Automatically Facilitating Discussion in Online Forums

Kishaloy Halder

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2019

Supervisor:
Associate Professor Min-Yen Kan

Thesis Examiners:
Dr Zhao Jin
Professor Lee Wee Sun
Associate Professor Chirag Shah, Rutgers University

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Kishaloy Halder

December 2019

To the eternal human curiosty

Abstract

Online discussion forums provide users a platform to learn from the collective wisdom of the community. Forum users ask questions, share anecdotal observations with others in the community, in the hope of getting relevant information from them. The growing number of users, turning to these forums for fulfilling their information need and the continuous influx of new topics to discuss, pose significant challenges in making the discussion forums run in an efficient manner. In this thesis, we investigate how Natural Language Processing, and Information Retrieval techniques such as Recommendation Engines can be designed to help the users navigate through the online discussion forums efficiently.

Firstly, we propose building a recommendation system to improve the visibility of threads in online discussion forums. We develop a probabilistic graphical model to consider the interests explicitly mentioned by the user to recommend her posts she is likely to be interested in. We also show that our framework can provide explanation behind the recommendations. Unlike traditional recommendation system settings, discussion forum also suffers from new posts being generated all the time. We propose a deep neural network based framework that can represent a post based on the words used in it and utilize them to identify the potentially interested users for it. We propose to address this problem as an Extreme Multi-Class Multi-Labelling problem and show that this formulation works well in practice.

The open nature of the online forums attracts a large number of users to participate in the discussion. Although this is desirable, often the large number of posts in response to a discussion topic quickly becomes unmanageable as many repetitive or even irrelevant posts are frequently posted. To this end, we propose a neural network based framework that can identify helpful posts in a discussion thread automatically. Experimenting with large real-world datasets we show that our model performs significantly better compared to existing state-of-the-art solutions.

Publications Contributing to this Thesis

The following works have been published as peer-reviewed publications.

1. Chapter 3 – **Kishaloy Halder**, Min-Yen Kan, and Kazunari Sugiyama. "Health Forum Thread Recommendation Using an Interest Aware Topic Model." Proceedings of the ACM on Conference on Information and Knowledge Management, 2017. [25]

2. Chapter 4 – **Kishaloy Halder**, Lahari Poddar, and Min-Yen Kan. "Cold Start Thread Recommendation as Extreme Multi-label Classification." Companion of the The Web Conference, 2018. [28]

3. Chapter 5 – **Kishaloy Halder**, Min-Yen Kan and Kazunari Sugiyama. "Predicting Helpful Posts in Open-Ended Discussion Forums: A Neural Architecture." Proceedings of the 2019 North American Chapter of the Association for Computational Linguistics. NAACL, 2019. [26]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

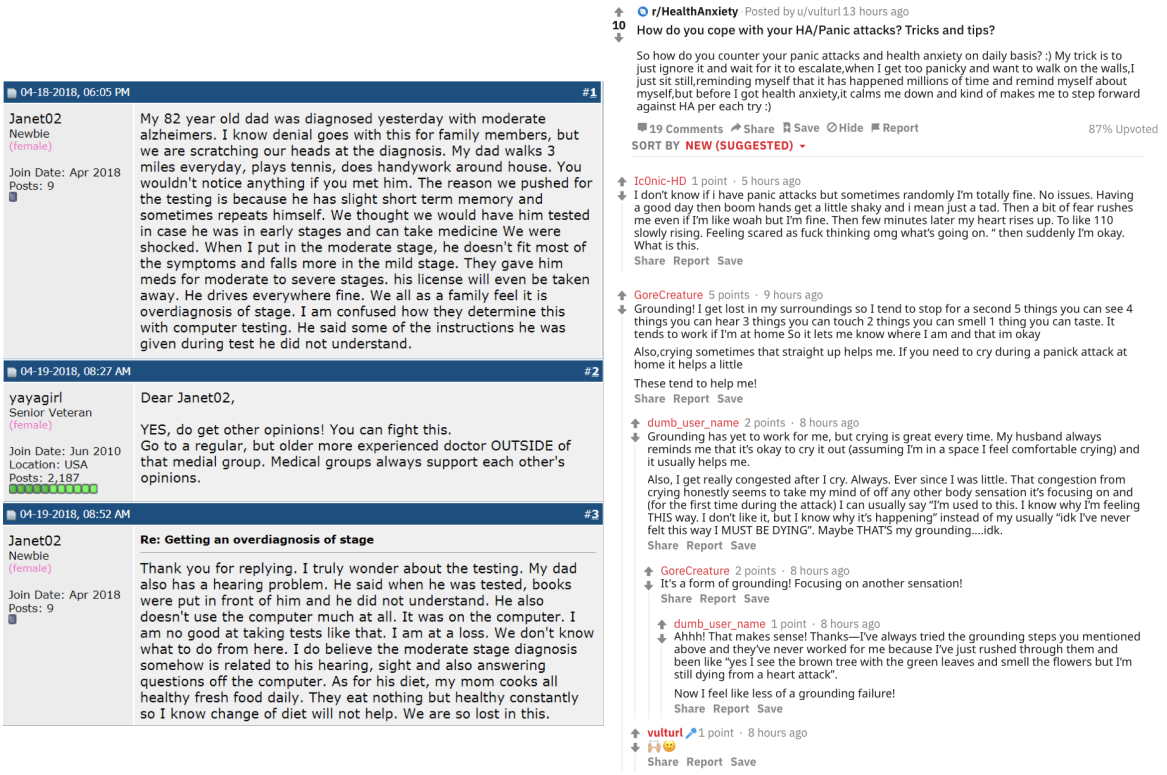
Online discussion forums have become an important social media platform across many domains such as health[1], education[2], technical question answering[3], e-commerce, generic trending affairs[4], government policy making to name a few. Users take part in these forums to learn from the collective wisdom of the community-users, by posting a question or asking others for opinions on a certain topic. Apart from the specific forum users, the public forums are often indexed by popular search engines, and thus cater to a larger audience as they show up in search results in response to user queries. The discussion forum platforms have proven to be socially transformative for troubleshooting and personal well being in many domains [74, 84, 27]. As a result, platforms such as reddit has seen one of the highest user engagement rates among the social media channels [36].

1.1 Background

Before we discuss the issues pertaining to these discussion forums, let us first provide some background on the structure of them for the unfamiliar readers. Discussions in these forums happen in the form of *threads* as shown in Figure 1.1.

A thread consists of many individual *posts* written by the forum users at different points

[1] https://www.healthboards.com/boards/
[2] https://www.coursera.org
[3] https://www.stackoverflow.com
[4] https://www/reddit.com



(a) HealthBoards (b) Reddit

Figure 1.1: Illustrative public threads from (a)`HealthBoards`, and (b)`Reddit`. By default, most platforms present a chronologically sorted list of posts in the discussion thread. The thread structure is linear in `HealthBoards`, but forms a tree in case of `Reddit`.

of time. Most commonly the posts are presented in a chronologically sorted order, whereas some platforms present them sorted according to their "upvotes" (or other similar scores such as "likes", "mark as helpful" etc). The structure of the thread varies as well depending on the platform. Typically forums within a niche community (such as e-health) keep a simple linear structure (*cf* Figure 1.1a). Platforms with a wider audience *e.g.,* reddit, keep a tree-like structure where a new post can be written in response to any of the past posts (*cf* Figure 1.1b). In our study, we have observed that despite these differences, they share a key characteristic of discussion. The first post usually consists of questions or some anecdotal experience shared by a user who seeks answers or opinions from others in these communities. Others join in the discussion by sharing their experiences, responding to the queries, or to make casual comments on the topic with sarcasm, humour or both. Depending on the platform, the thread presentation format, and usage of different types of multimedia varies.

