $t_{l_u+1}^u$.

Table 7.2: Notations used in this chapter.

Symbol	Description
$\mathcal{U}, \mathcal{I}$	The set of users and items
$\mathcal{E}, \mathcal{F}$	The set of word features and image features for an item
M, N	The number of image features and word features for an item
$\boldsymbol{v}, \boldsymbol{x}$	The image embedding and the text description embedding
D	Dimensionality of the image and text embedding
$\boldsymbol{C}$	The affinity matrix whose element represents the similarity between image region and text word
$\boldsymbol{\eta}_t^u$	User u's historical interested topics representation
ψ	The number of topics
$\boldsymbol{i}$	Item representation after multi-model fusion
w_c	The contextual window length
$\tilde{\boldsymbol{r}}_t$	User's interest representation at timestamp t
$\boldsymbol{T}_w$	One hot encoding vector of time of week
$\boldsymbol{\delta t}$	One hot encoding vector of time interval
$\boldsymbol{o}_t$	The output vector from GRU

7.3.2 Image Embedding with Textual Alignment

Inspired by the work of Lee et al. [231], we learn image and its corresponding text description in a joint manner. Though items can be expressed by multiple ways such as image, video, sound, text and so on, the combined representations of items should require a feature fusion mechanism to ensure that multiple inputs are appropriately integrated. Furthermore, the strategy that synchronizes different inputs of multi-modalities at the same level is an effective way as well [307]. Therefore, in this Chapter, we consider to represent image at the word level because word is an important basic unit of representing users' interests, and thus image-based item representation and text/word-based item representation can be projected at the same space.

Suppose an item includes a set of word features $\mathcal{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_N\}$ in which each element $\boldsymbol{e}_i \in \mathbb{R}^D, i = 1,2, \dots, N$ denotes a word representation in the text description, and a set of image features $\mathcal{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \dots, \boldsymbol{f}_M\}$ in which each element $\boldsymbol{f}_j \in \mathbb{R}^D, i = 1,2, \dots, M$ denotes a region representation in the image. Same with Lee et al. [231], the image region representations are derived by adopting the Faster R-CNN model in conjunction with ResNet-101 pre-trained by Anderson et al. [186] to recognize the salient and obvious objects from image. Then, a fully-connected layer is added to transform each region representation into D-dimensional space. The word representations are achieved using bi-directional GRU to find the relationship between words and map language to the same dimensional semantic vector space as image regions.

Given $\mathcal{E} \in \mathbb{R}^{D \times N}$ and $\mathcal{F} \in \mathbb{R}^{D \times M}$ the image embedding with textual alignment starts with defining an affinity matrix $\boldsymbol{C} \in \mathbb{R}^{N \times M}$ whose element c_{ij} represents the similarity between the corresponding feature vector pair of $\boldsymbol{e}_i \in \mathcal{E}$ and $\boldsymbol{f}_j \in \mathcal{F}$. Specifically, $\boldsymbol{C}$ is defined as

$$\boldsymbol{C} = \tanh\left(\mathcal{E}^T \boldsymbol{W}^b \mathcal{F}\right) \tag{1}$$

where $\boldsymbol{W}^b \in \mathbb{R}^{D \times D}$ denotes the correlation matrix to be learned.

Next, based on the affinity matrix, to weigh the alignment of each image region with respect to the text description, we adopt a weighted summation of all word representations denoted as

$$\boldsymbol{a}_j^e = \sum_{i=1}^{N} \alpha_{ij}^e \, \boldsymbol{e}_i \tag{2}$$

where $\alpha_{ij}^e = \exp(c_{ij}) \, / \sum_{i=1}^{N} \exp\,(c_{ij})$ score on how well the jth image region and the ith word match. After that, to determine the importance of image regions given the text description, the relevance between the jth region and the corresponding description can be defined as

$$R\left(\boldsymbol{f}_j, \boldsymbol{a}_j^e\right) = \frac{\boldsymbol{f}_j^T \boldsymbol{a}_j^e}{\|\boldsymbol{f}_j\| \cdot \left\|\boldsymbol{a}_j^e\right\|} \tag{3}$$

Then, the similarity between image $\mathcal{F}$ and text description $\mathcal{E}$ can be defined as

$$\mathrm{S}(\mathcal{E},\mathcal{F}) = \frac{1}{M}\sum_{i=1}^{M} R(\boldsymbol{f}_j, \boldsymbol{a}_j^e) \tag{4}$$

And the representation $\boldsymbol{v}$ of image $\mathcal{F}$ can be represented as the weighted summation of all regions with respect to the alignments of text.

$$\boldsymbol{v} = \sum_{j=1}^{M} R\left(\boldsymbol{f}_j, \boldsymbol{a}_j^e\right) \cdot \boldsymbol{f}_j \tag{5}$$

In Lee et al. [231], the authors only focus on the hardest negatives in a mini-batch when formulating the objective function. In practice, for computational efficiency, rather than summing over all the negative samples as Kiros et al. [308], it usually considers only the hard negatives in a mini-batch of stochastic gradient descent. Thus, we define our triplet ranking loss as

$$l(\mathcal{E},\mathcal{F}) = \max\,[0, \alpha_1 - \mathrm{S}(\mathcal{E},\mathcal{F}) + \mathrm{S}\left(\hat{\mathcal{E}},\mathcal{F}\right)] \tag{6}$$

where α_1 denotes the margin in triplet loss, $\hat{\mathcal{E}} = argmax_{t \neq \mathcal{E}}\mathrm{S}(t,\mathcal{F})$ represents the hardest negative. Different from Lee et al. [231], we only take into account the image-text alignment instead of both image-text and text-image alignments for that we care about how the text description can help image representations solely.

7.3.3 Topic-based Text Embedding

To introduce user's historical interested topics into the model learning procedure and help to learn a better representation of text description, we propose a topical attention network which incorporates topic distribution to weigh the importance and relevance of each word in the text. Specifically, we first conduct topic modelling approach on all the users' historical behaviour streams to build a shared user topic space and learn the topical distribution for each user. Users' historical behaviours are collected at a certain time interval, for instance daily, hourly and weekly. In this chapter, we leverage stream LDA model introduced by Gao et al. [266] to learn topic distributions and update the model with every user's coming streams incrementally. Therefore, the learned topic space is timely updated and can well track the recent focuses on user behaviours. After that, we aggregate all historical topic distributions of each user to derive the representation of user interested topics at the current time. Furthermore, a time decay approach [309] is adopted to weight the different importance of the coming streams. Thus, the user's interested topics at time stamp t can be defined as:

$$\eta_t^u = \frac{1}{N_u} \sum_{i=1}^{t} \xi_i^u \cdot e^{-\lambda |t-i|} \tag{7}$$

where ξ_i^u is the user's topic distribution at time stamp i, $|t-i|$ indicates the time difference between the current time and the topic time stamp i. N_u is a normalization parameter and λ is the time decay parameter.

Then, we can derive the interested topics embedding of each user u as $\boldsymbol{\eta}_t^u \in \mathbb{R}^{\psi \times 1}$ at time stamp t, where ψ is the number of topics. After that, the topical attention network outputs the text embedding $\boldsymbol{x} \in \mathbb{R}^D$ for each item i computed as a weighted summation of each word embedding $\boldsymbol{e}_j$:

$$\boldsymbol{x} = \sum_{j=1}^{N} a_j \boldsymbol{e}_j \tag{8}$$

where D is the dimension of the word embedding, $a_j \in [0, 1]$ is the attention weight of $\boldsymbol{e}_j$ and $\sum_j a_j = 1$. To obtain $a_j, j \in [1, N]$, we use the following equation to compute scores on how well the interested topics embedding $\boldsymbol{\eta}_t^u$ matches the word embedding in position j

$$g_j = \boldsymbol{q}_a^T \tanh\left(\boldsymbol{W}^a \boldsymbol{\eta}_t^u + \boldsymbol{U}^a\, \boldsymbol{e}_j\right) \tag{9}$$

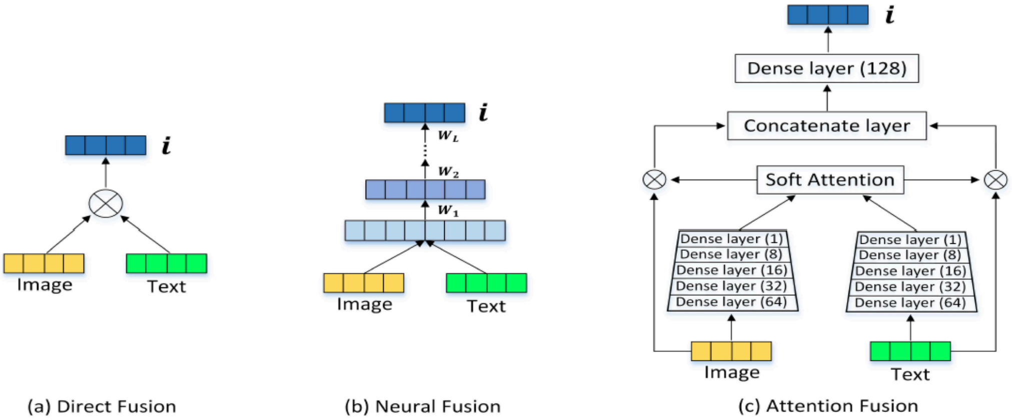
where $\boldsymbol{W}^a \in \mathbb{R}^{D \times \psi}$, $\boldsymbol{U}^a \in \mathbb{R}^{D \times D}$ and $\boldsymbol{q}_a \in \mathbb{R}^{D \times 1}$ are the weight matrices. Finally, the topical attentive weight score a_j can be calculated with a softmax function

$$a_j = softmax(g_j) = \frac{\exp(g_j)}{\sum_{j=1}^{n} \exp(g_j)} \tag{10}$$

7.3.4 Multi-Model Fusion

In previous sections, we have described the ways to extract image and text representations, but how to model the interactions between these two features and obtain a better fusion representation is still a problem worth exploring. Therefore, in this section, we consider three different multi-modal fusion methods as shown in Figure 7.3 to explore the sophisticated effects.

Direct fusion. An intuitive way to do the feature fusion is to combine the learned representations of multi-modalities directly. Normally, there are three ways to fuse the learned representations, namely *concatenation*, *addition* and *element-wise* product. Here, we apply element-wise product which has been verified its effectiveness by Chen et al. [310] and reveals favored performance in our experiments.



(a) Direct Fusion (b) Neural Fusion (c) Attention Fusion

Figure 7.3: Multi-model fustion architectures.

$$\boldsymbol{i} = \boldsymbol{v} \otimes \boldsymbol{x} \tag{11}$$

where $\boldsymbol{v}$ and $\boldsymbol{x}$ denote the learned textual and visual representations and $\boldsymbol{i} \in \mathbb{R}^D$ is the output after fusion. We omit subscript t for a simpler expression.

Neural fusion. Inspired by the work of Cheng et al. [305] but differently, we first concatenate the visual and textual representations directly to keep the original modality characteristics, and then leverage a neural network to fuse them in a complex non-linear way.

$$\boldsymbol{i} = DNN([\boldsymbol{v}; \boldsymbol{x}]) \tag{12}$$

where ; represents the concatenation operation. As for $DNN(\cdot)$ model, we leverage several fully connected layers stacked together to derive the non-linear output.

$$\boldsymbol{r}'_0 = [\boldsymbol{v}; \boldsymbol{x}]$$

$$\boldsymbol{r}'_1 = \varphi(\boldsymbol{W}_1 \boldsymbol{r}'_0 + \boldsymbol{b}_1)$$

$$\boldsymbol{r}'_2 = \varphi(\boldsymbol{W}_2 \boldsymbol{r}'_1 + \boldsymbol{b}_2)$$

$$\dots \dots,$$

$$\boldsymbol{i} = \varphi(\boldsymbol{W}_L \boldsymbol{r}'_{L-1} + \boldsymbol{b}_L)$$

where $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$ denote the weight matrix and bias for the lth fully connected layer. $\varphi(\cdot)$ denotes the activation function.

Attention fusion. Same modality may have different contributions for different recommendation tasks. For instance, people show more interests on visual-related features than textual descriptions on image recommendation tasks, such as Pinterest and Instagram. While textual features might provide more useful information than other kinds of modalities in news or movie recommendations. To fully exploit the difference of multimodal nature in recommendation tasks, we apply an attention mechanism to assign different weights for multi-modalities.

Different from previous two fusions, attention fusion adopts an attention network over the extracted representations of modality-specific features, helping the recommender system to tell the different importance of the different modalities. Following the work of Gu et al. [311], we adopt a tower pattern network structure as the base of our attention network. The bottom layer is the widest and each successive layer has smaller number of neurons. Ultimately, the output from the last layer has the dimension of k, representing the relative importance for k different modalities. In this chapter, we set $k = 2$ denoting the visual and textual modalities. Then, a softmax layer is applied to generate the weighted score for the modalities:

$$\boldsymbol{s} = softmax(TowerNet([\boldsymbol{v}; \boldsymbol{x}])) \tag{13}$$

where $TowerNet(\ \cdot\ )$ represents the deep neural network with tower structure. $\boldsymbol{s} = [s_v, s_t]$ is a $k = 2$ dimensional vector representing the visual and textual attention score. Finally, a dense layer is used to learn the associations across weighted multi-modalities:

$$\boldsymbol{i} = tanh(\boldsymbol{W}_e[(1 + s_v)\boldsymbol{v}; (1 + s_t)\boldsymbol{x}] + \boldsymbol{b}_e) \tag{14}$$

where $\boldsymbol{i} \in \mathbb{R}^D$ denotes the final fused item representation. $\boldsymbol{W}_e \in \mathbb{R}^{D \times 2D}$ and $\boldsymbol{b}_e \in \mathbb{R}^D$ are parameters for the dense layer. We also keep the original modality characteristics by using $(1 + s)$.

7.3.5 Contextual Attention Mechanism

Given a sequence of items $\mathcal{J} = \{i_1, i_2, \dots, i_t\}$ that the user u interacted with and ordered according to time, where t represents the current time stamp. Recall that it represents the fusion embedding of item it. Let i_t be a context matrix consisting of recent w_c inputs, where w_c is the window width of the context. To learn user's current representation considering the contextual effects, one can simply average all the representations of his/her clicked items within the contextual window:

$$\tilde{r}_t = \frac{1}{w_c} \sum_{j=t-w_c+1}^{t} \boldsymbol{i}_j \tag{15}$$

However, user's interests are full of stochasticity and contingency, which means a user might accidentally click on wrong items or s/he is attracted by some unrelated items due to curiosity. And we argue that time plays the key role on the user's next possible behaviour. For instance, one would like to watch detective or horror movies on Friday and Saturday but might prefer comedies on other days during a week. Besides, the time interval between item i_t and item i_j, where $j < t$, also matters. Normally, events with fewer time intervals with respect to the current time have greater impact on current behaviour. Thus, apart from the representations of items within contextual window, our contextual attention mechanism also considers two other factors[28] as shown in Figure 7.4.

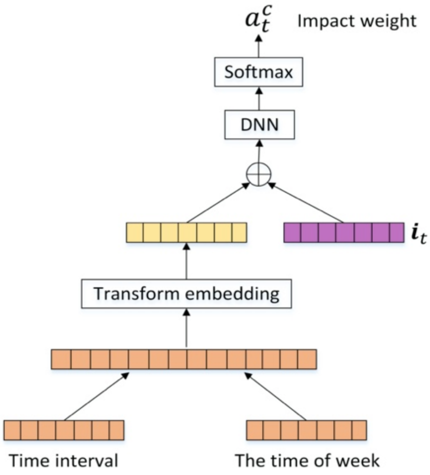


Figure 7.4: Diagram of contextual attention network.