In this thesis, we would focus on the textual mode of communication which is the most

common medium across forums from all the domains mentioned earlier. We also observe that a plethora of user-interactions exist in different platforms. In the example shown in Figure 1.1, `reddit` allows users to "share", "save" a thread/post if they like it, providing reputation scores *e.g.,* "karma" for the users. We acknowledge that these signals could potentially be useful in understanding the dynamics of the community and the content produced. However these are often tightly coupled with individual platforms (*e.g.,* aforementioned `reddit` features are not present in `HealthBoards`), hence difficult to generalize from the modelling perspective. We investigate some of the salient features shared by most of the online discussion forums, and propose solutions to improve the challenges associated with scaling them, based on recent advancements in natural language processing (NLP), and information retrieval (IR) techniques.

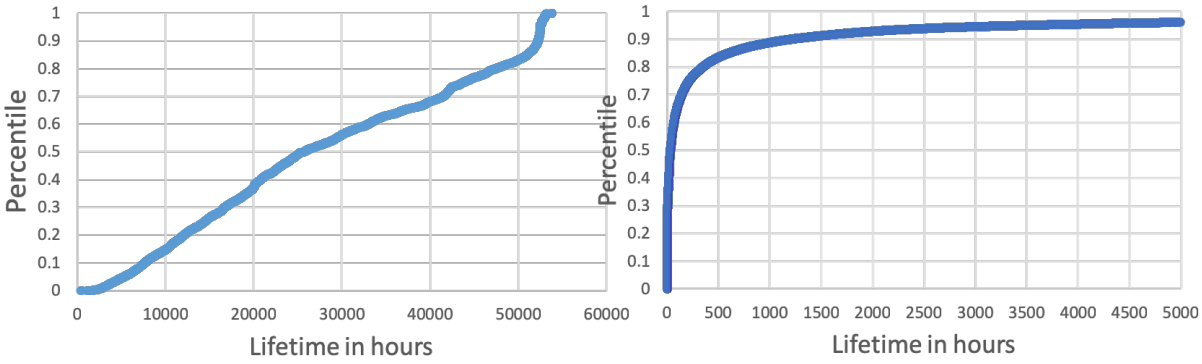
1.2 Issues Pertaining to Discussion Forums

Formidable challenges exist that hinder the effectiveness of online discussion forums. We categorize them into the following.

Finding The "Right" Thread: The abundance of user generated content in these forums, explosive growth of user-base [36], and lack of structure [66] give rise to a scalability problem, making it inefficient for the users to navigate through. The task of filtering threads to one's personal preferences falls in the guise of recommendation systems (RS). Popularized by the Netflix Prize[5] competition, RSs typically concern two sets of entities i.e., users, and items (e.g., movies, products, books and so on). Research around RS has seen significant advancements through the last decade, with the advent of various feature based, and model based algorithms such as collaborative filtering (CF). A specific set of CF based algorithms were also proposed for textual items such as abstracts of scientific papers [101]. However we find that the unique challenges of discussion forums were largely unaddressed.

Influx of New Threads: The continuous influx of newly created threads makes it difficult for a user to find the threads relevant to her interests or information need. Traditional CF

[5] `https://en.wikipedia.org/wiki/Netflix_Prize`



(a) Movies in `Netflix` (b) Threads in `HealthBoards`

Figure 1.2: Distribution of lifetime of movies (left), and threads (right). Threads have a much shorter lifetime. The 80^{th} percentile lifetimes are 5.7 years, and 16 days for movies and threads respectively.

based algorithms often struggle to perform well when there is limited information about users or items, referred to as *cold start problem* in the literature. We observe that this challenge becomes particularly overwhelming in case of discussion forums because (i) new threads are being posted on the web continuously, (ii) the average lifetime[6] of a thread is much shorter as shown in Figure 1.2. This indicates,

If a thread does not get enough interactions quickly after being posted, it is likely to remain unnoticed forever.

We note that the 80^{th} percentile lifetime for a thread in `HealthBoards` is mere 16 days compared to that of 5.7 years in case of movies. The constant influx of new thread encompassing ever changing discussion topics on the web could be a major limiting factor for these forums.

Varying Quality: Many threads in these forums receive repetitive, or even irrelevant posts which constitute a sub-optimal experience for the users going through the threads to satisfy their information needs [32]. Apart from these, the discussion forums are also plagued with non-informative, sarcastic, troll posts [62, 10]. There has been sporadic efforts to improve this aspect in a handful of domains [87, 70, 69]. However it often requires domain expertise to gather ample annotated data which is both expensive and difficult to generalize.

Safety Issues: The *openness* of online platforms such as discussion forums promotes de-

[6]the time difference between the oldest and newest post

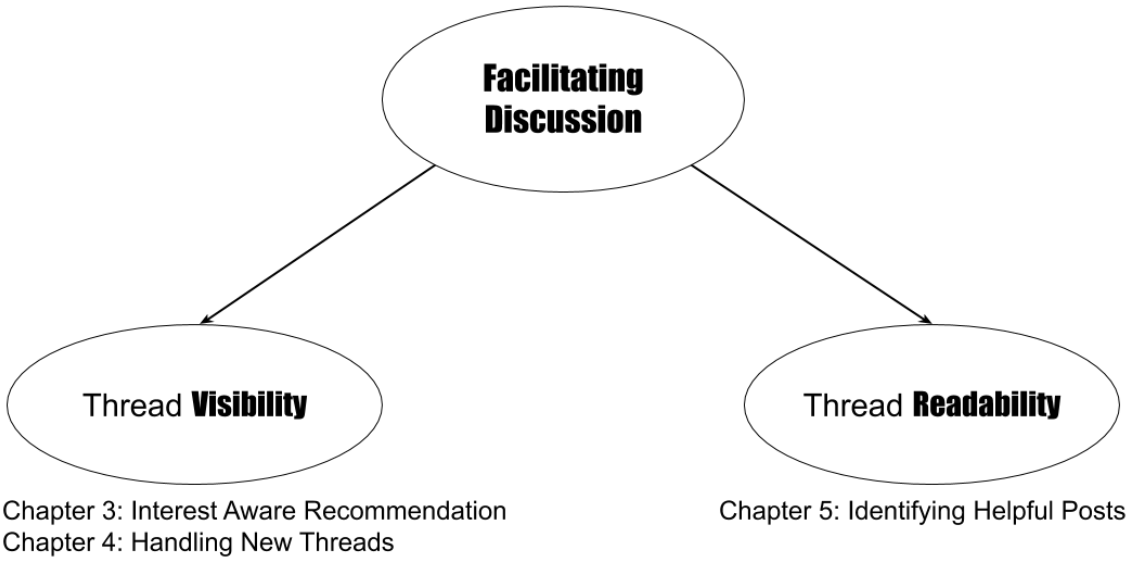


Figure 1.3: Thesis Organization

mocratization of human expertise through massive amounts of user generated content (UGC). Nevertheless, as any other UGC platform, discussion forums also suffer since there are no "gatekeepers". We note that many techniques have been proposed in the literature to combat the peripheral safety issues in this regard, such as spam, profanity, hate speech detection [37, 67]. There are certain behavioral traits of open community members as well that are studied in the literature through surveys in the past such as personal attack, bullying [32, 24]. While we acknowledge that these are important issues to address, they do not help the user much to understand the content being discussed as such.

1.3 Thesis Contribution and Organization

We aim to address these challenges in this thesis through systematic study of the underlying problems and providing certain automated solutions. We contribute in facilitating discussion in online forums by making improvements in two major areas (i) Thread Visibility, and (ii) Thread Readability.

Thread Visibility: We aim to improve the visibility of the threads by finding the right set of users that might be interested in them. We observe that users have interests that are explicitly mentioned in their profiles, as well as some implicit (unobserved) interests that are only

reflected in their thread interaction patterns and the textual content that they post themselves. To this end, we propose models towards understanding both these interests of the users. We employ probabilistic graphical models that incorporate explicitly observed signals such as user reported interests, and user profile text to make the recommended threads personalized to users' tastes. We would discuss this in detail in Chapter 3.

From our experiments, we find that the interest aware model is not enough for threads that are completely new and thus have no user-interaction history. We dedicate Chapter 4, to understand the underlying problem in such cases. We propose a supervised Extreme Multi-label classification (XMLC) approach to find the interested set of users for all the incoming threads in these forums. We combine ideas from state-of-the-art XMLC research as well as text processing frameworks such as a novel technique called *cluster sensitive attention* to achieve this.

Thread Readability: In the later part of the thesis, we investigate one step further. We believe that recommending the right set of threads to the users solves only half of the actual problem. The reading experience remains sub-optimal even after the recommendation process, since they still have to go through a long list of posts in those threads to get relevant information. In Chapter 5, we propose an automatic way of identifying the helpful posts *inside* a thread. We utilize the user provided helpfulness scores (such as "upvotes", "likes", "mark it as helpful" etc) in the past threads and build a supervised model to capture the nuances of the text used in a target post content to determine how helpful it would be perceived as to the user. Not only this approach could aid the user read an entire thread efficiently, but it can also improve the user engagement since the interested users can be notified automatically every time a helpful reply is posted in a thread.

We validate all our proposed solutions by rigorous experiments with large real-world datasets from multiple domains such as e-health, massive open online courses (MOOCs), `reddit` and so on. We show that our models outperform the state-of-the-art baseline systems comfortably. We provide a detailed literature review in Chapter 2 to understand relevant research in this direction. Finally we conclude our work and present some future works along with a few caveats to make progress in this field in Chapter 6. We believe that

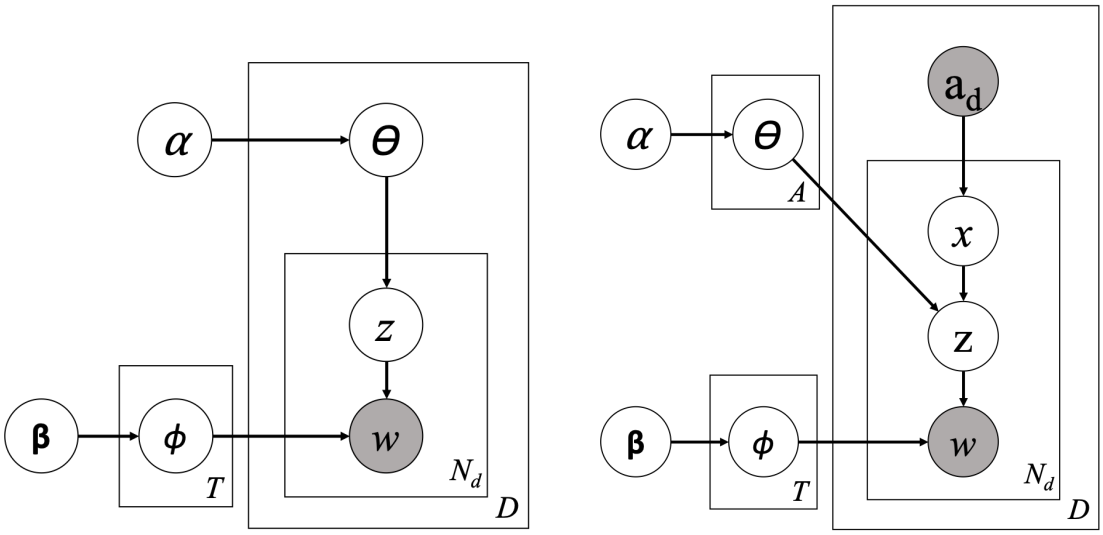
our work is timely. With more and more people making use of discussion forums over the internet, our work could benefit this significant portion of world wide web users.

Chapter 2

Literature Review

This thesis encompasses research on two domains, (i) Recommendation Systems, and (ii) Text Classification to facilitate the process of finding personalized helpful posts from online discussion forums. In the following, we would give some background on these two, and would discuss some existing works that have tried to bridge the gap between them.

We note that there exists some systematic studies from specific domains such as e-health in recent years [7, 109, 51]. However, to the best of our knowledge, such work has been limited to large-scale surveys of self-reported behavior, and the community has not seen much development of practical recommendation techniques for online discussion forums as of current. While acknowledging the varied societal and emotional support needs of users, we find value in addressing the primary information need for the forum users. As such, our task falls into the guise of recommendation systems, an area which has seen much recent interest with the popularity of Web 2.0 systems that integrate users and items into Web applications. For brevity, we limit our discussion to relevant prior work in the areas of topic modeling, content-based and context-aware recommendations, and community question answering. We would use "item" and "thread" interchangeably since in our context we aim to recommend threads to the users.



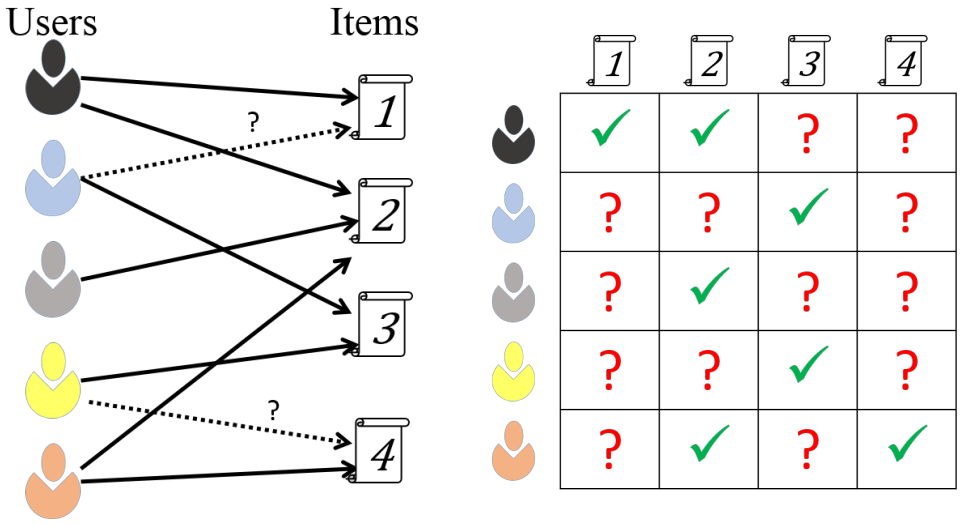
(a) LDA (b) Author Topic Model

Figure 2.1: Generative model for (a) Latent Dirichlet Allocation [13], (b) Author Topic Model [81]. Variables in shaded circles are observed, others remain latent and are learnt from the data.

2.1 Topic Models

Topic Models provide a probabilistic framework to cluster textual documents (such as news, wikipedia, or in our case, discussion threads, user posts etc) according to their hidden semantic structure ("topics") in an unsupervised manner. They regard documents as mixtures of latent topics with certain distributional properties. For textual documents, several works have focused on modeling latent factors of the content using latent Dirichet allocation (LDA) [13] and its variants [81, 60, 78].

LDA assumes that every textual document has a few underlying topics, and every topic is defined by a probability distribution of words occurring in it (*cf.* Figure 2.1a). LDA being relatively inexpensive to train on large corpora, has been rendered as one of the most commonly used text mining tool in many domains such as news, social media, scholarly documents to name a few. Later on, other derivatives were proposed e.g., the author–topic model [81] learns the topic distribution of authors for a set of documents (*cf.* Figure 2.1b). On the other hand, labeled LDA relies on annotated tags to constrain the possible topics for each document [78].