- **Time of Week T_w** is the time within a week measured by hour. Specifically, we divide one week into 24 × 7 = 168h ordered from Monday to Sunday, and adopt a vector with 169 dimensions (the first 168 dimensions are for each hour of a week and the last one is for everything older than that) to embed the time of week T_w. If a user clicked an item at such as 00:10 on Monday, then this event belongs to the first hour and the value in the first dimension of the vector will be set to 1. If an event was happened out of 168h, the 169th dimension will be set to 1.

[28] It is worth noting that other kinds of factors such as location can also be considered and according to the same transformation mechanism but it is beyond the scope of this thesis.

- **Time Interval $\boldsymbol{\delta_t}$** is the time difference between the user's historical behaviour and the current time. Similar with T_w, we apply a 169-dimensional vector and the first 168 dimensions represent that the time intervals between the timestamp of previous clicked item and the current timestamp are within 0 to 168h. The 169th dimension represents everything happened older than 168h. In this way, we can explore how time difference affects the user's next behaviour.

For each context vector $\boldsymbol{i}_j, j \in [t - w_c + 1, t]$ in $\boldsymbol{C}_i^t$, we can obtain its corresponding representation of $\boldsymbol{T}_{w,j}$ and $\boldsymbol{\delta}_{t_j}$. To learn the two factors and item's representation $\boldsymbol{i}_j$ together, one ordinary way is the simple concatenation strategy as $[\boldsymbol{i}_j;\, \boldsymbol{T}_{w,j}\,;\, \boldsymbol{\delta}_{t_j}]$. However, we argue that factor embedding and item embedding are learned differently, which means they are in different representation space. Thus, we introduce the transformed embeddings

$$\boldsymbol{i}_j^* = g([\boldsymbol{T}_{w,j}\,;\, \boldsymbol{\delta}_{t_j}]) \tag{16}$$

where $g(\cdot)$ is the transformation function, and can be either linear

$$g\left(\left[\boldsymbol{T}_{w,j}\,;\, \boldsymbol{\delta}_{t_j}\right]\right) = \boldsymbol{W}_f([\boldsymbol{T}_{w,j}\,;\, \boldsymbol{\delta}_{t_j}]) \tag{17}$$

or non-linear

$$g\left(\left[\boldsymbol{T}_{w,j}\,;\, \boldsymbol{\delta}_{t_j}\right]\right) = sigmoid(\boldsymbol{W}_f\left(\left[\boldsymbol{T}_{w,j}\,;\, \boldsymbol{\delta}_{t_j}\right]\right) + \boldsymbol{b}_f) \tag{18}$$

where $\boldsymbol{W}_f \in \mathbb{R}^{D\times 338}$ $(338 = 2 \times 169)$ is the trainable transformation matrix and $\boldsymbol{b}_f \in \mathbb{R}^{D\times 1}$ is the trainable bias. Since the transformation is continuous, it can map factor embeddings to item space while preserving their original relationship. We therefore can concatenate these two embeddings as $\tilde{\boldsymbol{i}}_j = [\boldsymbol{i}_j^*; \boldsymbol{i}_j]$.

After that, we perform the following attention mechanism:

$$a_j^c = softmax(\mathcal{G}(\tilde{\boldsymbol{i}}_j)) = \frac{\exp\,(\mathcal{G}(\tilde{\boldsymbol{i}}_j))}{\sum_{t-w_c+1\le k\le t} \exp\,(\mathcal{G}(\tilde{\boldsymbol{i}}_k))} \tag{19}$$

where $\mathcal{G}$ is a deep neural network regarded as attention network and $softmax(\cdot)$ is the softmax function to calculate the normalized impact weight. The attention network $\mathcal{G}$ receives concatenation embedding as input and outputs the impact weight. Finally the embedding of user's current representation can be calculated as weighted summation of all item embeddings within the contextual window:

$$\tilde{\boldsymbol{r}}_t = \sum_{t-w_c+1\leq k\leq t} a_k^c \boldsymbol{i}_k \tag{20}$$

We will demonostrate the efficacy of the attention network in the experiment section.

7.3.6 Ante-RNN Model

Long Short-Term Memory (LSTM) is a special form of RNN, widely used to model sequence data. LSTM uses input gate, forget gate and output gate vectors at each position to control the passing of information along the sequence and thus improves the modelling of long-range dependencies. Gated Recurrent Unit (GRU) is the simplified version of LSTM networks but still maintains all their properties (Cho et al. [275]). In GRU unit, the activation h_{t-1} at time t is a linear interpolation between the previous activation $\widetilde{h}_t$ and the candidate activation $\tilde{\boldsymbol{r}}_t$. After we get the output vector from contextual attention layer as the input to the GRU layer, the following intermediate calculations can be achieved recursively during model learning procedure:

$$\begin{aligned} \boldsymbol{z}_t &= \sigma\,(\boldsymbol{W}_z\tilde{\boldsymbol{r}}_t + \boldsymbol{U}_z\,\boldsymbol{h}_{t-1}) \qquad (21)\\ \boldsymbol{r}_t &= \sigma\,(\boldsymbol{W}_r\tilde{\boldsymbol{r}}_t + \boldsymbol{U}_r\,\boldsymbol{h}_{t-1})\\ \widetilde{\boldsymbol{h}}_t &= tanh\,(\boldsymbol{W}_c\tilde{\boldsymbol{r}}_t + \boldsymbol{U}_c(\,\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}))\\ \boldsymbol{h}_t &= (1-\boldsymbol{z}_t)\boldsymbol{h}_{t-1} + \boldsymbol{z}_t\widetilde{\boldsymbol{h}}_t \end{aligned}$$

where update gate $\boldsymbol{z}_t$ decides how much the unit updates its activation or content. $\boldsymbol{r}_t$ is a set of reset gate to control the flow of information, and $\odot$ is an element-wise multiplication. $\sigma(\cdot)$ and $\tanh(\cdot)$ are the element-wise logistic function and hyperbolic tangent function used to do non-linear projection. The length of the output vector $\boldsymbol{o}_t$ from GRU layer is the number of all candidate items, and a softmax layer is added after GRU layer to output the probability distributions of all candidate items.

Illuminated by the recent successes of probabilistic sequential translation model (Pan et al. [312]), given a set of user's interacted items $\mathcal{I}_u = \{i_1, i_2, \ldots, i_t\}$ and current user's interested topics $\boldsymbol{\eta}_t^u$ we formulate our recommendation problem as a coherence loss, where the log probability of the recommendation is given by the sum of log probabilities over the clicked items:

$$l(\boldsymbol{\eta}_t^u, \mathcal{I}_u) = -logPr(\mathcal{I}_u|\boldsymbol{\eta}_t^u) = \sum_{t=1}^{N_u} -logP(\boldsymbol{i}_t|\boldsymbol{\eta}_t^u, \boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_{t-1}; \Theta) \tag{22}$$

where $\{\boldsymbol{i}_1, \boldsymbol{i}_2, \dots, \boldsymbol{i}_{N_u}\}$ is the sequentially predicted items. Here, $\boldsymbol{i}$ is corresponding to the fused image and textual embedding. By performing our contextual attention mechanism, for each time stamp t, we can get the user interest embedding $\tilde{\boldsymbol{r}}_t \in \mathbb{R}^D$ as the GRU input, as shown in Figure 7.2. Θ is the set of parameters of our framework, including contextual attention and GRU layer, the parameters in image embedding network, topical attention network and multi-model fusion component. By minimizing the above loss, the user interest evolvement can be described dynamically, making the recommendation more coherent and reasonable. Then the above probabilities can be achieved through softmax classification function demonstrated below:

$$P(\boldsymbol{i}_t = p|\boldsymbol{\eta}_t^u, \boldsymbol{i}_1, \boldsymbol{i}_2, \dots, \boldsymbol{i}_{t-1}; \Theta) = \frac{\exp(\boldsymbol{W}_s^{(p)}\boldsymbol{h}_t)}{\sum_{j=1}^{|\mathcal{J}|}\exp(\boldsymbol{W}_s^{(j)}\boldsymbol{h}_t)} \tag{23}$$

where $|\mathcal{J}|$ is the number of candidate items, $\boldsymbol{W}_s$ is the parameter matrix of the softmax layer in our model.

Finally, we can obtain the objective function as:

$$\mathcal{L} = \sum_{u \in \mathcal{U}} l(\boldsymbol{\eta}_t^u, \mathcal{J}_u) + \lambda_1 \sum_{\mathcal{E},\mathcal{F}} l(\mathcal{E}, \mathcal{F}) + \lambda_2 \|\Theta\|_2^2 \tag{24}$$

where λ_1 is the trade-off parameters for these objectives. $\lambda_2 \geq 0$ is the coefficient of the weight decay term. Then, Ante-RNN can be learned by the stochastic gradient descent and BPTT. The parameters are automatically updated by Theano (Bergstra et al. [313]). By optimizing the above overall loss function in a unified framework, our proposed method achieves personalized dynamic recommendation with considering image-textual alignment, user's interested topics, multi-model fusion and contextual influence jointly.

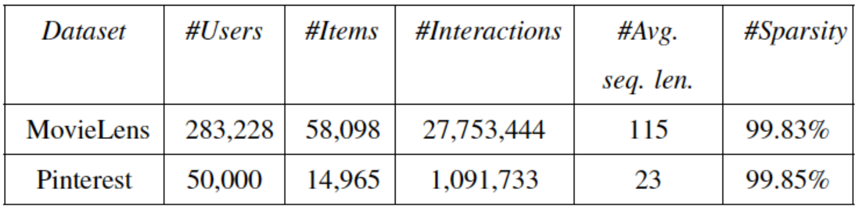
7.4 Experiments

In this section, we conduct our experiments on two real-world datasets. First, we introduce the datasets, evaluation metrics and baseline methods as well as parameter settings. Then we make comparison between Ante-RNN model and the baselines. After that, the recommendation efficiency and the effectiveness of the hybrid attention mechanism proposed in this Chapter will be tested, followed by the analysis on users with different sparsity level and various parameters. Finally, we illustrate the recommendation explainability.

7.4.1 Datasets

Experiments are conducted on two large scale datasets, namely Movielens [29] and Pinterest[30]. The basic statistics of them are listed in Table 7.3. For both datasets, we sort all user-item interaction pairs in the ascending interaction time order. The first 80% sequential histories are selected as training set and the rest 20% as test set. Besides, we randomly hold-out 10% interaction history of each user from training set as validation sets. To measure the statistical significance of Ante-RNN over the baselines, we repeat the splitting process five times (i.e., generating five pairs of training and validation sets). Averaged results are reported in the following subsection.

Table 7.3: Main properties of the experimental datasets



Dataset	*#Users*	*#Items*	*#Interactions*	*#Avg. seq. len.*	*#Sparsity*
MovieLens	283,228	58,098	27,753,444	115	99.83%
Pinterest	50,000	14,965	1,091,733	23	99.85%

MovieLens dataset contains 27,753,444 ratings from 283,228 users on 58,098 movies from January 09, 1995 to September 26, 2018. In order to mimic implicit data, we binarized all ratings independent of their values, considering them as positive feedback as it has been done by Rendle et al. [201]. Using the timestamps provided, we thus got an ordered sequence of consumption events for each user. The dataset contains only sequences with a minimum length of 20. The average sequence length is 115. We aimed at predicting the next movie to watch. In order to obtain the textual information and poster image corresponding to each movie, we downloaded descriptions and images according to *tmdbID* property provided in links.csv file through TMDb API[31].

Pinterest is one of the largest social curation networks. This dataset with implicit feedback is constructed by Geng et al. [202] for evaluating image recommendation. Due to the large volume and high sparsity of this dataset, for instance, over 20% of users have only one pin,

[29] http://files.grouplens.org/datasets/movielens/ml-latest-README.html

[30] https://sites.google.com/site/xueatalphabeta/academic-projects

[31] https://www.themoviedb.org/documentation/api

we filter the dataset by retaining the top 15,000 popular images and sampling 50,000 users who have interactions on these images. This results in a subset of data that contains 50,000 users, 14,965 images and 1,091,733 interactions. Each interaction denotes whether the user has pinned the image to his/her own board. Since there is no description information on images, we also collect corresponding descriptions by using Pinterest API[32].

A sample of the dataset can be accessible through the link[33], and the full version of our dataset is available on request.

7.4.2 Evaluation Metrics

Based on temporally ordered lists of pinned/rated items, our objective is to correctly predict the next item a target user will likely pin/rate. The ground truth at a particular time step is therefore represented by a single user-item tuple. To present the user with adequate recommendations, the target item should be among the top few recommended items. Since we are interested in measuring top-K recommendation instead of rating prediction, we measure the quality by looking at the *Recall@K* and *NDCG@K*, which are widely used for evaluating top-K recommender systems.

- Recall@K is defined as the fraction of cases where the item actually consumed in the next event is among the top K items recommended [314].
- NDCG@K (Normalized Discounted Cumulative Gain) is adopted to evaluate ranking performance by taking the positions of the correct items into consideration [315], and thus to assess if the items that a user has actually consumed are ranked in higher positions in the recommendation list.

We set $K = 20$ as it appears desirable from a user's perspective to expect the target among the first 20 items (Hidasi et al. [55]).

7.4.3 Baselines

To validate the effectiveness of Ante-RNN, we compared our model with the following methods. Note that all model-based Collaborative Filtering approaches are learned by optimizing the same pairwise ranking loss of Bayesian Personalized Ranking (BPR) for a

[32] https://developers.pinterest.com/docs/api/

[33] https://www.dropbox.com/sh/hinouvmaj7lginn/AABpgBifZLQBYrHLHaVzzSUQa?dl=0

fair comparison. BPR will be introduced in detail below.

- BPR[34]: This method optimizes the latent factor model with a pairwise ranking loss, which is tailored to learn from implicit feedback. It is a highly competitive and popular baseline for item recommendation [201]. We adopt matrix factorization as the prediction component for BPR.
- VBPR[35]: The Visual Bayesian Personalized Ranking (VBPR) model is a state-of-the-art method for recommendation leveraging item visual images [192].
- CTR[36]: Collaborative Topic Regression (CTR) learns interpretable latent structure from user generated contents so that probabilistic topic modelling can be integrated into collaborative filtering [316].
- GRU[37]: It is the state-of-the-art sequential recommendation method, and an extension of RNN for capturing the long-term dependency [53]. GRU is also the basic of our Ante-RNN model.
- IARN[38]: Interacting Attention-gated Recurrent Network (IARN) model proposed by Pei et al. [288] integrates an attention mechanism into BRNN when modelling both user and item representations for the sequential recommendation. Then the inner product of user and item representations is performed to predict user ratings.
- MLAM [39] : The Multi-level Attraction Model (MLAM) is a state-of-the-art interpreterable recommendation algorithm, which leverages attention-based multi-level contextual information for Top-K recommendation and meanwhile provides explanations [282]. In our situation, we apply image features instead of the cast level module and then build attractions over them.
- MV-RNN[40]: Multi-View Recurrent Neural Network (MV-RNN) proposed by Cui et al. [9] is a newly proposed algorithm especially for sequential recommendations. Similarly, it incorporates visual and textual information to deal with cold start issue

[34] https://github.com/gamboviol/bpr

[35] https://sites.google.com/a/eng.ucsd.edu/ruining-he/

[36] https://github.com/blei-lab/ctr

[37] https://github.com/LaceyChen17/DREAM

[38] https://github.com/wenjiepei/IARN

[39] https://github.com/rainmilk/ijcai18mlma

[40] https://github.com/cuiqiang1990/MV-RNN

and meanwhile applies a recurrent structure to dynamically capture the users' interests. Differently, they do not consider time factors between user's historical interactions and they use a denoising autoencoder for multi-modality fusion.

Besides, we also adopt two variations of our Ante-RNN model, namely **t-Ante-RNN** and **v-Ante-RNN**. In former model, we only keep text description of item as input and remove all modules that are related to image processing to perform recommendations, whereas the latter one only leverages images as model inputs and modules with respect to text processing are excluded when generating Top-N rank list. Other two variations of Ante-RNN are **Ante-RNN-D**, **Ante-RNN-N** represent Ante-RNN with direct fusion and neural fusion respectively, while we use Ante-RNN to represent Ante-RNN with attention fusion for it achieves the best performance of all fusion methods.

7.4.4 Parameter Settings

For image embedding of Ante-RNN model, we use Faster R-CNN in conjunction with ResNet-101 pre-trained by Anderson et al. [186] to extract Region Of Interests (ROIs) for each image. The Faster R-CNN implementation uses an intersection over union (IoU) threshold of 0.7 for region proposal suppression, and 0.3 for object class suppression. The class detection confidence threshold is set as 0.2 to select salient image regions, and top 36 ROIs with highest confidence scores are selected. We extracted features after average pooling, resulting in the final representation of 2048 dimensions. The embedding dimension D is set to 128. Topic numbers ψ is set to 70 and 100 for MovieLens and Pinterest datasets respectively. For time decay rate λ, we set it to 0.2 for MovieLens dataset, but a relatively slow decay $\lambda = 0.1$ for Pinterest dataset. In the model training phase, the trade-off parameter λ_1 is set to 0.2 by grid-search over $\{0.2, 0.4, 0.6, 0.8\}$. The coefficient λ_2 of weight decay term is set to 0.0001. The contrastive margin $\alpha 1$ is set to 0.3. Learning rate is set to 0.001. The window sizes wc are set as 5 and 3 for MovieLens and Pinterest dataset respectively.

The hyper-parameters of each baseline are tuned with the validation set during training phase. Specifically, the dimension of latent factors (or embedding size) is set to 128 for baselines. The regularization coefficient is set to 10 that works best for BPR and VBPR. We set α of MLAM to 1, 4 and 2 for image, word and sentence level attention model. Optimization for baselines terminate until convergence or 150 learning epochs. Other parameters are set the same with our model if not specified.

7.4.5 Performance Evaluation

Table 7.4: Performance comparison (Mean ± Standard Deviation) w.r.t. Recall@K and NDCG@K (K=5, 10, 15, 20, 25) on two datasets (MovieLens and Pinterest). "*" indicates that the improvements of our model over the best baseline are statistically significant for p-value < 0.01 with paired t-test.

MovieLens	Recall@5	Recall@10	Recall@15	Recall@20	Recall@25	NDCG@5	NDCG@10	NDCG@15	NDCG@20	NDCG@25
BPR	0.128 ± 0.021	0.154 ± 0.012	0.172 ± 0.011	0.199 ± 0.016	0.215 ± 0.014	0.083 ± 0.009	0.096 ± 0.008	0.102 ± 0.011	0.114 ± 0.021	0.127 ± 0.012
CTR	0.189 ± 0.024	0.206 ± 0.013	0.223 ± 0.019	0.237 ± 0.021	0.251 ± 0.018	0.096 ± 0.011	0.103 ± 0.009	0.115 ± 0.015	0.128 ± 0.006	0.142 ± 0.013
VBPR	0.211 ± 0.018	0.226 ± 0.023	0.240 ± 0.021	0.256 ± 0.015	0.274 ± 0.019	0.102 ± 0.013	0.116 ± 0.012	0.129 ± 0.019	0.141 ± 0.011	0.153± 0.016
GRU	0.261 ± 0.031	0.274 ± 0.026	0.298 ± 0.019	0.311 ± 0.021	0.326 ± 0.024	0.133 ± 0.011	0.142 ± 0.016	0.157 ± 0.009	0.173 ± 0.018	0.185 ± 0.014
IARN	0.282 ± 0.016	0.297 ± 0.022	0.316 ± 0.034	0.331 ± 0.023	0.343 ± 0.021	0.157 ± 0.015	0.163 ± 0.011	0.179 ± 0.019	0.192 ± 0.008	0.208 ± 0.017
MLAM	0.309 ± 0.033	0.321 ± 0.026	0.342 ± 0.029	0.358 ± 0.018	0.371 ± 0.013	0.174 ± 0.011	0.186 ± 0.014	0.203 ± 0.012	0.217 ± 0.017	0.232 ± 0.015
MV-RNN	0.326 ± 0.023	0.338 ± 0.018	0.361 ± 0.019	0.375 ± 0.022	0.389 ± 0.025	0.191 ± 0.014	0.204 ± 0.011	0.225 ± 0.009	0.231 ± 0.015	0.253 ± 0.013
v-Ante-RNN	0.273 ± 0.026	0.291 ± 0.021	0.309 ± 0.023	0.326 ± 0.016	0.337 ± 0.020	0.146 ± 0.013	0.157 ± 0.011	0.162 ± 0.016	0.184 ± 0.012	0.203 ± 0.014
t-Ante-RNN	0.301 ± 0.018	0.315 ± 0.016	0.329 ± 0.019	0.347 ± 0.021	0.364 ± 0.022	0.162 ± 0.011	0.174 ± 0.015	0.193 ± 0.009	0.209 ± 0.014	0.226 ± 0.011
Ante-RNN-D	0.342 ± 0.013	0.361 ± 0.017	0.384 ± 0.008	0.393 ± 0.016	0.407 ± 0.014	0.208 ± 0.014	0.221 ± 0.012	0.239 ± 0.017	0.252 ± 0.013	0.275 ± 0.016
Ante-RNN-N	0.359 ± 0.008	0.385 ± 0.016	0.401 ± 0.021	0.416 ± 0.014	0.429 ± 0.017	0.220 ± 0.013	0.238 ± 0.018	0.251 ± 0.007	0.265 ± 0.016	0.286 ± 0.014
Ante-RNN	**0.365 ± 0.006***	**0.389± 0.013***	**0.408± 0.016***	**0.421± 0.015***	**0.436± 0.011***	**0.227± 0.012***	**0.246± 0.007***	**0.263± 0.016***	**0.272± 0.021***	**0.298± 0.015***

Pinterest	Recall@5	Recall@10	Recall@15	Recall@20	Recall@25	NDCG@5	NDCG@10	NDCG@15	NDCG@20	NDCG@25
BPR	0.056 ± 0.018	0.073 ± 0.014	0.085 ± 0.011	0.094 ± 0.016	0.107 ± 0.023	0.032 ± 0.013	0.039 ± 0.011	0.047 ± 0.009	0.052 ± 0.014	0.058 ± 0.018
CTR	0.071 ± 0.011	0.083 ± 0.014	0.092 ± 0.017	0.106 ± 0.012	0.124 ± 0.015	0.039 ± 0.012	0.048 ± 0.016	0.053 ± 0.015	0.062 ± 0.019	0.067 ± 0.021
VBPR	0.078 ± 0.013	0.092 ± 0.024	0.103 ± 0.018	0.114 ± 0.012	0.135 ± 0.016	0.042 ± 0.008	0.053 ± 0.014	0.059 ± 0.017	0.064 ± 0.011	0.071 ± 0.019
GRU	0.109 ± 0.024	0.120 ± 0.013	0.131 ± 0.011	0.142 ± 0.016	0.153 ± 0.021	0.058 ± 0.012	0.061 ± 0.011	0.066 ± 0.016	0.071 ± 0.014	0.076 ± 0.012
IARN	0.113 ± 0.016	0.126 ± 0.012	0.139 ± 0.020	0.152 ± 0.017	0.164 ± 0.014	0.061 ± 0.008	0.067 ± 0.012	0.072 ± 0.011	0.078 ± 0.017	0.085 ± 0.014
MLAM	0.151 ± 0.019	0.173 ± 0.021	0.186 ± 0.017	0.201 ± 0.013	0.218 ± 0.014	0.079 ± 0.010	0.085 ± 0.014	0.093 ± 0.013	0.102 ± 0.016	0.114 ± 0.020
MV-RNN	0.175 ± 0.017	0.188 ± 0.015	0.207 ± 0.008	0.219 ± 0.012	0.237 ± 0.011	0.093 ± 0.016	0.101 ± 0.014	0.108 ± 0.019	0.119 ± 0.021	0.130 ± 0.017
t-Ante-RNN	0.136 ± 0.017	0.142 ± 0.023	0.157 ± 0.018	0.169 ± 0.015	0.183 ± 0.013	0.068 ± 0.011	0.074 ± 0.016	0.079 ± 0.016	0.091 ± 0.013	0.098 ± 0.011
v-Ante-RNN	0.141 ± 0.022	0.154 ± 0.015	0.162 ± 0.017	0.177 ± 0.016	0.197 ± 0.019	0.071 ± 0.010	0.076 ± 0.007	0.083 ± 0.015	0.095 ± 0.013	0.103 ± 0.016
Ante-RNN-D	0.202 ± 0.018	0.216 ± 0.022	0.229 ± 0.015	0.245 ± 0.011	0.258 ± 0.014	0.109 ± 0.012	0.122 ± 0.015	0.126 ± 0.011	0.137 ± 0.019	0.154 ± 0.013
Ante-RNN-N	0.218 ± 0.012	0.231 ± 0.016	0.241 ± 0.016	0.257 ± 0.013	0.269 ± 0.018	0.124 ± 0.008	0.136 ± 0.014	0.141 ± 0.012	0.149 ± 0.014	0.166 ± 0.012
Ante-RNN	**0.223 ± 0.011***	**0.238± 0.014***	**0.252± 0.008***	**0.264± 0.015***	**0.276± 0.013***	**0.129± 0.010***	**0.145± 0.008***	**0.148± 0.016***	**0.162± 0.014***	**0.181± 0.014***

The performance of Ante-RNN and the baselines are reported in terms of Recall@K and NDCG@K on two kinds of datasets in Table 7.4. K ranges over {5, 10, 15, 20, 25}. From the results, we can see that: (1) The performance of BPR fails to surpass the rest baseline models since that the latter ones integrate either visual or text features into their modelling process. This observation verifies that side information,e.g.image or text, is complementary to ratings/ implicit feedbacks and thus can help to improve recommendation performance in real-world applications. Furthermore, by incorporating both visual and textual information, MLAM, MV-RNN and our Ante-RNN models obtain the best performance among all comparison methods. (2) It is worth noting that different side information takes on different importance for different datasets. For instance, t-Ante-RNN achieves better performance than v-Ante-RNN on MovieLens while performs worse than v-Ante-RNN on

Pinterest. The reason may be that for movie recommendations, the users pay more attention to the plot and descriptions on movies instead of posters, while users on Pinterest focus more on images than other side information. (3) For the baselines, neural recommendation algorithms, namely MLAM, IARN, GRU, MV-RNN, Ante-RNN and its variations, greatly perform better than the other baselines for that they can either better learn the latent features of items or better model user's dynamic interests over time from sequential inputs. Among these, the performance of MV-RNN is better than MLAM and IARN which verifies the importance of both capturing user's sequential patterns and integrating multiple side information. (4) Our Ante-RNN model outperforms over the baselines on all datasets and evaluation measures by combining visual and textual information into representation learning process. Furthermore, the performance of Ante-RNN is better than MV-RNN because the hybrid attention mechanism also helps to model user's long and short-term dynamic preferences. On MovieLens, it outperforms the best baseline MV-RNN by 4.6% on Recall@20 and 4.1% on NDCG@20, and much higher than the other baseline models.

Besides, we also analyze Ante-RNN with three fusion methods. From the table, we can observe that Ante-RNN-N always beats the Ante-RNN-D. It is because the non-linear transformation boosts the interactions among multi-modalities which leads to a better fusion. The best performance appears with attention fusion but the advantage is not prominent. It indicates that the attention mechanism is more likely able to better capture the different importance of multiple input features.

7.4.6 Recommendation Efficiency

In addition to the advantage of recommendation accuracy, we have also evaluated the efficiency of Ante-RNN on both datasets. Table 7.5 shows the runtime comparison with GRU, IARN and MV-RNN. Other baselines are not listed here as the implementation cannot leverage the computation power of GPU. Experiments were conducted on a machine with a NVIDIA TITAN X Pascal GPU. From this results, we observe that Ante-RNN is comparable with other state-of-the-art approaches not utilizing the image information. Moreover, due to the efficient sampling strategy for image-text alignment, our method converges faster than MV-RNN which integrates image and text features by using autoencoder.

During prediction process, given user clicking item i_t at time stamp t, the image embedding with textual alignment v and word representations $\{\boldsymbol{e}_1, \dots, \boldsymbol{e}_N\}$ of its corresponding description can be achieved beforehand. For user u, the user interested topics embedding $\boldsymbol{\eta}_t^u$ can also be derived separately according to a certain time interval, hour for instance according to Eq. 7. Therefore, the actual online prediction can be accelerated by only performing basic matrix operations with GPU.

Table 7.5: Runtime comparison (seconds) for training model on both datasets.

Dataset	*GRU*	*IARN*	*Ante-RNN*	*MV-RNN*
MovieLens	3725.03	4306.82	4913.67	12416.51
Pinterest	918.34	1150.96	1431.29	3504.02

7.4.7 Effect of Attention Mechanism

To get a better understanding of our Ante-RNN model, we further evaluate the key component - topical (T) and contextual (C) attention mechanisms. In order to prove the importance of time factors, we also evaluate two variations of contextual attention mechanism, $C - \boldsymbol{T}_w$ and $C - \boldsymbol{\delta}_t$, by removing the time of week or the time interval parameter from contextual attention network. Table 7.6 shows the effect of our basic Ante-RNN model with or without attention mechanism(s) for $K = 20$. Note that: when we consider neither topical attention or contextual attention mechanism, it means we only adopt image embedding fused with textual embedding as GRU input for model learning, and the text embedding is the average of word representation in the text. From the table, we can observe that:

(1) When both topical and contextual attention mechanisms are applied, the recommendation performance is improved compared with the other combinations. The good performance of attention mechanism shows that the characteristics of user's long-term and short-term interests are reflected at both levels.

(2) The contextual attention mechanism contributes more for our model on two datasets as compared to topic-based attention mechanism since the performance of our model deteriorates more without contextual attention component. This may be due to the fact that the contextual attention method can strengthen the user's short-term interest modelling which GRU may lack, and capture the user's main focus during a limited time period, while the topic-based attention mechanism can assist GRU to model user's long-term interest pattern in a better way, which also leads to the improvement of recommendation performance compared with the model without topic-based attention. Furthermore, two

kinds of time factors integrated in contextual attention method further strengthen the discriminating ability of user's short-term focus.

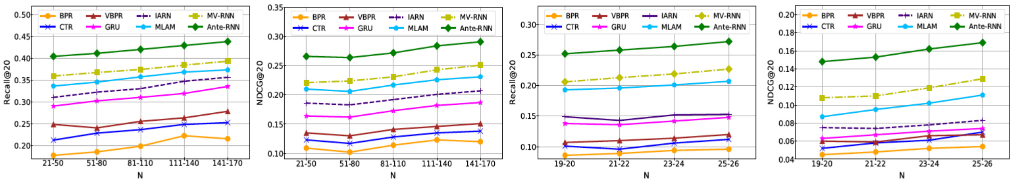
(3) When time interval is removed from contextual attention network, the recommendation performance deteriorates more than the contextual attention without time of week. For example, comparing to the attention network C, the performance degradation of $C - \boldsymbol{\delta_t}$ is 1.9% and 2.1% on Recall@20 in MovieLens and Pinterest datasets respectively, while the performance degradation of $C - \boldsymbol{T_w}$ is 0.8% and 0.7% correspondingly. It demonstrates that time interval is more important to capture the user's short-term interest compared with time of week.

Table 7.6: Effect of topical (T) and contextual (C) attention mechanisms as well as their variations w.r.t. Recall@20 and NDCG@20. "*" indicates the statistical significance for *p*-value < 0.01.