(a) Interaction Graph. (b) Interaction Matrix.

Figure 2.2: Illustration of a Recommendation Problem. (a) Solid Edges represent user interactions. Dotted Edges represent some of the missing interactions. (b) In interaction matrix: '✓' ⇒ interaction, '?' ⇒ no interaction.

While these models are useful on their own for modeling either users or items, they do not capture the dynamics between both. LDA can be used as a starting point for refinement to account these factors. Agarwal *et al.* [1] leveraged LDA-discovered latent topic distributions for matrix factorization-based collaborative filtering (CF). They report modest improvement over other methods — the reason being that often the topic distributions of different items look similar, even though they appeal to different sets of people. There also exists a set of focused topic models that cater to specific use cases [19, 111]. Chen *et al.* [19] proposed a Contextual Focused Topic Model, where they assume a word to be generated from either the author or the venue or the document characteristics – not from a joint combination of them, as in our case.

2.2 Recommendation Systems

Recommendation Systems can be considered as a basis to find content relevant to a user's taste in online discussion forums (*cf.* Figure 2.2). Research on Recommendation Systems in general has seen significant progress during the last decade.

Collaborative Filtering (CF) based approaches have been the most popular across domains. Matrix Factorization based methods [49, 47, 63, 82, 105] map users and items into a shared latent feature space and compute the inner product of their latent vectors to reflect the interactions between users and items i.e. some form of ratings. Lately, deep learning based models are proposed to learn the user, and item representations from the historical interactions[30]. CF based approaches suffer when the data is sparse (i.e. limited amount of interaction history available for a user or item) and do not work for cold start (i.e. *no* interaction history available for a new item or user).

Certain content-based recommendation systems further account for information associated with content associated with users. Collaborative Topic Regression (CTR) [102] proposes an elegant method of using the textual content for recommendation. It is a probabilistic graphical model that integrates a topic model, latent Dirichlet allocation (LDA) [14] for modeling contents of a document, and uses the LDA-discovered topics while doing the regression later with probabilistic matrix factorization (PMF) [63]. In the same context, Charlin *et al.* [18] showed that the cold-start performance of a similar model can be improved if there is bootstrapping information available in the form of document content associated with users. Although it is possible to use a textual content agnostic off-the-shelf CF method to recommend articles to a user to comment on [88], considering associated textual content improves performance significantly in other platforms, *e.g.,* news articles or blogs [6], demonstrating the efficacy of modeling such side information.

Along with the past user–item interaction history, *Context Aware Recommendation* methods consider the interaction contexts which can be equated to the self-declared interests of a user (e.g., "politics", "sports" for a user of news mediums; "diabetes", "cancer" for a user in health forums) in our scenario. Tensor Factorization [42] and Factorization Machines [79] are two promising methods, primarily designed to predict ratings in an explicit feedback-based system. Nguyen *et al.* [65] demonstrate that such techniques can also be profitably applied in implicit feedback scenarios such as ours. In community question answering systems, prior work has addressed recommending semantically related question threads that reflect different aspects of the user's query and provide supplementary information. Wang

et al. [106] recommend more relevant threads by extending a language model with the popularity of a question. Pedro and Karatzoglou [73] extend Learning to Rank to supervised LDA applied specifically to recommend relevant question threads. Zhou *et al.* [118] propose a translation model-based thread recommendation by incorporating answer information. In recent work, Omari *et al.* [71] and Palotti *et al.* [72] improve ranking of relevant discussion threads in health forums. However, both works do not address the recommendation of relevant threads to specific user's interests.

Cold-Start Aware Recommendation System: In order to make recommendations for a new item (thread in our case), in absence of any interaction history, the recommender system needs to make use of additional information such as item content or metadata. Similar to our setting, the authors in [89] tackle the problem of recommending incoming news articles for users to comment. However, they do not use the whole article content but only use the tags associated with a document. This approach could be difficult to generalize as in our case the forums are open to anyone as opposed to a news article, which is subject to expert curation before publication.

Some recent works [98, 107, 104, 52, 100, 105] have explored deep learning models for recommendation based on item content. In [98] the authors use CNNs to model the acoustic signals present in a music video in order to predict the latent factors to be used by a CF model to make recommendation. Similar to CTR, Collaborative Deep Learning (CDL) has been proposed that uses stacked denoising autoencoders (SDAE) [99] for representation learning of the textual content, and collaborative filtering for the rating matrix. To eliminate the bag-of-words assumption of CDL, Collaborative Recurrent Autoencoder (CRAE) [105] is proposed to model the sequence information in item content. For the denoising autoencoder based approaches mentioned above, the input is first corrupted by masking out some parts and then the neural network is used to reconstruct the original input by filling in the blank parts. The output of the bottleneck layer are regarded as features for the CTR model, and the whole network is optimized with additional fine-tuning. Instead of using a denoising autoencoder, CVAE [52] uses a Bayesian generative approach for the content representation

and reportedly outperforms the other methods. Recently, for cold-start recommendation, dropout is applied to input mini-batches, for training Deep Neural Networks to generalize for a missing input [100].

2.3 Extreme Multi-label Text Classification

Matching users with the contents that are relevant to them can also be considered as an Extreme Multi-label Classification (XMLC) problem. Embedding based approaches have proved to be popular for handling the extreme multi-label learning problem by reducing the effective number of labels. Generally, they assume that the label matrix is low-rank, and project label vectors into a lower dimensional subspace. Hence, instead of predicting the original high-dimensional label vector for each instance, they reliably train for prediction of embedded label vectors, and then employ a decompression algorithm to map the embedded label vectors back to the original label space. Various compression and decompression techniques have been proposed in the literature to achieve this [22, 41, 34, 117, 5, 11].

In order to avoid the loss of information during the compression phase of the embedding based approaches, tree-based methods have been proposed that try to partition the label space similar to a decision tree. It recursively partitions the huge label space in subtrees until only a few labels are left at each leaf node. A base classifier at each leaf node then focuses on only the active labels in the node. The LPSR [110] method focuses on learning a hierarchy over a base classifier or ranker starting with a base multi-label classifier for the entire label set - this becomes computationally expensive to train if a discriminative classifier (e.g. SVM) is used. Instead of using a base classifier, MLRF [2] uses an ensemble of randomized trees with a modified Gini index for partitioning the nodes. In FastXML [77] an NDCG-based objective is used at each node of the hierarchy for optimization.

Despite the success of deep learning in many fields, it has not been explored much for XMLC tasks. Recently, a CNN based approach (XML-CNN [55]) has been proposed, which uses convolutional layers for text representation and a feed forward layer acting as a bottle-neck layer for scalability. This has been shown to outperform both embedding based and

tree based approaches for XMLC - we would use this method for comparison in our experiments later.

2.4 Answer Quality in Community Question Answering Platforms

To the best of our knowledge, predicting helpful posts in generic open-ended discussion forums has not been studied before. However, some researchers have been working on similar directions; where they evaluate the *quality* (which may not correlate with perceived helpfulness by the community users) of posts in specific domains such as health [70, 69, 8] and online education [16, 17, 38]. External medical resources and thesauri such as UMLS[1] have been used to identify patterns of helpfulness in health [3]. In MOOC platforms, apart from the textual content of the forums, additional signals such as user reputation (*e.g.*, average homework scores, number of courses taken) have been used to estimate post quality [38]. However, these techniques are tightly coupled with the target domain, and may not be generalizable to new domains.

Past work has addressed the evaluation of answer quality in Community Question Answering (CQA) sites [39, 33, 87, 113, 71]. Typically posed as a classification problem, they use both textual and non-textual feature-based approaches. Since it is quite common for popular questions to attract many potential answers, answer ranking based on perceived quality is another line of approach [95, 12, 108]. Closer to our approach, Omari et al. [71] proposed a novelty-based greedy ranking algorithm that depends on a pre-trained parser to identify different propositions, useful for predicting helpfulness.