Model	Attention Type	MovieLens		Pinterest	
		Recall@20	*NDCG@20*	*Recall@20*	*NDCG@20*
Ante-RNN	None	0.364	0.228	0.205	0.112
	T	0.385	0.246	0.227	0.133
	C-$\boldsymbol{T_w}$	0.398	0.253	0.246	0.147
	C-$\boldsymbol{\delta t}$	0.387	0.249	0.232	0.136
	C	0.406	0.259	0.253	0.151
	T+C-$\boldsymbol{T_w}$	0.414	0.268	0.258	0.159
	T+C-$\boldsymbol{\delta t}$	0.409	0.261	0.251	0.148
	T+C	**0.421***	**0.272***	**0.264***	**0.162***

7.4.8 Analysis on Users with Different Sparsity Levels

In this section, we study the impact of different sequence lengths on the recommendation performance. Note that we do not retrain our model with different sets of users, instead we divide the test set into different groups by the number of items per user. The results are shown in Figure 7.5, and we have the following observations:



(a) MovieLens-Recall@20 (b) MovieLens-NDCG@20 (c) Pinterest-Recall@20 (d) Pinterest-NDCG@20

Figure 7.5: Performance of Recall@20 and NDCG@20 w.r.t. the number of items per user on two datasets.

(1) As sequence length increases, the performance of all methods generally improves, indicating that sufficient temporal context could ensure models that capture user's interest patterns in a better way. This also explains why the overall performance on MovieLens dataset is better than that on Pinterest.

(2) Overall, Ante-RNN achieves the best performance across all different configurations of all the datasets, especially, when the sequence length gets larger. On average, the relative improvements w.r.t. the second best method are 4.5% with training records length of 21–50 and 4.3% with training records length of 25–26 on Recall@20 in MovieLens and Pinterest datasets respectively. This implies the remarkable advantage of Ante-RNN in dealing with long sequences. Besides, we also find that when the number of items per user is relatively small, Ante-RNN still keeps the advantages in performance, which indicates that the hybrid attention mechanism and visual information integration could improve the recommendation quality when there is insufficient training data for each user.

7.4.9 Parameter Analysis

In this section, we analyse the influence of the embedding size D and window length w_c in the contextual attention mechanism to the performance of our Ante-RNN model.

Analysis of embedding size D. The empirical results displayed in Figure 7.6 indicate the substantial influence of embedding size upon Ante-RNN and other baselines. During experiments, we range D in $\{32, 64, 128, 256, 512\}$ and fix other hyper-parameters to plot the corresponding results for $K = 20$ with respect to Recall@K and NDCG@K on two datasets. Similar trends can be found on all models of both datasets. On one hand, the recommendation performance improves along with the increasing of dimensionality, which means that the representations hold more and informative resources extracted from users and items. On the other hand, the performance of recommendations will drop when the dimensionality continues raising, which demonstrates that the models may suffer from over-fitting problem. It is worth noting that the performance of our Ante-RNN model only slightly deteriorates compared with other methods for that our model holds a higher stability for the changes of dimensionality.

Analysis of window size w_c. In this part, we investigate the best window size w_c for Ante-RNN. The window size wc ranges from 1 to 6 with other hyper-parameter fixed. When w_c is set as 1, it can be considered as the special case without contextual attention mechanism in our experimental cases. Figure 7.7 shows the performance results for with respect to

Recall@K and NDCG@K on MovieLens and Pinterest datasets. For both datasets, slightly difference can be observed on Recall@20 when $w_c > 2$ while there is an obvious difference on NDCG@20. We can also observe that the best window size can be chosen as $w_c = 3$ and $w_c = 5$ on Pinterest and MovieLens respectively. The superior window size on MovieLens is larger than that of Pinterest, which may be because the average sequence length on MovieLens is longer.

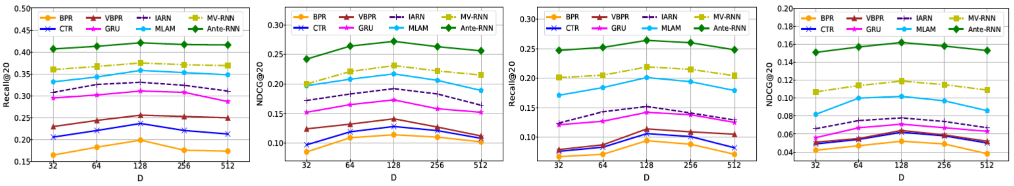


Figure 7.6: Performance of Recall@20 and NDCG@20 w.r.t. the embedding size $D \in [32, 64, 128, 256, 512]$ on two datasets.

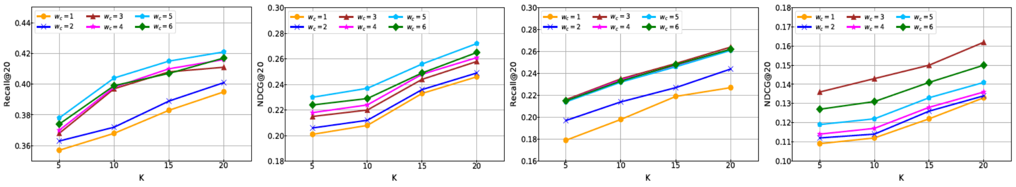


Figure 7.7: Performance of Recall@20 and NDCG@20 w.r.t. the window size $w_c \in [1, 2, 3, 4, 5, 6]$ on two datasets.

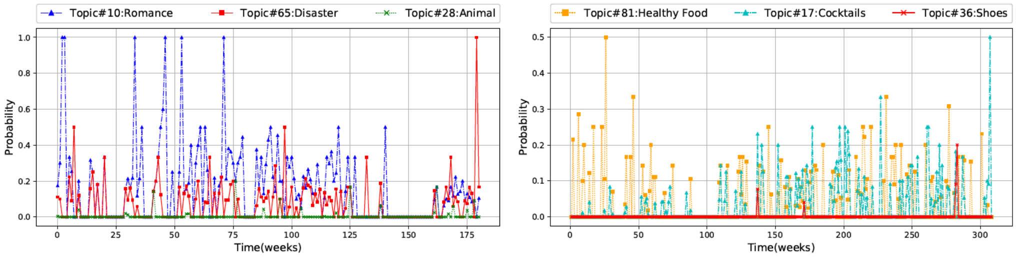
7.4.10 Recommendation Explainability

In this section, we evaluate the explanations generated by Ante-RNN from both qualitative and quantitative perspectives based on the Movielens and Pinterest datasets.

Qualitative evaluation. To provide better intuitions for the generated multi-model explanations of our recommendation results and to provide a better understanding of our hybrid attention mechanism, we present and analyze two examples learned by the model in a qualitative manner. We also compare our method with MLAM, a state-of-the-art explainable recommendation algorithm on two datasets. The examples are shown in Figure 7.9. In particular, we show one user for each dataset with their topic historical information on the left side. The user's recent pinned/rated four items are displayed according to time order as well as the topics that they belong to and their corresponding top-4 topic words extracted from item descriptions. The number on top of each image represents the weight calculated from contextual attention layer and higher value means that the item is more important in next recommendation task. When the model has predicted the next possible item i_{t+1}, the text description of i_{t+1} will be compared with user's interested topics and related topic words are tagged with red in our examples. Then, the highlighted regions in red square of item images are determined by performing Eqs. (1) and (2).

The changes of user's interested topic distribution across different weeks of two datasets are shown in Figure 7.8. Here we only select 3 representative topics appeared in users' recent historical records to illustrate the dynamic nature of users' interests, or else there will be too many lines tangled in one figure. We can see that, for example, on MovieLens, user 1 shows his/her long-term interests on topic #10 about "Romance" movies (blue line in Figure 7.8(a)) which frequently occurred in his/her historical records. However, the user's current interests shift to topic #65 "Disaster" movies (red line in Figure 7.8(a)) and topic #28 "Animal" movies (green line in Figure 7.8(a)). Our model can capture user's real-time interests through the dynamic contextual attention mechanism and recommend "Disaster" related movie. Some of the highlighted topic words i.e. storm and seas, can also be found in user's visited items. However, though MLAM can also provide explanations on image and its description separately, they cannot align highlighted image region together with its significant words. Besides, MLAM thinks the major interest of user 1 is "Romance" movies (topic #10) and thus recommends *Remember Me* instead, which verifies that our model can capture users' dynamic preferences, whereas MLAM can only model static users' interests.

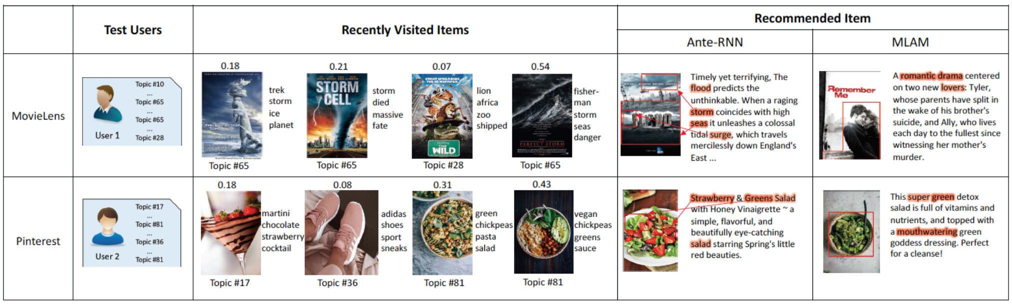
On Pinterest dataset, our model demonstrates the ability of considering both long and short-term interests when recommending items. Specifically, user 2 shows stable interest on topic #81 "Healthy Food" as the yellow line in Figure 7.8(b) signifies, while she also shows the recent active interest on topic #17 "Cocktail" with cyan dotted line. Consequently, the recommended item shows the combination features on both "Healthy Food" and "Cocktail" with highlighted topic words of strawberry, greens and salad. Meanwhile, greens and strawberry are marked in image to show the focuses of user's interests. Although MLAM also recommends "Healthy Food" related item, it still prefers the most frequently occurred items and no alignment can be found between visual and textual information.



(a) User1@ MovieLens

(b) User2@ Pinterest

Figure 7.8: User's interested topic distribution across dierent weeks on two datasets.


Test Users
Recently Visited Items
Recommended Item
Ante-RNN
MLAM
MovieLens
User 1
Topic #10
...
Topic #65
...
Topic #65
...
Topic #28
0.18
trek
storm
ice
planet
Topic #65
0.21
storm
died
massive
fate
Topic #65
0.07
lion
africa
zoo
shipped
Topic #28
0.54
fisher-
man
storm
seas
danger
Topic #65
Pinterest
User 2
Topic #17
...
Topic #81
...
Topic #36
...
Topic #81
0.18
martini
chocolate
strawberry
cocktail
Topic #17
0.08
adidas
shoes
sport
sneaks
Topic #36
0.31
green
chickpeas
pasta
salad
Topic #81
0.43
vegan
chickpeas
greens
sauce
Topic #81

Timely yet terrifying, The flood predicts the unthinkable. When a raging storm coincides with high seas it unleashes a colossal tidal surge, which travels mercilessly down England's East ...

A romantic drama centered on two new lovers: Tyler, whose parents have split in the wake of his brother's suicide, and Ally, who lives each day to the fullest since witnessing her mother's murder.

Strawberry & Greens Salad with Honey Vinaigrette ~ a simple, flavorful, and beautifully eye-catching salad starring Spring's little red beauties.

This super green detox salad is full of vitamins and nutrients, and topped with a mouthwatering green goddess dressing. Perfect for a cleanse!

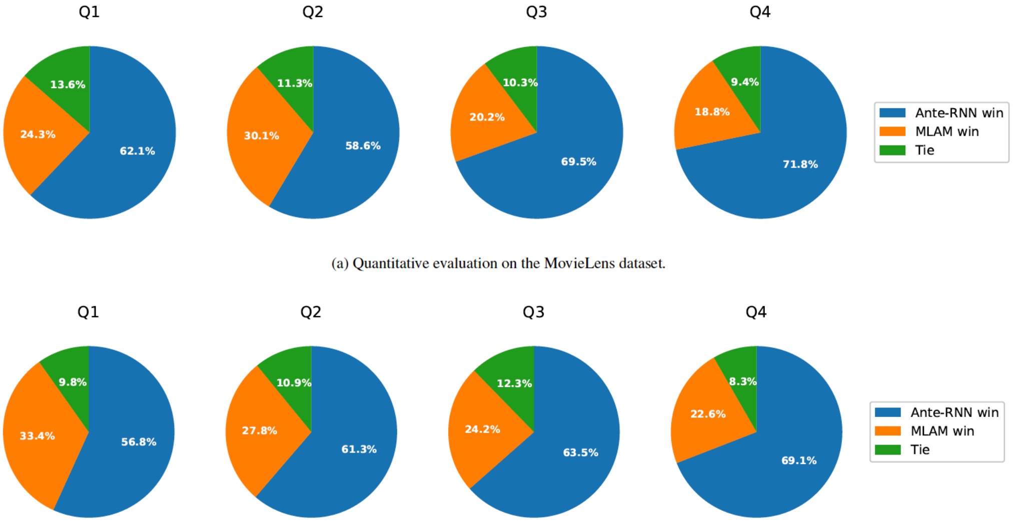
Figure 7.9: Examples of the visual and textual explanations.

Quatitative evaluation. To quantitatively evaluate our model's explainability, we conduct crowd-sourcing evaluation by comparing our model with MLAM. Specifically, we select the top-100 most active users from the two datasets separately. For each of the users, we present the image and its corresponding text description of the items that the user previously clicked for the worker to read. The workers are expected to infer this user's personalized preference from these information. Then they will be asked several questions to compare the recommendations and explanations generated by our model and MLAM model. Based on discussions in Tintarev et al. [317], we carefully designed the survey questions to evaluate different aspects of the recommender algorithm as follows:

- **Q1**: Which recommendation are you more satisfied with?
- **Q2**: Which model could provide you with more ideas about the recommended item?
- **Q3**: Which recommended item are you more likely to click after receiving an explanation?
- **Q4**: Based on the recommended items, which model generated explanation could help you know more easily and clearly why we recommend it to you?

For each question, the workers are required to choose from three options (i.e., A:Ante-RNN, B:MLAM, C:Tie). We intend to use Q1, Q2, Q3 to evaluate satisfaction, effectiveness, and persuasiveness of an explainable recommender algorithm, and use Q4 to judge if our attention mechanism is more effective in this problem.

To perform more accurate evaluations, we recruit 3 workers through Amazon Mechanical Turk for each user's case, and one result is valid only when more than 2 workers share the same opinion. Besides, we require the workers to come from an English-speaking country, older than 18 years, and have online entertainment experience for involving a more diverse population of users. The statistical results are shown in Figure 7.10. From the results, we can see that our proposed model apparently outperforms MLAM in all aspects of user study. Moreover, the results in Q3 and Q4 manifest that the explanations generated by our model's attention weights could promote the persuasiveness and satisfaction of the recommender algorithm, which verifies the effectiveness of our designed attention mechanism.



(a) Quantitative evaluation on the MovieLens dataset.

(b) Quantitative evaluation on the Pinterest dataset.

Figure 7.10: Results of the quantitative evaluation.

7.4.11 Limitations

We demonstrated that the Ante-RNN model is able to generate both multi-modal and adaptive explanations with recommendation performance comparable to the state-of-the-art methods (Table 7.4). Yet there are still some limitations: (1) Ante-RNN uses the Faster R-CNN model in conjunction with ResNet-101 pre-trained by Anderson et al. [186] to learn image region representations. However, not all the images in the dataset are regular and easy to distinguish. Some of them were graffiti, selfies, or even just screenshots of smart phones. Simply adopting a pre-trained weight may cause deviations and inaccurate image-text matching. Moreover, the named entities that are involved in the images cannot be well aligned with text. For example, the ship in the first movie poster of Figure 7.1 cannot be aligned to "TITANIC" in its corresponding text description. Designing a fine-tuning strategy for the pre-trained model and incorporating knowledge graph into image-text alignment may help with the problem and such is left as a matter for future work. (2) Due to the gating mechanism of recurrent neural networks, our model cannot provide users with a direct and meaningful way to correct the recommendation process if they are unsatisfied with the results. Developing recommendation approaches that are more scrutable would be an interesting research topic and needs to be addressed in future work.

7.5 Conclusion

User preferences often evolve over time, and it is essential to model their temporal dynamics for recommendation tasks while providing explanations on them. In this Chapter, we presented an Attentive Recurrent Neural Network (Ante-RNN) for dynamic personalized recommendations. The proposed model allows combining visual image information with text descriptions for better recommendation. Furthermore, a novel hybrid attention mechanism is introduced to strengthen user's short-term preference modelling and capture user's long-term interest dynamics in a better way. The learned attention weights can in turn help to provide reasonable interpretations on recommendation results. We also explore different fusion methods for multi-modality integration. Extensive experiments on two real-world large scale datasets verify that our model can not only provide competitive recommendation performance, but also provide reasonable visual aligned with textual explanations for the recommended items.

Part V

Conclusions and Future Work

Chapter 8
Conclusions

In this chapter, we first summarize our main findings in this thesis, and then provide some promising directions for future research.

8.1 Contributions

With the growing influence of social networks to our daily life, recent years have witnessed a surge of research on social recommendation techniques for mitigating the information overload and improving the recommendation performance. The overall research goal of this thesis is to extract the latent contexts from textual reviews, social network structures and multimedia data for recommending relevant items in social network scenario. This goal is achieved from threefold: first, we gain an insight into the significance of the user-generated textual information in improving the effectiveness of social recommendation. Second, we study the dynamic nature of social network structure, which is utilized for online social recommendations. Finally, we analyze multimedia information which is then incorporated into a unified model for enhancing the effectiveness as well as the interpretability of social recommendations. Specifically, our main contributions of this work can be summarized as follows.

8.1.1 Exploitation of Textual Contexts for Social Recommendations

In Chapter 3, we explore the dynamic nature of data streams in social networks for sentiment analysis task. To better employ textual reviews with user interactions, we propose a dynamic topic-based sentiment analysis model, DTSA, which is capable of extracting topics and topic-specific sentiments from the online news comments and tracking their evolution over time simultaneously. The DTSA model incorporates the links among news comments to avoid the error caused by user interactions. To efficiently handle streaming data, we derive online inference procedures based on a stochastic Expectation-Maximization (EM) algorithm, in which the model is sequentially updated using newly arrived data and the parameters of the previously estimated model. To prove the validity of our model, we evaluate the proposed method on several real-world datasets.

In Chapter 4, we further investigate the problem of employing textual information for social recommendations on the basis of Chapter 3. Despite the effectiveness of topic model based approaches, we realize that topics extracted from these topic modelling methods are probabilistic distributions over independent words or phrases, and thus contextual information of words are neglected during the training process. Besides, short reviews make topic model related approaches more difficult to estimate the topic distributions [230]. Meanwhile, with the dramatic expansion of international markets, consumers write reviews in different languages, which poses a new challenge for recommender systems dealing with this increasing amount of multilingual information. To solve these problems, we utilize an unsupervised aspect-based autoencoder to learn a set of language-independent aspect embeddings. Then Multiple Instance Learning (MIL) framework integrated with hierarchical attention mechanism is designed to predict the aspect-specific sentiment distributions of review sentences, and learn aspect-aware sentence representations guided by the overall ratings. Note that the overall ratings serve both as a proxy of sentiment labels of reviews and as a bridge among languages. In addition, we further consider multilingualism in e-commerce and social media platforms, and develop a multilingual recommendation module to infer the overall rating through a prediction layer with its input of the aspect utilities estimated by a dual interactive attention mechanism, and the corresponding aspect importances of both the user and item considering the different contributions of multiple languages. To verify the effectiveness of our proposed framework, we apply our model to 9 real-world datasets from Amazon and Goodreads.

8.1.2 Exploitation of Network Structures for Social Recommendations

With the rapid proliferation of online social networks, personalized social recommendation has become an essential means to help people discover their potential friends or interested items in real-time. However, the cold-start issue and the special properties of social networks, such as rich temporal dynamics, heterogeneous and complex structures, render the most commonly used recommendation approaches (e.g. Collaborative Filtering) inefficient. Therefore in Chapter 5, we seek for a novel representation learning method that is capable of effective recommending both users and items simultaneously in real-time. Specifically, the proposed dynamic graph-based embedding (DGE) model can jointly capture the temporal semantic effects, social relationships and user behaviour sequential patterns in a unified way by embedding the constructed heterogeneous user-item (HUI) network into a shared low dimensional space. Then, with simple search methods or similarity calculations, we can use the encoded representation of temporal contexts to generate recommendations. Two real large-scale datasets are adopted to evaluate the performance of our model.

Chapter 6 goes deeper into the representation learning based social recommendation problem. On the basis of Chapter 5, we further consider the global community structure, together with the local context, i.e. the temporal semantic effects, social relationships and user behaviour sequential patterns in a unified way to address the issue of temporal dynamics, cold start and context awareness in an online social recommendation. Extensive experiments are designed to evaluate the effectiveness and efficiency of the proposed m-DNE model.

8.1.3 Exploitation of Multimedia Contexts for Social Recommendations

Explainable recommendation, which provides explanations about why an item is recommended, has attracted growing attention in both research and industry communities. However, most existing explainable recommendation methods cannot provide multi-model explanations consisting of both textual and visual modalities or adaptive explanations tailored for the user's dynamic preference, potentially leading to the degradation of customers' satisfaction, confidence and trust for the recommender system. Thus Chapter 7 aims to address the problem of explainability of social recommender systems by exploiting multimedia information apart from textual reviews. Specifically, we propose a novel Attentive Recurrent Neural Network (Ante-RNN) with textual and visual fusion for the dynamic explainable recommendation. Our model jointly learns image representations with textual alignment and text representations with topical attention mechanism in a parallel way. Then a novel dynamic contextual attention mechanism is incorporated into Ante-RNN for modelling the complicated correlations among recent items and strengthening the user's short-term interests. By combining the full latent visual-semantic alignments and a hybrid attention mechanism including topical and contextual attentions, Ante-RNN makes the recommendation process more transparent and explainable. To verify the performance of our Ante-RNN model, extensive experiments are conducted on two real large-scale datasets.

8.2 Answers to Research Questions

In the following, we reiterate the research questions raised in Chapter 1 and present the answers that we found in the course of the thesis.

RQ1: How can we extract topics and topic-specific sentiments from social media streams and analyse their evolution?

In Chapter 3, we present a probabilistic generative model incorporating the multiple timescale model to analyze topic and sentiment dynamics from social media streams (e.g. online news and tweets). Experimental results have shown that our model outperforms the baselines by 11.1% on average on the F1-score of sentiment classification task for that our method incorporates the co-effects caused by user interactions and the time factor. Meanwhile, our model can also exhibit the evolution of topics and topic-specific sentiments.

RQ2: To what extent can multilingual topic/aspect and sentiment information extracted from user reviews be used to improve the effectiveness, diversity and novelty of recommendation approaches?

In Chapter 4, we proposed a multilingual review-aware deep recommendation model which can not only extract aligned aspects and their associated sentiments in different languages, but also leverage the extracted information as multilingual contexts for overall rating prediction and item recommendation. The extensive experiments have shown an improvement in the recommendation effectiveness, diversity and novelty compared to state-of-the-art recommendation model. Specifically, our model outperforms the state-of-the-art baselines by 6.82% - 44.56%, 5.75% - 37.76% and 6.47% - 28.19% on average on the Mean Square Error (MSE), Intra-list Similarity (ILS) and Expected Popularity Complement (EPC) respectively. We believe the results benefit from the reasonable use of semantic information in multilingual textual reviews, and the consideration of both popular and long-tail items in modelling the fine-grained user-item interactions. Furthermore, our model can also interpret the recommendation results in great detail.

RQ3: How can the temporal contexts from large-scale heterogeneous networks be exploited to enhance social recommendation in real-time?

We constructed a heterogeneous user-item (HUI) network including the semantic contexts, social relationships as well as user-item interactions and proposed an updating mechanism as the social network evolves for recommendation tasks in Chapter 5. Experimental results on large-scale datasets have shown the effectiveness of dynamic graph embedding over HUI network in recommendation performance and addressing cold start issues. Specifically, our model can outperform the best baseline by 25.5% (Recall@10) and 19% (ARHR@10) in the item recommendation, as well as 11.6% (Recall@10) and 13.8% (ARHR@10) in the friend recommendation.

RQ4: Can community information induced from the network structure improve existing graph embedding models for the task of social recommendation?

We extended our DGE model proposed in Chapter 5 by incorporating the global context, which is community information derived from network structure, into the graph embedding model for social recommendation in Chapter 6. Our experiments clearly indicated that it could outperform the cases without adopting community information by 12.7% (Recall@10) and 28.4% (ARHR@10) in the item recommendation, as well as 12.3% (Recall@10) and 14.5% (ARHR@10) in the friend recommendation.

RQ5: Can social recommendation benefit from incorporating visual context in terms of performance and interpretability?

We provide a thorough investigation of dynamic user preference modelling and multi-modality fusion strategies for explainable social recommendation in Chapter 7. The extensive experiments verified that our model with textual and visual fusion could provide not only competitive recommendation performance among which the Recall@10 and NDCG@10 of our Ante-RNN model surpass the best baselines by 20.8% and 32.1% on average, but also reasonable visual aligned with textual explanations for the recommended items.

8.3 Reflection and Limitations

The work presented in this thesis could be of interest and inspiration to the academic and industry researchers when it comes to analyzing latent knowledge in the complicated online social environment, and modelling dynamic user behaviour patterns for social recommendations. Several methods have been proposed and evaluated at real-world data sources in various social domains. Despite their effectiveness in our initial explorations, there are still some limitations listed below:

1) Since our MrRec recommender system does not consider geolocation information when

providing recommendations to users, the generated recommendation results incorporate items with reviews in all languages from world-wide websites, while cannot be effectively differentiated according to the real-time location of the target user. Besides, a few error cases in our experiments show that the sentiment attention weights distribute evenly on nearly all words because the sentence does not contain any explicit sentiment words or expresses special sentiments such as sarcasm.

2) The random walk generator in our (m-)DGE model sets the same number of random walks starting from each vertex and constrains the length of walks to be the same, which limits the capability of the generator and makes it difficult to generate a “corpus” of vertices following the real-world social network with a power-law distribution [318].

3) Due to the gating mechanism of recurrent neural networks, our Ante-RNN model cannot provide users with a direct and meaningful way to correct the recommendation process if they are unsatisfied with the results. This limits the scrutability of recommender systems and degrades customers' satisfaction with recommendation services.

4) Our proposed social recommender systems take advantage of rich side-information like user-features or item-features to improve the effectiveness of recommendation. Such information may include users’ privacy (such as shopping records and rating records), and there is a risk of sensitive information leakage. For example, the attackers can infer whether an individual rating is included in the training set (known as inference attack) or predict the exact value of some sensitive features about a target user based on some background information (known as reconstruction attack) [319].

5) In social networks, there exist many fake users (also called “shills” or “water army”) with the intention of spreading particular contents [320, 321]. Ratings or relationships injected by fake users seriously affect the authenticity of the recommendations as well as users’ trustiness on our recommender systems.

6) Social networking sites allow users to upload information or resources in the form of documents, images, videos, audios, and check-ins. Some resources can also be assigned to social tags. All the aforementioned information can be of great value in modelling users’ preferences or characterizing items. Due to the lack of time or relavant datasets, regardless of limited sources referred in this thesis, we are unable to fuse them in a unified model, or explore their influence on recommendation performance in distinct application scenarios.

8.4 Future Work

In this thesis, we present the following promising research directions for future work:

Our future work on *exploring the textual context in social recommendation* includes combining explicit contexts, such as spatial-temporal information into social recommendation. In reality, user preferences often evolve over time, and they are influenced by variable users' inclinations, users' social relations, item popularities etc. Temporal information thus plays a crucial role in modelling dynamic user preferences. Many existing researches have verified the effectiveness and importance of time factor for recommendation domain. Also considering the first limitation discussed in Section 8.3, spatial-temporal information is therefore an important type of information for recommendations. Besides, our future work will focus on detecting special forms of sentiment expressions in sentences and thereby integrating it into recommendation tasks as well.

Our future research direction on *analyzing network structure for social recommendation* will integrate attributes from multiple social sites. In recent years, users tend to participate in multiple social networks such as Twitter, Facebook, LinkedIn, and so on. These social networks are inter-connected through the deployment of cross-linking functionality [322], which offers information about the same users from different perspectives. The data from these cross sources can be multi-modal, which provide distinct information for recommendation. Despite its promising opportunity, there exist many challenges of integrating cross-source information into recommendation tasks. To begin with, although users with accounts on multiple social media sites give potentials to fully understand users' interests, how to differentiate identical users across multiple online social networks with big, noisy, incomplete and highly-unstructured properties, is a non-trivial task. Besides, data integration can be another issue for cross-source recommendation since multiple data sources often describe distinct sides of users' characteristics with inconsistent forms.

Our planned future works on *analyzing visual context for social recommendation* is multifold. First, based on the third limitation in section 8.3, developing recommendation approaches that are more scrutable would be an interesting research topic and needs to be addressed in future works. Second, current deep learning based approaches focus on generating explainable recommendation results from attention weights over texts, images or videos, whereas the researches on explainable deep learning for recommendation is still in its initial stage, which requires more focus in the long run. In addition, knowledge graph is a promising external source for providing precise explainable recommendations, especially for some specific domains such as movie, music and news. Some studies have delved into incorporating knowledge graph into recommender systems with representation learning techniques, but how to perform explainable recommendations with the help of knowledge graph still needs further research. Besides, considering many heterogeneous multi-modal information sources existing in an online social environment, how to leverage such kind of information for better recommendation while performing explanation to users also needs to be considered in the future.

Numerous studies have exploited different latent contexts extracted from online social networks to improve the quality of recommendations, especially in cold-start and data sparsity scenario. In this thesis, we basically deal with each kind of latent contexts in separate. Although the heterogeneous information is considered in Chapter 5 and the fusion of both textual and visual information is investigated in Chapter 7, the social recommender system, in reality, may encounter more complex scenarios (such as a recommender system with user reviews, social relationships, and multimedia information) or provide various explanations about recommended items from multiple perspectives (such as to integrate aspect opinions, community and image regions etc. together). Furthermore, we observe from the experiments that not all latent contexts are of equal importance to arbitrary datasets. For instance, visual features play a more crucial role in explaining the recommendation results in Pinterest than MovieLens dataset for that the posters of the latter one are more impressionistic rather than expressing specific meanings. Unreasonable integration of all latent contexts may generate noise and degrade the effectiveness and efficiency of recommender systems. To this end, a comprehensive model which can integrate and balance all information sources and preserve user privacy is a promising direction.

Bibliography

[1] Kouki, Pigi, Shobeir Fakhraei, James Foulds, Magdalini Eirinaki, and Lise Getoor. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 99-106. ACM, 2015.

[2] Shi, Chuan, Zhiqiang Zhang, Ping Luo, Philip S. Yu, Yading Yue, and Bin Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 453-462. ACM, 2015.

[3] Zhengshen Jiang, Hongzhi Liu, Bin Fu, Zhonghai Wu, and Tao Zhang. Recommendation in Heterogeneous Information Networks Based on Generalized Random Walk Model and Bayesian Personalized Ranking. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pp. 288-296. ACM, 2018.

[4] Liu, Xin, and Karl Aberer. SoCo: a social network aided context-aware recommender system. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 781-802. ACM, 2013.