Often in the CQA answer quality evaluation literature, quality is measured through the human evaluators' annotations during experimentation [87, 70, 71]. However, we are interested in modeling the "helpfulness" for actual users in discussion forums (in term of "Upvotes", etc.) and not annotators following some predefined guidelines to mark answer

[1] https://www.nlm.nih.gov/research/umls/

quality, which might present other forms of bias.

Modeling Novelty in IR: Novelty detection in information retrieval, such as search result diversification [15, 92, 119, 23], is also prior art. Carbonell and Goldstein proposed maximal marginal relevance (MMR) to diversify the set of documents returned for a search query [15]. Similar approaches were also used later in Multi-Document Summarization (MDS) tasks [64]. These approaches address the problem either as a ranking task (ordering search results) or as a subset selection problem (such as MDS), where all documents are simultaneously made available. In contrast, in our discussion thread scenario, we need to model the discussion posts' sequential nature to understand the context of a later post and, in turn, determine its helpfulness.

Neural Network Based Models: Recently, neural network-based models have outperformed existing classifiers in many text classification tasks. They are widely adopted as they induce useful features on their own, given sufficient data. Although there are differences, the problem of answer selection is relevant: the goal is to rank the potential answers to a target question from multiple candidate answers in order of their similarity [114, 103, 86]. In our case, all posts in a thread are similar to the original post to an extent. Helpful posts are thus more difficult to identify; computing similarity is not viable as a single source solution.

To summarize the literature, we realize that while the previous work can handle recommendations in discussion forums, there is important evidence that needs to be modeled to achieve better recommendation accuracy [53, 115, 18]. In particular, in health forums, each user can often express explicit interests in different conditions. We note that, in most the of existing recommendation systems, construction of user profiles has been independent of the recommendation process itself. Inspired from the aforementioned works, we propose a unified framework of both user profiles and user participation in a discussion forum in Chapter 3 and 4. This might have implications not only on performance improvement over other state-of-the-art methods but also enhancement of transparency in the recommendation

task. Finally to improve the reading experience of the users, we understand that gauging the helpfulness of individual posts is the key. Inspired by all the previous works in answer quality determination from CQA, we propose a neural architecture to predict the helpfulness of posts in open-ended discussion forums in Chapter 5. To make it generic and easily adaptable to multiple domains, we study the problem from a linguistic viewpoint, where only the textual contents of the discussion threads would be considered.

Chapter 3

Thread Recommendation with Users' Interest Aware Topic Model

3.1 Introduction

We aim to improve the visibility of the discussion threads in this chapter. People participate in online health forums in part to discuss their symptoms and clinical conditions with others. They post health related questions to learn from the experience of the community. The majority of users participate in online health communities with the goal of meeting a medical information need [40, 61, 58]. This is the problem we address in this work. We acknowledge that patients also participate for emotional support and social reasons [7, 109, 27], but this is beyond the scope of our work. Finding relevant information can be difficult, and recommendation systems can help bridge this gap by providing users with discussion threads relevant to their condition- and symptom-specific interests.

We observe that the symptoms experienced by patients with different clinical conditions are often similar. However, the proper treatment crucially depends on the underlying cause (*i.e.*, the clinical condition or disease). This leads to many lexically similar user queries which require different answers as shown in Table 3.1. Many traditional approaches — such as topic models — struggle to identify the correct underlying condition, as they mainly use word co-occurrence to determine relevant answers.

ALS Threads	**Parkinson's Disease Threads**
is there anyone experiencing lower back pain while standing up after sitting for a while?...	... My lower back and hips pains a lot. It even hurts to walk some days ...
	Back pain has been bothering me for 4 years and it is getting worse. The lower back gets ...

Table 3.1: A query ALS thread (left), and lexically similar but unrelated posts for Parkinson's Disease (right).

We observe that context is a key factor to identify the appropriate latent conditions and symptoms. In this scenario, the key contextual evidence is the participation that a user manifests with respect to a specific medical condition, either by subscribing to a subforum related to a condition or by authoring a post in the forum inferrably related to a condition[1]. We believe that such context must be accounted for in order to recommend relevant discussion threads in health forums. We introduce a two-stage approach that captures such context.

We introduce a general, interest-aware topic model (IATM), in which known higher-level interests on topics expressed by each user can be modeled. We then specialize the IATM for use in consumer health forum thread recommendation by equating each user's self-reported medical conditions as interests and topics as symptoms of treatments for recommendation. The IATM additionally models the implicit interests embodied by users' textual descriptions in their profiles. To further enhance the personalized nature of the recommendations, we introduce jointly normalized collaborative topic regression (JNCTR) which captures how users interact with the various symptoms belonging to the same clinical condition.

Our solution leverages the topic model framework to properly incorporate the contextual information. Our topic model — which we term the *interest-aware topic model* (IATM) — is a general model that encompasses both the evidence of each user's thread and word interactions, but crucially, also the user's self-reported (and thus observed) interests. A key characteristic of the IATM is that even though it can model explicit user interests (*i.e.*, a

[1]In our scenario, we require actions that leave a traceable correlation with interest. This allows our framework to be applied even in cases where the recommendation is done by a third party (as done in our evaluation) and not necessarily done by the service provider.

patient is subscribed to a Parkinson's disease subforum), in the absence of such explicitly indicated interests, the IATM treats users' interests as a partially-observed random variable and attempts to infer the full and latent value. As users may not explicitly type themselves, yet actively participate, this is important to account for.

The IATM also natively models the side information of user profiles. User profiles are ubiquitous to many Web 2.0 sites, inclusive of health forums. In IATM, user profiles are treated as normal documents during the training process, used in determining the interests of the user. In our health forum recommendation scenario, user descriptions do give useful information about the user, which significantly aids the recommendation process, especially for users that have little interaction history — a form of cold start. In further analyses of our datasets, we note further modeling difficulties. Even when confined to a single condition, discussion on different symptoms also often appear similar due to commonly affected parts of the body. Consider the following posts:

(a) [about back pain] "*Have been suffering from back pain for couple of years now and it's getting worse -now I can barely put **weight** on the right **leg**...*"

(b) [about leg cramps] "*I get a lot of cramps in the **leg**. The only remedy is to stand up and put the body **weight** on it...*"

These posts have common words (bolded) but are about different symptoms of Parkinson's disease. Although the topic distributions in two posts (documents) are similar, each user's preferences are clearly directed towards different particular topics. We observe similar distribution bias with users' participation in clinical treatment discussions and other condition-specific topics.

To address this second, fine-grained disambiguation problem, we develop a novel graphical model, jointly normalized collaborative topic regression (JNCTR). JNCTR is a logical adaptation of the original collaborative topic regression (CTR) model [101], itself motivated to handle such divergences in each user's interests in documents with similar topic distributions. JNCTR extends CTR by taking both the user–topic and thread–topic distributions coming from IATM as input, but additionally accounts for the user–thread interaction his-

tory in the form of ratings. This model allows us to understand the differences between symptoms that originate from a single condition. JNCTR updates both the user–topic and thread–topic distributions based on the past user–thread interactions. We compute thread recommendations for each user using the resultant user–topic and thread–topic distributions.

Online health forum users often use their own words and phrases to describe their experiences [43, 66]. Standard medical ontologies and thesauri (*e.g.*, UMLS[2]) struggle to cover the medical terms found in user–generated medical content [31]. We believe that our specialized IATM+JNCTR model is the first attempt to understand how clinical conditions and their symptoms and treatments explain the interaction of users in a health forum.