[5] Chen, Li, Guanliang Chen, and Feng Wang. Recommender systems based on user reviews: the state of the art. User Modeling and User-Adapted Interaction, 25(2), pp.99-154.

[6] Hernández-Rubio, María, Iván Cantador, and Alejandro Bellogín. A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction*, 29.2 (2019): 381-441.

[7] Yin, Bin, Yujiu Yang, and Wenhuang Liu. Exploring social activeness and dynamic interest in community-based recommender system. In *Proceedings of the 23rd International Conference on World Wide Web*, pp. 771-776. ACM, 2014.

[8] Li, Hui, Dingming Wu, Wenbin Tang, and Nikos Mamoulis. Overlapping community regularization for rating prediction in social recommender systems. In *Proceedings of the*

9th ACM Conference on Recommender Systems, pp. 27-34. ACM, 2015.

[9] Cui, Qiang, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[10] Cheng, Zhiyong, Xiaojun Chang, Lei Zhu, Rose C. Kanjirathinkal, and Mohan Kankanhalli. MMALFM: Explainable recommendation by leveraging reviews and images. ACM Transactions on Information Systems (TOIS), 37.2 (2019): 16.

[11] Peffers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. Journal of management information systems, 24(3), pp.45-77, 2007.

[12] Hevner, Alan R., Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. MIS quarterly, pp.75-105, 2004.

[13] Iwata, Tomoharu, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multiscale dynamic topic models. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 663-672. ACM, 2010.

[14] Aggarwal, Charu C. Recommender Systems: The Textbook. Cham: Springer International Publishing, 2016.

[15] Lin, Chenghua, and Yulan He. Joint sentiment/topic model for sentiment analysis. In Proceedings of the 18th ACM conference on Information and knowledge management, pp. 375-384. ACM, 2009.

[16] Linden, Greg, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing 1 (2003): 76-80.

[17] Koren, Yehuda, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer 8 (2009): 30-37.

[18] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. Social Network Analysis and Mining, 3(4):1113–1133, 2013.

[19] Leily Sheugh and Sasan H Alizadeh. A note on pearson correlation coefficient as a metric of similarity in recommender system. In AI & Robotics (IRANOPEN), 2015, pages 1–6. IEEE, 2015. 15

[20] Koutrika, Georgia, Benjamin Bercovitz, and Hector Garcia-Molina. FlexRecs: expressing and combining flexible recommendations. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp. 745-758. ACM, 2009.

[21] Jannach, Dietmar, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender systems: an introduction. Cambridge University Press, 2010. Chapter 2, 3

[22] Ke Ji, Hong Shen, Hui Tian, Yanbo Wu, and Jun Wu. Two-phase layered learning recommendation via category structure. In PAKDD, pages 13{24. Springer, 2014.

[23] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative fltering model. In KDD, pages 426-434. ACM, 2008.

[24] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In NIPS, pages 1257-1264, 2008.

[25] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. 1999. 16, 42, 55

[26] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. Factorization meets the item embedding: Regularizing matrix factorization with item cooccurrence. In RecSys, pages 59-66. ACM, 2016.

[27] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, pp. 225-232. ACM, 2016.

[28] Wei-Yin Loh. Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1):14–23, 2011. 16.

[29] Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In CIT, volume 1, 2002.

[30] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative fltering using cluster-based smoothing. In SIGIR, pages 114-121. ACM, 2005.

[31] Wang, Hao, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1235-1244. ACM, 2015.

[32] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In Proceedings of the 26th International Conference on Wold Wide Web, pages 173–182. International World Wide Web Conferences Steering Committee, 2017. vii, 16, 27, 28, 29, 32, 108, 125, 161.

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

[34] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In Proceedings of ACM SIGIR workshop on recommender systems, volume 60. Citeseer, 1999.

[35] Pazzani, Michael J. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, pages 393-408, December 1999.

[36] Melville, Prem, Raymod J. Mooney, and Ramadass Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, Canada, 2002.

[37] Soboroff, Ian, and Charles Nicholas. Combining content and collaboration in text filtering. In IJCAI'99 Workshop: Machine Learning for Information Filtering, August 1999.

[38] Schein, Andrew I., Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 253-260. 2002.

[39] Duen-Ren Liu, Pei-Yun Tsai, and Po-Huan Chiu. Personalized recommendation of popular blog articles for mobile applications. Information Sciences, 181(9):1552–1572, 2011.

[40] He, Xiangnan, and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 355-364. ACM, 2017.

[41] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of

social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710, ACM, 2014.

[42] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Proceedings of ICIR, 2013.

[43] Tang, Jian, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web, pp. 1067-1077. International World Wide Web Conferences Steering Committee, 2015.

[44] Grover, Aditya, and Jure Leskovec. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855-864. 2016.

[45] Dong, Yuxiao, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 135-144. 2017.

[46] Tang, Jian, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1165-1174. 2015.

[47] Cavallari, Sandro, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 377-386. 2017.

[48] Li, Jundong, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 387-396. 2017.

[49] Wang, Xiao, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community Preserving Network Embedding. In Thirty-first AAAI conference on artificial intelligence, pp. 203-209 (2017).

[50] Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering. In Proceedings of the Poster Track of RecSys, 2016b. 27.

[51] Grbovic, Mihajlo, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-commerce in your inbox: Product recommendations at scale. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1809-1818. 2015.

[52] Sun, Zhu, Jie Yang, Jie Zhang, Alessandro Bozzon, Yu Chen, and Chi Xu. MRLR: Multi-level Representation Learning for Personalized Ranking in Recommendation. In IJCAI, pp. 2807-2813. 2017.

[53] Yu, Feng, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 729-732. ACM, 2016.

[54] Song, Yang, Ali Mamdouh Elkahky, and Xiaodong He. Multi-rate deep learning for temporal recommendation. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 909-912. ACM, 2016.

[55] Hidasi, Balázs, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015).

[56] Hidasi, Balázs, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In Proceedings of the 10th ACM conference on recommender systems, pp. 241-248. ACM, 2016.

[57] Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. End-to-end memory networks. In Advances in neural information processing systems, pp. 2440-2448. 2015.

[58] Chen, Xu, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In Proceedings of the eleventh ACM international conference on web search and data mining, pp. 108-116. ACM, 2018.

[59] Huang, Jin, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In The 41st International ACM SIGIR Conference on Research & Development in

Information Retrieval, pp. 505-514. ACM, 2018.

[60] Chen, Minmin, Kilian Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising auto-encoders for nonlinear representations. In International Conference on Machine Learning, pp. 1476-1484. 2014.

[61] Wu, Yao, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 153-162. ACM, 2016.

[62] Zhang, Shuai, Lina Yao, and Xiwei Xu. Autosvd++: An efficient hybrid collaborative filtering model via contractive auto-encoders. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 957-960. ACM, 2017.

[63] Van den Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In Advances in neural information processing systems, pp. 2643-2651. 2013.

[64] Zhu, Yu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. Addressing the item cold-start problem by attribute-driven active learning. IEEE Transactions on Knowledge and Data Engineering (2019).

[65] Zhang, Xi, Jian Cheng, Shuang Qiu, Guibo Zhu, and Hanqing Lu. Dualds: A dual discriminative rating elicitation framework for cold start recommendation. Knowledge-Based Systems 73 (2015): 161-172.

[66] Houlsby, Neil, José Miguel Hernández-Lobato, and Zoubin Ghahramani. Cold-start active learning with robust ordinal matrix factorization. In: Proceedings of the 31st international conference on machine learning, pp 766–774, 2014.

[67] Sedhain, Suvash, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. Social collaborative filtering for cold-start recommendations. In: Proceedings of the 8th ACM conference on recommender systems. ACM, pp 345–348, 2014.

[68] Qian, Xueming, He Feng, Guoshuai Zhao, and Tao Mei. Personalized recommendation combining user interest and social circle. IEEE transactions on knowledge and data engineering, 26(7):1763–1777, 2014.

[69] Anderson, Chris. The Long Tail. Hyperion press, 2006.

[70] Papagelis, Manos, Dimitris Plexousakis, and Themistoklis Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In International Conference on Trust Management, pp. 224-239. Springer, Berlin, Heidelberg, 2005.

[71] Adamopoulos, Panagiotis, and Alexander Tuzhilin. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In Proceedings of the 8th ACM Conference on Recommender systems, pp. 153-160. 2014.

[72] Zhang, Mi, and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In Proceedings of the 2008 ACM conference on Recommender systems, pp. 123-130. 2008.

[73] Hsu, Shang H., Ming-Hui Wen, Hsin-Chieh Lin, Chun-Chia Lee, and Chia-Hoang Lee. AIMED-A personalized TV recommendation system. In European conference on interactive television, pp. 166-174. Springer, Berlin, Heidelberg, 2007.

[74] Das, Abhinandan S., Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web, pp. 271-280. 2007.

[75] Li, Lei, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. SCENE: a scalable two-stage personalized news recommendation system. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pp. 125-134. 2011.

[76] Medo, Matúš, Yi-Cheng Zhang, and Tao Zhou. Adaptive model for recommendation of news, EPL (Europhysics Letters) 88 (3).

[77] Vargas, Saúl, and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In Proceedings of the fifth ACM conference on Recommender systems, pp. 109-116. ACM, 2011.

[78] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the conference on Empirical methods in natural language processing-Volume 10, pp. 79-86. Association for Computational Linguistics, 2002.

[79] Tang, Duyu, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 1014-1023. 2015.

[80] Nakagawa, Tetsuji, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using CRFs with hidden variables. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 786-794. Association for Computational Linguistics, 2010.

[81] Habernal, Ivan, Tomáš Ptáček, and Josef Steinberger. Supervised sentiment analysis in Czech social media. Information Processing & Management 50.5 (2014): 693-707.

[82] Singh, Pravesh Kumar, and Mohd Shahid Husain. Methodological study of opinion mining and sentiment analysis techniques. International Journal on Soft Computing 5.1 (2014): 11.

[83] Zhongwu Zhai, Hua Xu, Bada Kang, and Peifa Jia. Exploiting effective features for chinese sentiment classification. Expert Systems with Applications 38.8 (2011): 9139-9146.

[84] Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. Computational linguistics 37, no. 2 (2011): 267-307.

[85] Thelwall, Mike, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. Journal of the American Society for Information Science and Technology 63, no. 1 (2012): 163-173.

[86] Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1555-1565. 2014.

[87] Tang, Duyu, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1422-1432. 2015.

[88] Hu, Minqing, and Bing Liu. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

[89] Popescu, Ana-Maria, and Orena Etzioni. Extracting product features and opinions

from reviews. In Natural language processing and text mining. Springer, London, pp. 9-28, 2007.

[90] Moghaddam, Samaneh, and Martin Ester. Opinion digger: an unsupervised opinion miner from unstructured product reviews. In Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010.

[91] Jin, Wei, Hung Hay Ho, and Rohini K. Srihari. A novel lexicalized HMM-based learning framework for web opinion mining. In Proceedings of the 26th annual international conference on machine learning. Citeseer, 2009.

[92] Jakob, Niklas, and Iryna Gurevych. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2010. p. 1035-1045.

[93] Choi, Yejin, and Claire Cardie. Hierarchical sequential learning for extracting opinions and their attributes. In Proceedings of the ACL 2010 conference short papers. Association for Computational Linguistics, 2010. p. 269-274.

[94] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022, 2003.

[95] Griffiths, Thomas L., and Mark Steyvers. Finding scientific topics. Proceedings of the National academy of Sciences, 101(suppl 1), 5228-5235, 2004.

[96] Jo, Yohan, and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.

[97] Li, Chengtao, Jianwen Zhang, Jian-Tao Sun, and Zheng Chen. Sentiment topic model with decomposed prior. In Proceedings of the 2013 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, pp. 767-775, 2013.

[98] Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, pp. 142-150, 2011.

[99] Xiang, Bing, and Liang Zhou. Improving twitter sentiment analysis with topic-based

mixture modeling and semi-supervised training. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 434-439, 2014.

[100] Ren, Yafeng, Yue Zhang, Meishan Zhang, and Donghong Ji. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In Thirtieth AAAI conference on artificial intelligence, 2016.

[101] Wang, Chong, David Blei, and David Heckerman. Continuous Time Dynamic Topic Models. In Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence, pp. 579–586. AUAI Press, 2008.

[102] Wang, Yu, Eugene Agichtein, and Michele Benzi. TM-LDA: efficient online modeling of latent topic transitions in social media. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 123-131, 2012.

[103] Dermouche, M., Velcin, J., Khouas, L., & Loudcher, S. A joint model for topic-sentiment evolution over time. In 2014 IEEE International Conference on Data Mining. IEEE, pp. 773-778, 2014.

[104] He, Yulan, Chenghua Lin, Wei Gao, and Kam-Fai Wong. Tracking sentiment and topic dynamics from social media. In Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM), pp. 483-486. Association for the Advancement of Artificial Intelligence, 2012.

[105] He, Yulan, Chenghua Lin, Wei Gao, and Kam-Fai Wong. Dynamic joint sentiment-topic model. ACM Transactions on Intelligent Systems and Technology (TIST), 5(1), pp.1-21.

[106] Zheng, M., Wu, C., Liu, Y., Liao, X., & Chen, G. Topic sentiment trend model: modeling facets and sentiment dynamics. In: 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE). IEEE, pp. 651-657, 2012.

[107] Wan, Xiaojun. Bilingual co-training for sentiment classification of Chinese product reviews. Computational Linguistics 37, no. 3 (2011): 587-616.

[108] Kim, Soo-Min, and Eduard Hovy. Automatic identification of pro and con reasons in online reviews. In Proceedings of the COLING/ACL on Main conference poster sessions.

Association for Computational Linguistics, pp. 483-490, 2006.

[109] Banea, Carmen, Rada Mihalcea, and Janyce Wiebe. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In Proceedings of the Conference on Language Resources and Evaluations, pp. 2764-2767, 2008

[110] Pan, Junfeng, Gui-Rong Xue, Yong Yu, and Yang Wang. Cross-lingual sentiment classification via bi-view non-negative matrix tri-factorization. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 289-300. Springer, Berlin, Heidelberg, 2011.

[111] Lu, Bin, Chenhao Tan, Claire Cardie, and Benjamin K. Tsou. Joint bilingual sentiment classification with unlabeled parallel corpora. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 320-330. Association for Computational Linguistics, 2011.

[112] Meng, Xinfan, Furu Wei, Xiaohua Liu, Ming Zhou, Ge Xu, and Houfeng Wang. Cross-lingual mixture model for sentiment classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 572-581. Association for Computational Linguistics, 2012.

[113] Zhou, Guangyou, Zhiyuan Zhu, Tingting He, and Xiaohua Tony Hu. Cross-lingual sentiment classification with stacked autoencoders. Knowledge and Information Systems, 47(1), 27-44, 2016.

[114] Rasooli, Mohammad Sadegh, Noura Farra, Axinia Radeva, Tao Yu, and Kathleen McKeown. Cross-lingual sentiment transfer with limited resources. Machine Translation, 32(1-2), 143-165, 2018.

[115] Popat, Kashyap, A. R. Balamurali, Pushpak Bhattacharyya, and Gholamreza Haffari. The haves and the have-nots: Leveraging unlabelled corpora for sentiment analysis. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 412-422, 2013.

[116] Lambert, Patrik. Aspect-level cross-lingual sentiment classification with constrained SMT. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 781-787, 2015.

[117] Almeida, Mariana SC, Cláudia Pinto, Helena Figueira, Pedro Mendes, and André FT Martins. Aligning opinions: Cross-lingual opinion mining with dependencies. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 408-418, 2015.

[118] Klinger, Roman, and Philipp Cimiano. Instance selection improves cross-lingual model training for fine-grained sentiment analysis. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, pp. 153-163, 2015.

[119] Zhang, Duo, Qiaozhu Mei, and ChengXiang Zhai. Cross-lingual latent topic extraction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 1128-1137, 2010.