In our experiments on two real-world consumer health forums, our proposed model significantly outperforms competitive state-of-the-art baselines by over 10% in recall. Importantly, we show that our IATM+JNCTR pipeline also imbues the recommendation process with added transparency, allowing a recommendation system to justify its recommendation with respect to each user's interest in certain health conditions.

The contributions of our work are summarized as follows:

• We formalize the problem of interest-aware recommendation, of which health forum thread recommendation is a specific instantiation of *conditions-as-interests*. We investigate how to best utilize user participation in the forum, formulating this as an implicit feedback-based recommendation problem.

• We apply our framework to a real-world dataset obtained from HealthBoards[3], demonstrating significant improvement over state-of-the-art baselines.

• We extend our experiments to demonstrate how our proposed IATM+JNCTR model deals gracefully with cold-start items ("threads" in our work). The model can explain a recommendation due to its modeling of latent variables. We further investigate how our model performs in recommendation justification by analyzing its recommendations to specific users.

[2] https://www.nlm.nih.gov/research/umls/
[3] http://www.healthboards.com

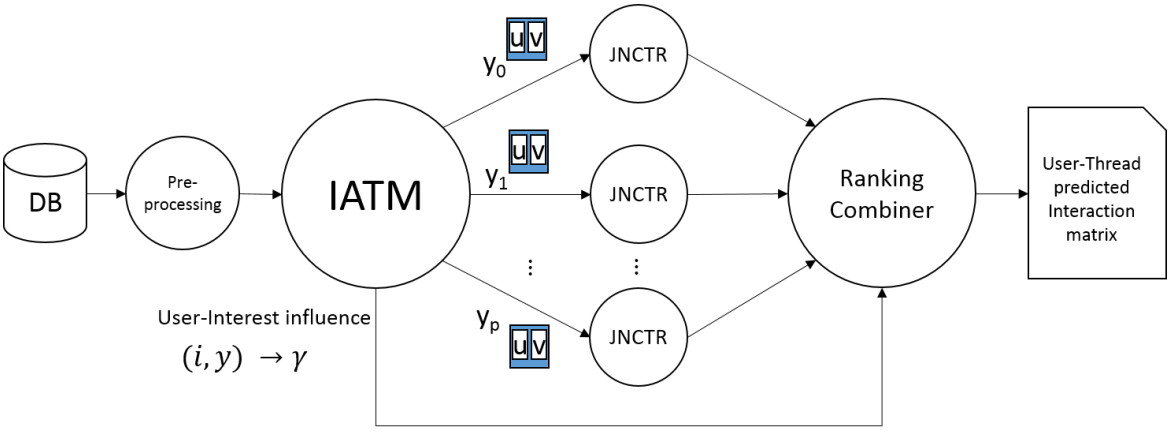


Figure 3.1: The pipeline for our three-stage recommendation framework. IATM first provides user–topic and document–topic distributions for all the interests, and then JNCTR further optimizes those distributions. Finally, the ranking combiner merges the ranked documents depending on the user–interest alignment estimated from the first stage.

3.2 Methods

Our recommendation methodology takes full advantage of the different sources of evidence that influence recommendations of items in a generic context. It is a three-stage methodology comprising of:

1) a topic model (IATM), 2) topic regression (JNCTR), and 3) ranking combination, as shown in Figure 3.1. We first give a short overview of the first two key models before describing how we instantiate them for the health forum recommendation task. We then describe the three stages in technical detail, and finally discuss our instantiation of the model for health forum thread recommendation to create a condition-aware topic model.

Method Overview. Our proposed interest-aware topic model is a generic topic model that can be used in many recommendation scenarios involving users. Without loss of generality, IATM assumes that *users* interact with *documents* (or *items*, as in the literature). The interactions generate some textual evidence that ties users and documents together — such as contributing a post within a larger, multi-user thread (document), commenting or authoring the entire document, such as one's own user profile. Like the standard topic model, documents are modeled as mixtures of *topics*; however, a key distinction in IATM is that it

assumes that topics are related to certain higher-level *interests* in a generative relationship. IATM captures explicit expressions of user *interests*, but crucially maintains the observations of these interests only as partially observed. This distinction allows the IATM to infer other interests of the user that are suggested by the contextual evidence of the user's other interactions.

JNCTR advances this step further, taking in the output of IATM's user–topic and document–topic distributions and further accounting for user–document interactions. As IATM already accounts for interests, we can instantiate standard collaborative topic regression for each interest separately, and jointly normalize them to output refined user–topic and document–topic distributions. These are then fused to generate recommendations.

Instantiating the Model for Health Forum Recommendations. IATM+JNCTR can be applied to various Web 2.0 contexts — recommendation tasks such as ones for movies, products, and discussion forums. These contexts all have document–user interactions, where user interests are partially observed through forum subscriptions or folksonomy tags, among other means.

For clarity, we now instantiate IATM+JNCTR for the health forum thread recommendation problem. In our scenario, users express their interests by subscribing to forums at health websites, which are largely specific to a medical condition. As in the general case, we do not expect users to necessarily subscribe to all the condition-specific forums that are relevant to them; we model such subscriptions as being partially observed.

The goal of our recommendation system is to recommend relevant health forum threads to users. Users can participate in forum threads by contributing posts, which forms the user–document interactions in our IATM+JNCTR framework. A user can report her clinical conditions as part of her user profile's free text description (*e.g.*, "About Me"). Such user documents are only used as evidence during training; to be clear, we do not recommend user profiles. Finally, individual threads on a particular condition discuss different symptoms and treatments in differing proportions. We assume that users are interested in certain symptoms that they experience, and treatments that they are undergoing.

In our health forum thread recommendation, we equate the following IATM terms with

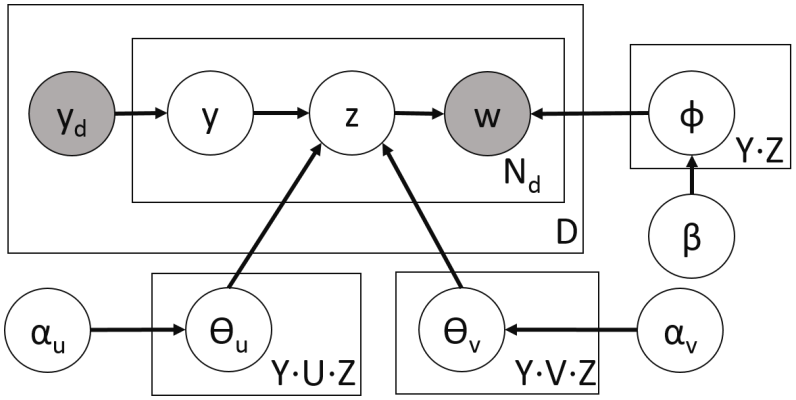


Figure 3.2: Plate notation for our interest-aware topic model (IATM). We observe the document words as well as partially observe the interests that select the topics for the words in the document (gray nodes). Topics are dependent on the interest (y), user (θ_u), and thread document (θ_v).

ones specific to our scenario: interest $\rightarrow$ condition; topic $\rightarrow$ (symptom, treatment); and document $\rightarrow$ thread.

3.2.1 Interest-Aware Topic Model (IATM)

We use the standard plate notation for the graphical model as shown in Figure 3.2. There are U users and V thread documents. Since each user has a user document (*i.e.,* a user profile), there are U user documents; hence we have altogether $D = U + V$ documents. Y denotes the set of all possible interests. In Figure 3.2, an interest y is sampled from a uniform distribution from the set of interests $y_d \subset Y$, where y_d is the union of all interests reported by the users participating in document d. Each interest y has Z latent topics which denote the fine-grained sub-topics of an interest (*e.g.,* in our instantiated IATM for medical conditions, they would be different symptoms or medications for a condition).