[120] Boyd-Graber, Jordan, and David M. Blei. Multilingual topic models for unaligned text. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, pp. 75-82, 2009.

[121] Guo, Honglei, Huijia Zhu, Zhili Guo, Xiaoxun Zhang, and Zhong Su. OpinionIt: a text mining system for cross-lingual opinion analysis. In Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, pp. 1199-1208, 2010.

[122] Boyd-Graber, Jordan, and Philip Resnik. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 45-55, 2010.

[123] Lin, Zheng, Xiaolong Jin, Xueke Xu, Weiping Wang, Xueqi Cheng, and Yuanzhuo Wang. A cross-lingual joint aspect/sentiment model for sentiment analysis. In Proceedings of the 23rd ACM international conference on conference on information and knowledge management. ACM, pp. 1089-1098, 2014.

[124] Lin, Zheng, Xiaolong Jin, Xueke Xu, Yuanzhuo Wang, Xueqi Cheng, Weiping Wang, and Dan Meng. An unsupervised cross-lingual topic model framework for sentiment classification. IEEE/ACM transactions on audio, speech, and language processing, 24.3: 432-444, 2015.

[125] Jakob, Niklas, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion. ACM, pp. 57-64, 2009.

[126] Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. 2005.

[127] Lippert, Christoph, Stefan Hagen Weber, Yi Huang, Volker Tresp, Matthias Schubert, and Hans-Peter Kriegel. Relation prediction in multi-relational domains using matrix factorization. In Proceedings of the NIPS 2008 Workshop: Structured Input-Structured Output, Vancouver, Canada. 2008.

[128] Wang, Yuanhong, Yang Liu, and Xiaohui Yu. Collaborative filtering with aspect-based opinion mining: A tensor factorization approach. In IEEE 12th International Conference on Data Mining. IEEE, pp. 1152-1157, 2012.

[129] Qiu, Guang, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. Computational linguistics, 37.1: 9-27, 2011.

[130] Wang, Hongning, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 783-792, 2010.

[131] Ganu, Gayatree, Yogesh Kakodkar, and AméLie Marian. Improving the quality of predictions using textual information in online user reviews. Information Systems, 38.1: 1-15, 2013.

[132] McAuley, Julian, and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems. ACM, pp. 165-172, 2013.

[133] Diao, Qiming, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 193-202, 2014.

[134] Wu, Yao, and Martin Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. ACM, pp. 199-208, 2015.

[135] Chen, Xu, Zheng Qin, Yongfeng Zhang, and Tao Xu. Learning to rank features for recommendation over multiple categories. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, pp. 305-314, 2016.

[136] Liu, Hongyan, Jun He, Tingting Wang, Wenting Song, and Xiaoyang Du. Combining user preferences and user opinions for accurate recommendation. Electronic Commerce Research and Applications, 12(1): 14-23, 2013.

[137] Chen, Li, and Feng Wang. Preference-based clustering reviews for augmenting e-commerce recommendation. Knowledge-Based Systems, 50: 44-59, 2013.

[138] MILLER, George A. WordNet: a lexical database for English. Communications of the ACM, 38.11: 39-41, 1995.

[139] Esuli, Andrea, and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In: LREC. 2006. p. 417-422.

[140] Aciar, Silvana, Debbie Zhang, Simeon Simoff, and John Debenham. Informed recommender: Basing recommendations on consumer product reviews. IEEE Intelligent systems, 22.3: 39-47, 2007.

[141] Yates, Alexander, James Joseph, Ana-Maria Popescu, Alexander D. Cohn, and Nick Sillick. Shopsmart: product recommendations through technical specifications and user reviews. In Proceedings of the 17th ACM conference on Information and knowledge management. ACM, pp. 1501-1502, 2008.

[142] Dong, Ruihai, Markus Schaal, Michael P. O'Mahony, Kevin McCarthy, and Barry Smyth. Opinionated product recommendation. In International conference on case-based reasoning. Springer, Berlin, Heidelberg, pp. 44-58, 2013.

[143] Bauman, Konstantin, Bing Liu, and Alexander Tuzhilin. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 717-725, 2017.

[144] Musto, Cataldo, Marco de Gemmis, Giovanni Semeraro, and Pasquale Lops. A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews. In Proceedings of the eleventh ACM conference on recommender systems. ACM, pp. 321-325, 2017.

[145] Caputo, Annalina, Pierpaolo Basile, Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, and Gaetano Rossiello. SABRE: a sentiment aspect-based retrieval engine. In Information Filtering and Retrieval. Springer, Cham, pp. 63-78, 2017.

[146] Li, Chenliang, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. A Capsule Network for Recommendation and Explaining What You Like and Dislike. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 275-284, 2019.

[147] NEWMAN, Mark EJ. Detecting community structure in networks. The European Physical Journal B, 38.2: 321-330, 2004.

[148] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence, 22(8): 888-905, 2000.

[149] Flake, Gary William, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 150-160, 2000.

[150] Newman, Mark EJ, and Michelle Girvan. Finding and evaluating community structure in networks. Physical review E, 69(2): 026-113, 2004.

[151] Andersen, Reid, and Kevin J. Lang. Communities from seed sets. In Proceedings of the 15th international conference on World Wide Web. ACM, pp. 223-232, 2006.

[152] Ruan, Jianhua, and Weixiong Zhang. An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In Seventh IEEE International Conference on Data Mining. IEEE, pp. 643-648, 2007.

[153] Girvan, Michelle, and Mark EJ Newman. Community structure in social and biological networks. In Proceedings of the national academy of sciences, 99.12: 7821-7826, 2002.

[154] McCallum, Andrew, Xuerui Wang, and Natasha Mohanty. Joint group and topic discovery from relations and text. In ICML Workshop on Statistical Network Analysis.

Springer, Berlin, Heidelberg, pp. 28-44, 2006.

[155] Palla, Gergely, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435 (2005): 814-818.

[156] Shi, Chuan, Yanan Cai, Di Fu, Yuxiao Dong, and Bin Wu. A link clustering based overlapping community detection algorithm. Data & Knowledge Engineering, 87: 394-404, 2013.

[157] Yahia, Sihem Amer, Michael Benedikt, and Philip Bohannon. Challenges in searching online communities. IEEE Data Eng. Bull. 2007.

[158] Silva, Arlei, Wagner Meira Jr, and Mohammed J. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. In Proceedings of the VLDB Endowment, 5(5): 466-477, 2012.

[159] Liu, Yan, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link LDA: joint models of topic and author community. In Proceedings of the 26th annual international conference on machine learning. ACM, pp. 665-672, 2009.

[160] Yang, Tianbao, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: a discriminative approach. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 927-936, 2009.

[161] Qi, Guo-Jun, Charu C. Aggarwal, and Thomas Huang. Community detection with edge content in social media networks. In IEEE 28th International Conference on Data Engineering. IEEE, pp. 534-545, 2012.

[162] Zhou, Yang, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. In Proceedings of the VLDB Endowment, 2(1): 718-729, 2009.

[163] Sachan, Mrinmaya, Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. Using content and interactions for discovering communities in social networks. In Proceedings of the 21st international conference on World Wide Web. ACM, pp. 331-340, 2012.

[164] Heli Sun, Jianbin Huang, Xin Zhang, Jiao Liu, DongWang, Huailiang Liu, Jianhua

Zou, and Qinbao Song. IncOrder: Incremental density-based community detection in dynamic networks. Knowledge-Based Systems, 72: 1-12, 2014.

[165] Wenjun Wang, Pengfei Jiao, Dongxiao He, Di Jin, and Lin Pan. Autonomous overlapping community detection in temporal networks: A dynamic Bayesian nonnegative matrix factorization approach. Knowledge-Based Systems, 110: 121-134, 2016.

[166] Sun, Jimeng, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 687-696, 2007.

[167] Lin, Yu-Ru, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In Proceedings of the 17th international conference on World Wide Web. ACM, pp. 685-694, 2008.

[168] Chakrabarti, Deepayan, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 554-560, 2006.

[169] Zhu, Shenghuo, Kai Yu, Yun Chi, and Yihong Gong. Combining content and link for classification using matrix factorization. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 487-494, 2007.

[170] Ma, Hao, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM conference on Information and knowledge management. ACM, pp. 931-940, 2008.

[171] Ma, Hao, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In Proceedings of the fourth ACM international conference on Web search and data mining. ACM, pp. 287-296, 2011.

[172] Jamali, Mohsen, and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the fourth ACM conference on Recommender systems. ACM, pp. 135-142, 2010.

[173] Shen, Yelong, and Ruoming Jin. Learning personal+ social latent factor model for

social recommendation. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1303-1311, 2012.

[174] Sun, Yizhou, and Jiawei Han. Meta-path-based search and mining in heterogeneous information networks. Tsinghua Science and Technology, 18.4: 329-338, 2013.

[175] Vahedian, Fatemeh, Robin Burke, and Bamshad Mobasher. Weighted random walk sampling for multi-relational recommendation. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization. ACM, pp. 230-237, 2017.

[176] Shi, C., Hu, B., Zhao, W. X., & Philip, S. Y. Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering, 31.2: 357-370, 2018.

[177] Ying, Jia-Ching, Bo-Nian Shi, Vincent S. Tseng, Huan-Wen Tsai, Kuang Hung Cheng, and Shun-Chieh Lin. Preference-aware community detection for item recommendation. In Conference on Technologies and Applications of Artificial Intelligence. IEEE, pp. 49-54, 2013.

[178] Zhao, Gang, Mong Li Lee, Wynne Hsu, Wei Chen, and Haoji Hu. Community-based user recommendation in uni-directional social networks. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, pp. 189-198, 2013.

[179] Bellogin, Alejandro, and Javier Parapar. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. In Proceedings of the sixth ACM conference on Recommender systems. ACM, pp. 213-216, 2012.

[180] Cao, Cen, Qingjian Ni, and Yuqing Zhai. An improved collaborative filtering recommendation algorithm based on community detection in social networks. In: Proceedings of the 2015 annual conference on genetic and evolutionary computation. ACM, pp. 1-8, 2015.

[181] Shahriari, Mohsen, Martin Barth, Ralf Klamma, and Christoph Trattner. TCNSVD: A Temporal and Community-Aware Recommender Approach. In RecTemp@ RecSys, pp. 21-27, 2017.

[182] Shahriari, Mohsen, Sebastian Krott, and Ralf Klamma. Disassortative degree mixing and information diffusion for overlapping community detection in social networks (dmid).

In Proceedings of the 24th International Conference on World Wide Web. ACM, pp. 1369-1374, 2015.

[183] Pons, Pascal, and Matthieu Latapy. Computing communities in large networks using random walks. In International symposium on computer and information sciences. Springer, Berlin, Heidelberg, pp. 284-293, 2005.

[184] Simonyan, Karen, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[185] Sharif Razavian, Ali, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 806-813, 2014.

[186] Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6077-6086, 2018.

[187] Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning, pp. 647-655, 2014.

[188] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[189] Ioffe, Sergey. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In Advances in neural information processing systems, pp.1945-1953, 2017.

[190] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[191] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[192] He, Ruining, and Julian McAuley. VBPR: visual bayesian personalized ranking from implicit feedback. In Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[193] Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, pp. 675-678. 2014.

[194] Liu, Qiang, Shu Wu, and Liang Wang. DeepStyle: Learning user preferences for visual recommendation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 841-844, 2017.

[195] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9, 2015.

[196] Zhang, Qi, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In IJCAI, pp. 3420-3426, 2017.

[197] Zhang, Suwei, Yuan Yao, Feng Xu, Hanghang Tong, Xiaohui Yan, and Jian Lu. Hashtag Recommendation for Photo Sharing Services. In the Thirty-Third AAAI Conference on Artificial Intelligence, pp. 5805-5812, 2019.

[198] Chen, Jingyuan, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, pp. 335-344, 2017.

[199] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pp. 91-99, 2015.

[200] McAuley, Julian, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 43-52, 2015.

[201] Rendle, Steffen, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, pp. 452-461, 2009.

[202] Geng, Xue, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. Learning image and user features for recommendation in social networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 4274-4282, 2015.

[203] Wang, Suhang, Yilin Wang, Jiliang Tang, Kai Shu, Suhas Ranganath, and Huan Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 391-400, 2017.

[204] Zhang, Yongfeng, Qingyao Ai, Xu Chen, and W. Bruce Croft. Joint representation learning for top-n recommendation with heterogeneous information sources. In Proceedings of the ACM on Conference on Information and Knowledge Management. ACM, pp. 1449-1458, 2017.

[205] Chandramouli, Badrish, Justin J. Levandoski, Ahmed Eldawy, and Mohamed F. Mokbel. StreamRec: a real-time recommender system. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, pp. 1243-1246. ACM, 2011.

[206] Stern, David H., Ralf Herbrich, and Thore Graepel. Matchbox: large scale online bayesian recommendations. In Proceedings of the 18th international conference on World wide web, pp. 111-120. ACM, 2009.

[207] Christakopoulou, Konstantina, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 815-824. ACM, 2016.

[208] Agarwal, Deepak, Bee-Chung Chen, and Pradheep Elango. Fast online learning through offline initialization for time-sensitive recommendation. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 703-712, 2010.

[209] Diaz-Aviles, Ernesto, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. Real-time top-n recommendation in social streams. In Proceedings of the sixth ACM conference on Recommender systems. ACM, pp. 59-66, 2012.

[210] Chen, Chen, Hongzhi Yin, Junjie Yao, and Bin Cui. Terec: A temporal recommender system over tweet stream. In Proceedings of the VLDB Endowment, 6.12: 1254-1257, 2013.

[211] Huang, Yanxiang, Bin Cui, Wenyu Zhang, Jie Jiang, and Ying Xu. Tencentrec: Real-time stream recommendation in practice. In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM, pp. 227-238, 2015.

[212] Subbian, Karthik, Charu Aggarwal, and Kshiteesh Hegde. Recommendations for streaming data. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, pp. 2185-2190, 2016.

[213] Balahur, Alexandra, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. arXiv preprint arXiv:1309.6202, 2013.

[214] Lu Wang and Claire Cardie. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 97–106, Association for Computational Linguistics, 2014.

[215] Lin, Chenghua, Yulan He, Richard Everson, and Stefan Ruger. Weakly supervised joint sentiment-topic detection from text. IEEE Transactions on Knowledge and Data engineering 24, no. 6 (2011): 1134-1145.

[216] Dean, Jeffrey, and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. Communications of the ACM 51, no. 1 (2008): 107-113.

[217] Blei, David M. Probabilistic topic models. Communications of the ACM 55, no. 4 (2012): 77-84.

[218] Wang, Xuerui, and Andrew McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 424-433, ACM, 2006.

[219] Cheng, Zhiyong, Ying Ding, Lei Zhu, and Mohan Kankanhalli. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In Proceedings of the 2018 world wide web conference, pp. 639-648. ACM, 2018.

[220] Guan, Xinyu, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke

Peng, and Tat-Seng Chua. Attentive aspect modeling for review-aware recommendation. ACM Transactions on Information Systems (TOIS) 37, no. 3 (2019): 1-27.

[221] Chin, Jin Yao, Kaiqi Zhao, Shafiq Joty, and Gao Cong. ANR: Aspect-based neural recommender. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 147-156. ACM, 2018.

[222] Narducci, Fedelucio, Pierpaolo Basile, Cataldo Musto, Pasquale Lops, Annalina Caputo, Marco de Gemmis, Leo Iaquinta, and Giovanni Semeraro. Concept-based item representations for a cross-lingual content-based recommendation process. Information Sciences 374 (2016): 15-31.

[223] Lops, Pasquale, Cataldo Musto, Fedelucio Narducci, Marco De Gemmis, Pierpaolo Basile, and Giovanni Semeraro. Mars: a multilanguage recommender system. In Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 24-31. 2010.