For each word in a document, a latent topic z is sampled from an interest y according to the topic distribution of the user θ_u, as well as the topic distribution of the thread θ_v. The reason behind this approach is intuitive: when a user contributes to a thread document, the topic of the user's words are dependent on the overall thread topic as well as the user's own set of interests. However, in the case of a user document (*i.e.,* the user profile), the choice of topic is only dependent on the user's own interests. A topic z is sampled only from the

interest y and θ_u, for such user documents.

A word w is sampled from z and the word–topic distribution ϕ. We invoke blocked Gibbs sampling as the exact inference of the full posterior is intractable. The inference process is similar to the author–topic model [81]; but in IATM, the author of a word is observed. We have two sets of latent variables, z and y. We draw each (z, y) pair as a block, conditioned on all other variables:

$$P(z_x = h, y_x = k | w_x = m, \boldsymbol{z_{-x}}, \boldsymbol{y_{-x}}, \boldsymbol{w_{-x}}, \boldsymbol{y_d}) \propto$$
$$(q_1 \frac{n^i_{hk} + \alpha_u}{\sum_z n^i_{zk} + Z\alpha_u} + q_2 \frac{n^d_{hk} + \alpha_v}{\sum_z n^d_{zk} + Z\alpha_v}) * \frac{n^k_{mh} + \beta}{\sum_w n^k_{wh} + W\beta}, \tag{3.1}$$

where $z_x = h$ and $y_x = k$ denote that the x^{th} word in d^{th} document is assigned to topic h under interest k; $w_x = m$ represents that x^{th} word is the m^{th} word in the vocabulary; $\boldsymbol{z_{-x}}$ and $\boldsymbol{y_{-x}}$ represent all topic and interest assignments not including the x^{th} word; n^i_{hk} is the number of times topic h is assigned with interest k for user i, not including the instance under consideration; and W is the total number of unique words in the vocabulary. Similarly, n^d_{hk} represents the number of times topic h has appeared under interest k in the d^{th} document; and n^k_{mh} denotes the number of times the m^{th} word in the vocabulary has appeared in topic h under interest k — excluding the current instances in all the cases. The three factors in Equation (3.1) represent the random variables θ_u (probability of topic given interest and user), θ_v (probability of topic given interest and thread), and ϕ (probability of a word given interest and topic). The Dirichlet priors for these three distributions are α_u, α_v, and β, respectively. We use a Dirichlet mixture of the two individual Dirichlet densities (θ_u, θ_v) as the prior [91, 76], giving equal weights to the mixture coefficients (*i.e.*, $q_1 = q_2 = 0.5$). We also learn the user–interest distribution γ. These distributions are estimated from the samples using the following equations:

$$\theta^{hik}_u = \frac{n^i_{hk} + \alpha_u}{\sum_z n^i_{zk} + Z\alpha_u}, \theta^{hdk}_v = \frac{n^d_{hk} + \alpha_v}{\sum_z n^d_{zk} + Z\alpha_v}, \tag{3.2}$$

$$\phi^{mhk} = \frac{n^k_{mh} + \beta}{\sum_w n^k_{wh} + W\beta}, \tag{3.3}$$

$$\gamma^{ik} = \frac{n^i_k}{\sum_y n^i_y}, \tag{3.4}$$

where n^i_k is the number of times interest k is sampled for user i.

Once the distributions are learned, we create sub-spaces of the entire user–thread interaction matrix based on each interest. The interaction matrix R^k for interest k is defined by:

$$r^k_{ij} = \begin{cases} 1 & \text{if } R_{ij} = 1, k \in Y_i \\ 0 & \text{otherwise,} \end{cases}$$

where Y_i is the set of interests for user i, $R_{ij} = 1$ if user i participated in thread j; 0 otherwise. Similarly, we define the user–, thread–, and word–topic distributions for this sub-space as θ^k_u, θ^k_v, and ϕ^k, respectively:

$$\theta^k_u = \theta_u\{k\}, \theta^k_v = \theta_v\{k\}, \phi^k = \phi\{k\}.$$

As an example, given the three threads in Table 3.1, IATM places the left one in the *ALS* sub-space, and the right ones inside the *Parkinson's disease* sub-space.

3.2.2 Joint Normalized Collaborative Topic Regression (JNCTR)

We treat each of the resultant interest-specific user–thread sub-spaces originating from IATM as a separate problem instance and optimize them individually using JNCTR as shown in Figure 3.1.

Figure 3.3 shows the plate model for each individual instance of JNCTR. Here, we use the notations θ_u, θ_v, ϕ, and R without the interest-specific superscript k. I and J denote the set of users and threads within this sub-space, respectively. Note that we omit the plate for word generation as we do not assume any particular generative process for them, and CTR does not re-sample topics once θ_v is obtained from the topic model as discussed in [101]. As in CTR, we introduce a latent variable ϵ^i_u that offsets the topic proportions θ^i_u for

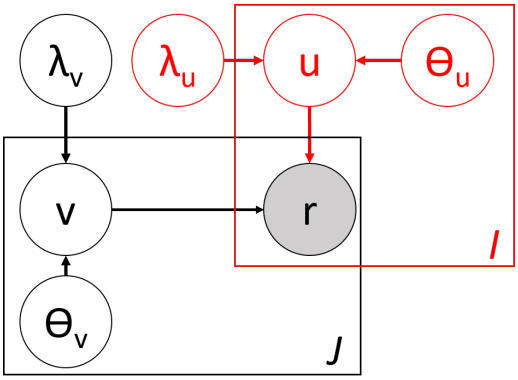


Figure 3.3: Plate notation for jointly normalized collaborative topic regression (JNCTR). Components in black are from collaborative topic regression (CTR [101]). Components in red are introduced for user modeling. Note that both plates for user and thread are form-identical.

i^{th} user when modeling the user's ratings. JNCTR assumes that there are Z topics both in user content and thread content $\beta = \beta_{1:Z}$. The generative process of JNCTR consists of the following steps:

(Step 1) For each user i, draw user latent offset $\epsilon_u^i \sim \mathcal{N}(0, \lambda_u^{-1}\mathbb{I}_Z)$ and set the user latent vector as: $u_i = \epsilon_u^i + \theta_u^i$,

(Step 2) For each thread j, draw thread latent offset

$\epsilon_v^j \sim \mathcal{N}(0, \lambda_v^{-1}\mathbb{I}_Z)$ and set the thread latent vector as: $v_j = \epsilon_v^j + \theta_v^j$,

(Step 3) For each user–thread pair (i, j), draw the rating as:

$r_{ij} \sim \mathcal{N}(u_i^T v_j, c_{ij}^{-1})$.

where $\mathbb{I}_Z$ is Z-dimensional identity matrix; λ_u and λ_v are the regularization parameters; c_{ij} is the precision parameter for r_{ij}, a confidence parameter for rating r_{ij}, where larger values denote higher trustworthiness. This is important in the case of implicit feedback-based systems like ours (note that $r_{ij} = 0$ denotes either that the i^{th} user is not interested in the j^{th} thread or the user is unaware of it). We set $c_{ij} = a$, if $r_{ij} = 1$, otherwise we set it to b, where a and b are tuning parameters satisfying $a > b > 0$. We discuss parameter tuning in Section 3.3.