[224] Magnini, Bernardo, and Carlo Strapparava. Improving user modelling with content-based techniques. In International Conference on User Modeling, pp. 74-83. Springer, Berlin, Heidelberg, 2001.

[225] Schmidt, Sebastian, Philipp Scholl, Christoph Rensing, and Ralf Steinmetz. Cross-lingual recommendations in a resource-based learning scenario. In European Conference on Technology Enhanced Learning, pp. 356-369. Springer, Berlin, Heidelberg, 2011.

[226] Martinez-Cruz, Carmen, Carlos Porcel, Juan Bernabé-Moreno, and Enrique Herrera-Viedma. A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling. Information Sciences 311 (2015): 102-118.

[227] Takasu, Atsuhiro. Cross-lingual keyword recommendation using latent topics. In Proceedings of the 1st international workshop on information heterogeneity and Fusion in recommender systems, pp. 52-56. 2010.

[228] Joulin, Armand, Piotr Bojanowski, Tomáš Mikolov, Hervé Jégou, and Édouard Grave. Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2979-2984. 2018.

[229] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural machine

translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015.

[230] He, Ruidan, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 388-397. 2017.

[231] Lee, Kuang-Huei, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 201-216. 2018.

[232] Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In Proceedings of NAACL-HLT, pp. 866-875. 2016.

[233] Pappas, Nikolaos, and Andrei Popescu-Belis. Explicit document modeling through weighted multiple-instance learning. Journal of Artificial Intelligence Research 58 (2017): 591-626.

[234] Angelidis, Stefanos, and Mirella Lapata. Multiple instance learning networks for fine-grained sentiment analysis. Transactions of the Association for Computational Linguistics 6 (2018): 17-31.

[235] Shen, Tao, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[236] Lu, Jiasen, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In Advances in neural information processing systems, pp. 289-297. 2016.

[237] Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th international conference on World wide web, pp. 811-820. 2010.

[238] Alam, Md Hijbul, Woo-Jong Ryu, and SangKeun Lee. Joint multi-grain topic sentiment: modeling semantic aspects for online reviews. Information Sciences 339 (2016): 206-223.

[239] Pontiki, Maria, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub et al. Semeval-2016 task 5: Aspect based sentiment analysis. In 10th International Workshop on Semantic Evaluation (SemEval 2016). 2016.

[240] Ziegler, Cai-Nicolas, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In Proceedings of the 14th international conference on World Wide Web, pp. 22-32. 2005.

[241] Niemann, Katja, and Martin Wolpers. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 955-963, 2013.

[242] He, Xiangnan, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: Neural attentive item similarity model for recommendation. IEEE Transactions on Knowledge and Data Engineering 30, no. 12 (2018): 2354-2366.

[243] Seo, Sungyong, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the eleventh ACM conference on recommender systems. ACM, pp. 297-305, 2017.

[244] Wang, Yequan, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 606-615. 2016.

[245] Hu, Mengting, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. CAN: Constrained Attention Networks for Multi-Aspect Sentiment Analysis. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 4593-4602. 2019.

[246] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.

[247] Deshpande, Mukund, and George Karypis. Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22, no. 1 (2004): 143-177.

[248] Rendle, Steffen, and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In Proceedings of the third ACM international conference on Web search and data mining. ACM, pp. 81-90, 2010.

[249] Vinagre, João, Alípio Mário Jorge, and João Gama. Fast incremental matrix factorization for recommendation with positive-only feedback. In International Conference on User Modeling, Adaptation, and Personalization, pp. 459-470. Springer, Cham, 2014.

[250] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pp. 3111-3119. 2013.

[251] Fagin, Ronald, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. Journal of computer and system sciences 66, no. 4 (2003): 614-656.

[252] Le, Quoc, and Tomas Mikolov. Distributed representations of sentences and documents. In International conference on machine learning, pp. 1188-1196. 2014.

[253] Djuric, Nemanja, Hao Wu, Vladan Radosavljevic, Mihajlo Grbovic, and Narayan Bhamidipati. Hierarchical neural language models for joint representation of streaming documents and their content. In Proceedings of the 24th international conference on World Wide Web. ACM, pp. 248-255, 2015.

[254] Jeh, Glen, and Jennifer Widom. Scaling personalized web search. In Proceedings of the 12th international conference on World Wide Web. ACM, pp. 271-279, 2003.

[255] Recht, Benjamin, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In Advances in neural information processing systems, pp. 693-701. 2011.

[256] Stefanidis, Kostas, Eirini Ntoutsi, Mihalis Petropoulos, Kjetil Nørvåg, and Hans-Peter Kriegel. A framework for modeling, computing and presenting time-aware recommendations. In Transactions on Large-Scale Data-and Knowledge-Centered Systems X, pp.146-172. Springer, Berlin, Heidelberg, 2013.

[257] De Meo, Pasquale, Antonino Nocera, Giorgio Terracina, and Domenico Ursino. Recommendation of similar users, resources and social networks in a Social Internetworking Scenario. Information Sciences 181, no. 7 (2011): 1285-1305.

[258] Buccafurri, Francesco, Gianluca Lax, Serena Nicolazzo, and Antonino Nocera. A

model to support design and development of multiple-social-network applications. Information Sciences 331 (2016): 99-119.

[259] Zhang, Jiawei, and Philip S. Yu. Pct: partial co-alignment of social networks. In Proceedings of the 25th International Conference on World Wide Web. ACM, pp.749-759, 2016.

[260] Jia, Yongpo, Xuemeng Song, Jingbo Zhou, Li Liu, Liqiang Nie, and David S. Rosenblum. Fusing social networks with deep learning for volunteerism tendency prediction. In Thirtieth AAAI conference on artificial intelligence. 2016.

[261] Hu, Yifan, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE International Conference on Data Mining. IEEE, pp. 263-272, 2008.

[262] Xie, Min, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning graph-based poi embedding for location-based recommendation. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. ACM, pp.15-24, 2016.

[263] Cremonesi, Paolo, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the fourth ACM conference on Recommender systems. ACM, pp. 39-46, 2010.

[264] Covington, Paul, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems. ACM, pp. 191-198, 2016.

[265] Li, Yongjin, and Jagdish C. Patra. Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. Bioinformatics 26, no. 9 (2010): 1219-1224.

[266] Gao, Yang, Jianfei Chen, and Jun Zhu. Streaming gibbs sampling for LDA model. arXiv preprint arXiv:1601.01142 (2016).

[267] Lin, Yu-Ru, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: community discovery via relational hypergraph factorization. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 527-536, 2009.

[268] Ahn, Yong-Yeol, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. Nature 466, no. 7307 (2010): 761-764.

[269] Yang, Jaewon, and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In Proceedings of the sixth ACM international conference on Web search and data mining. ACM, pp. 587-596, 2013.

[270] Zhang, Hongyi, Irwin King, and Michael R. Lyu. Incorporating implicit link preference into overlapping community detection. In Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.

[271] Tintarev, Nava, and Judith Masthoff. Designing and evaluating explanations for recommender systems. In Recommender systems handbook, pp. 479-510. Springer, Boston, MA, 2011.

[272] Zhang, Yongfeng, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, pp. 83-92, 2014.

[273] Wang, Nan, Hongning Wang, Yiling Jia, and Yue Yin. Explainable recommendation via multi-task learning in opinionated text data. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 165-174. 2018.

[274] Chen, Chong, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference. ACM, pp. 1583-1592, 2018.

[275] Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734. 2014.

[276] Liu, Qiang, Shu Wu, and Liang Wang. Multi-behavioral sequential prediction with recurrent log-bilinear model. IEEE Transactions on Knowledge and Data Engineering 29, no. 6 (2017): 1254-1267.

[277] Herlocker, Jonathan L., Joseph A. Konstan, and John Riedl. Explaining collaborative

filtering recommendations. In Proceedings of the 2000 ACM conference on Computer supported cooperative work. ACM, pp. 241-250, 2000.

[278] Ling, Guang, Michael R. Lyu, and Irwin King. Ratings meet reviews, a combined approach to recommend. In Proceedings of the 8th ACM Conference on Recommender systems. ACM, pp. 105-112, 2014.

[279] Bao, Yang, Hui Fang, and Jie Zhang. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. In Twenty-Eighth AAAI conference on artificial intelligence. 2014.

[280] Tay, Yi, Anh Tuan Luu, and Siu Cheung Hui. Multi-pointer co-attention networks for recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp. 2309-2318, 2018.

[281] Ai, Qingyao, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms 11, no. 9 (2018): 137.

[282] Hu, Liang, Songlei Jian, Longbing Cao, and Qingkui Chen. Interpretable Recommendation via Attraction Modeling: Learning Multilevel Attractiveness over Multimodal Movie Contents. In IJCAI, pp. 3400-3406. 2018.

[283] Wang, Pengfei, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, pp. 403-412, 2015.

[284] Kumar, Ankit, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In International conference on machine learning, pp. 1378-1387. 2016.

[285] Grefenstette, Edward, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded memory. In Advances in neural information processing systems, pp. 1828-1836. 2015.

[286] Zhu, Yu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to Do Next: Modeling User Behaviors by Time-LSTM. In IJCAI, vol. 17, pp.

3602-3608. 2017.

[287] Wang, Hongwei, Fuzheng Zhang, Xing Xie, and Minyi Guo. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 world wide web conference. AM, pp. 1835-1844, 2018.

[288] Pei, Wenjie, Jie Yang, Zhu Sun, Jie Zhang, Alessandro Bozzon, and David MJ Tax. Interacting attention-gated recurrent networks for recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, pp. 1459-1468, 2017.

[289] Li, Jing, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, pp. 1419-1428, 2017.

[290] Sun, Mingxuan, Fei Li, and Jian Zhang. A multi-modality deep network for cold-start recommendation. Big Data and Cognitive Computing 2, no. 1 (2018): 7.

[291] Bucak, Serhat S., Rong Jin, and Anil K. Jain. Multiple kernel learning for visual object recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, no. 7 (2013): 1354-13.

[292] Poria, Soujanya, Iti Chaturvedi, Erik Cambria, and Amir Hussain. Convolutional MKL based multimodal emotion recognition and sentiment analysis. In 2016 IEEE 16th international conference on data mining (ICDM). IEEE, pp. 439-448, 2016.

[293] Zadeh, Amir, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor Fusion Network for Multimodal Sentiment Analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1103-1114. 2017.

[294] Nojavanasghari, Behnaz, Deepak Gopinath, Jayanth Koushik, Tadas Baltrušaitis, and Louis-Philippe Morency. Deep multimodal fusion for persuasiveness prediction. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, pp. 284-288. 2016.

[295] Wörtwein, Torsten, and Stefan Scherer. What really matters—An information gain analysis of questions and reactions in automated PTSD screenings. In 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), pp.

15-20. IEEE, 2017.

[296] Wöllmer, Martin, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun, Kenji Sagae, and Louis-Philippe Morency. Youtube movie reviews: Sentiment analysis in an audio-visual context. IEEE Intelligent Systems 28, no. 3 (2013): 46-53.

[297] Liang, Hongru, Haozheng Wang, Jun Wang, Shaodi You, Zhe Sun, Jin-Mao Wei, and Zhenglu Yang. JTAV: Jointly Learning Social Media Content Representation by Fusing Textual, Acoustic, and Visual Features. In Proceedings of the 27th International Conference on Computational Linguistics, pp. 1269-1280. 2018.

[298] Zhang, Shiqing, Shiliang Zhang, Tiejun Huang, Wen Gao, and Qi Tian. Learning affective features with a hybrid deep model for audio–visual emotion recognition. IEEE Transactions on Circuits and Systems for Video Technology 28, no. 10 (2017): 3030-3043.

[299] Chen, Minghai, Sen Wang, Paul Pu Liang, Tadas Baltrušaitis, Amir Zadeh, and Louis-Philippe Morency. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, pp. 163-171. 2017.

[300] Poria, Soujanya, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 873-883. 2017.

[301] Guan, Yue, Qiang Wei, and Guoqing Chen. Deep learning based personalized recommendation with multi-view information integration. Decision Support Systems 118 (2019): 58-69.

[302] Tan, Yunzhi, Min Zhang, Yiqun Liu, and Shaoping Ma. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In IJCAI, vol. 16, pp. 2640-2646. 2016.

[303] Zhang, Wei, and Jianyong Wang. Integrating topic and latent factors for scalable personalized review-based rating prediction. IEEE Transactions on Knowledge and Data Engineering 28, no. 11 (2016): 3013-3027.

[304] Wang, Guolong, Junchi Yan, and Zheng Qin. Collaborative and Attentive Learning for Personalized Image Aesthetic Assessment. In IJCAI, pp. 957-963. 2018.

[305] Cheng, Zhiyong, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S. Kankanhalli. A^ 3NCF: An Adaptive Aspect Attention Model for Rating Prediction. In IJCAI, pp. 3748-3754. 2018.

[306] Lian, Jianxun, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. Towards Better Representation Learning for Personalized News Recommendation: a Multi-Channel Deep Fusion Approach. In IJCAI, pp. 3805-3811. 2018.

[307] Gu, Yue, Xinyu Li, Kaixiang Huang, Shiyu Fu, Kangning Yang, Shuhong Chen, Moliang Zhou, and Ivan Marsic. Human conversation analysis using attentive multimodal networks with hierarchical encoder-decoder. In Proceedings of the 26th ACM international conference on Multimedia. ACM, pp. 537-545, 2018.

[308] Kiros, Ryan, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014).

[309] Ding, Yi, and Xue Li. Time weight collaborative filtering. In Proceedings of the 14th ACM international conference on Information and knowledge management. ACM, pp. 485-492, 2005.

[310] Chen, Xu, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. Visually explainable recommendation. arXiv preprint arXiv:1801.10288 (2018).

[311] Gu, Yue, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. Hybrid Attention based Multimodal Network for Spoken Language Classification. In Proceedings of the 27th International Conference on Computational Linguistics, pp. 2379-2390. 2018.

[312] Pan, Yingwei, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4594-4602. 2016.

[313] Bergstra, James, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A CPU and GPU math compiler in Python. In Proc. 9th Python in Science Conf, vol. 1, pp. 3-10. 2010.

[314] Powers, D. M. W. Evaluation: from precision, recall and f-factor to roc.

Informedness, Markedness & Correlation (Tech. Rep.), 2007.

[315] Järvelin, Kalervo, and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 41-48, 2000.

[316] Wang, Chong, and David M. Blei. Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 448-456, 2011.

[317] Tintarev, Nava, and Judith Masthoff. A survey of explanations in recommender systems. In 2007 IEEE 23rd international conference on data engineering workshop. IEEE, pp. 801-810, 2007.

[318] Gao, Ming, Leihui Chen, Xiangnan He, and Aoying Zhou. Bine: Bipartite network embedding. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 715-724. 2018.

[319] Meng, Xuying, Suhang Wang, Kai Shu, Jundong Li, Bo Chen, Huan Liu, and Yujun Zhang. Personalized privacy-preserving social recommendation. In 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 (pp. 3796-3803). AAAI press.

[320] Jiang Feng , Min Gao, Qingyu Xiong, Junhao Wen and Yi Zhang. Robust Social Recommendation Techniques: A Review. In Socially Aware Organisations and Technologies, Impact and Challenges. ICISO 2016. IFIP Advances in Information and Communication Technology, vol 477. Springer, Cham.

[321] Zhang, Xiang-Liang, Tak Man Desmond Lee, and Georgios Pitsilis. Securing recommender systems against shilling attacks using social-based clustering. Journal of Computer Science and Technology, 28(4), pp. 616-624, 2013.

[322] Nie, Liqiang, Luming Zhang, Meng Wang, Richang Hong, Aleksandr Farseev, and Tat-Seng Chua. Learning user attributes via mobile social multimedia analytics. ACM Transactions on Intelligent Systems and Technology (TIST), 8(3), pp.1-19, 2017.