Learning the Parameters for JNCTR. Given topic parameter β, computing the full posterior of u_i, v_j, θ_u, θ_v is intractable. We need to develop an EM-style algorithm to learn these parameters. Extending the posterior mentioned in [101], given λ_u, λ_v, and β, the complete log likelihood $\mathcal{L}$ of U, V, $\theta_u^{1:I}$, $\theta_v^{1:J}$, and R is defined as follows:

$$\begin{aligned}\mathcal{L} = &-\frac{\lambda_u}{2}\sum_i (u_i - \theta_{u^i})^T(u_i - \theta_{u^i}) - \frac{\lambda_v}{2}\sum_j (v_j - \theta_{v^j})^T(v_j - \theta_{v^j}) \\ &+ \sum_i \sum_m \log(\sum_k \theta_{u^{ik}} \beta_{k,w_{im}}) + \sum_j \sum_n \log(\sum_k \theta_{v^{jk}} \beta_{k,w_{jn}}) \\ &- \sum_{i,j} \frac{c_{i,j}}{2}(r_{ij} - u_i^T v_j)^2. \quad (3.5)\end{aligned}$$

We optimize this likelihood function by coordinate ascent, optimizing the CF variables u_i, v_j iteratively. To update u_i and v_j, we take the gradient of $\mathcal{L}$ with respect to u_i and v_j and set it to zero. This yields:

$$u_i \leftarrow (VC_iV^T + \lambda_u I_K)^{-1}(VC_iR_i + \lambda_u \theta_u^i R_i), \tag{3.6}$$

$$v_j \leftarrow (UC_jU^T + \lambda_v I_K)^{-1}(UC_jR_j + \lambda_v \theta_v^j R_j), \tag{3.7}$$

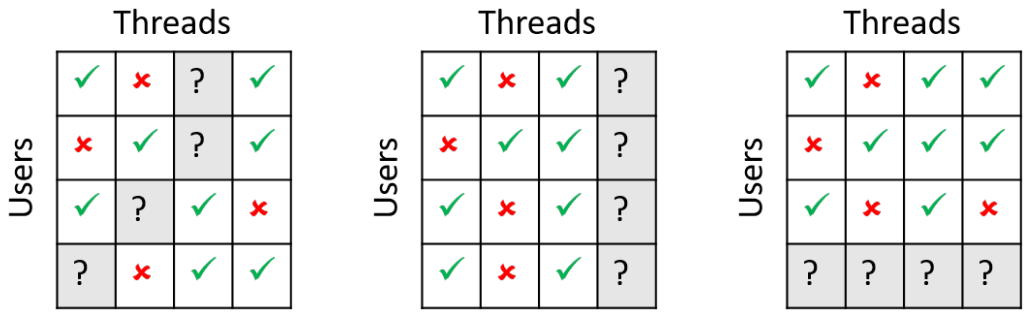
where $U = (u_i)_{i=1}^I$, $V = (v_j)_{j=1}^J$, C_i is a diagonal matrix with c_{ij} $(j = 1, \ldots, J)$ as its diagonal elements and $R_i = (r_{ij})_{j=1}^J$ for user i. C_j and R_j are similarly defined for thread j.

Prediction. Once the locally optimal parameters $U^*, V^*, \theta_u^*, \theta_v^*$ are learned, JNCTR can predict ratings. Given that D is the observed data, the prediction is estimated as:

$$\mathbb{E}[r_{ij}|D] \approx (\mathbb{E}[\theta_{U_i}|D] + \mathbb{E}[\epsilon_{u_i}|D])^T \cdot (\mathbb{E}[\theta_{V_j}|D] + \mathbb{E}[\epsilon_{v_j}|D]). \tag{3.8}$$

As shown in Figure 3.4, we address the prediction of ratings as follows:

(Fig. 3.4a) In-matrix prediction: All the users and threads have at least one interaction. We use



(a) In-matrix (b) Out-of-matrix (for thread) (c) Out-of-matrix (for user)

Figure 3.4: Illustration of three prediction tasks for our thread recommendation system. "✓," "× ," and "?" denote "like", "dislike", and "unknown" respectively.

the point estimate of $\theta_{u^i}, \theta_{v^j}, \epsilon_u^i$, and ϵ_v^j to approximate the expectations:

$$r_{ij}^* \approx (\theta_u^{i*} + \epsilon_u^{i*})^T (\theta_v^{j*} + \epsilon_v^{j*}) = (u_i^*)^T v_j^*. \tag{3.9}$$

(Fig. 3.4b) Out-of-matrix prediction (for a *thread*): Some threads do not have interaction history; *i.e.*,

$$r_{ij}^* \approx (\theta_u^{i*} + \epsilon_u^{i*})^T (\theta_v^{j*}) = (u_i^*)^T \theta_v^{j*}. \tag{3.10}$$

(Fig. 3.4c) Out-of-matrix prediction (for a *user*): Some users do not have interaction history; *i.e.*,

$$r_{ij}^* \approx (\theta_u^{i*})^T (\theta_v^{j*} + \epsilon_v^{j*}) = (\theta_u^{i*})^T v_j^*, \tag{3.11}$$

substituting u_i and v_j from Steps 1 and 2 in the JNCTR generative process. We thus obtain a ranked list of interest-specific threads that is recommended to a user.

Efficiency. Note that in Figure 3.1, multiple instances of JNCTR are run, but each instance is run on a partition of the full matrix. The computational complexity of JNCTR is comparable to the original CTR algorithm; the number of updates in both are identical.

3.2.3 Fusing the Final Ranked List

Once we obtain the condition–specific optimized user- and thread-topic distributions, we combine them into a single ranking using the γ distribution defined by Equation (3.4). We

explore three methods to fuse the individual, interest-specific lists:

1. Proportional Selection. For every interest that user i is interested in, we prepare a list of threads (in descending order of predicted score) that the user might find interesting. We select the top-M threads from each condition sub-space according to user i's γ distribution. For illustration, say John has a γ distribution of $\{\text{Multiple Sclerosis} : 0.8, \text{Asthma} : 0.2\}$. Then when recommending threads to John, 80% are chosen from the top threads in Multiple Sclerosis, and the remaining 20% come from Asthma.

2. Combined Score-Based Selection. Here, we obtain a single ordered list of threads from their combined score for each user. First, we normalize scores in each interest sub-space between $[0, 1]$. For user i, the combined score of a thread j is defined by Equation 3.12 which ranks the items in descending order of their total score:

$$R^*_{ij} = \sum_y \gamma^{iy} \times r^{ij}_y, (y \in Y_i). \tag{3.12}$$

3. Maximum Split Preference-Based Selection. This is similar to the binary preference based merging in [112]. For each user i, we only consider the condition with the highest preference γ score.

We note that although the proportional selection strategy makes the recommended list of threads more diverse in nature, the Combined Score-Based Selection obtains superior results on our datasets.

3.3 Experiments

To answer important questions about our model, we consider specific experimental settings. In the following, we describe the main results of our study after detailing the datasets, evaluation metrics, and baselines.

Dataset. We constructed following large real-world consumer health forum dataset to vali-

Dataset	# Users	# Threads	# Posts	Avg P:T	# Distinct Conditions	Avg C:U
HBD	127,903	155,863	716,744	4.6	235	4.01

Table 3.2: Statistics on our health forum dataset. "Avg P:T" and "Avg C:U" denote the average number of posts in a thread and conditions reported by a user, respectively.

date our model:

HealthBoards (HBD)[2]: We use the publicly available HealthBoards dataset[4]. We collate all the posts made by a user and treat them as the user document. We ignore generic categories of threads such as 'Family', 'Support', 'Healthcare' and 'General'.

We remove all stop words and select the top 8,000 words based on TF-IDF scores. The TF-IDF computation was done only on the training data for all the experiments. Similar to other recommendation works, we remove users with few interactions, namely, with less than three thread interactions. Table 3.2 shows some statistics on our datasets and their user reported conditions.

Metrics. Similar to [101], we do not rely on precision, as our ground truth is only implicit feedback. Samples with negative values could be threads that the user had not seen (but would have been interested in), as well as those where the user explicitly did not interact with. As such, we use three metrics to assess recommendation quality:

Recall@M considers how many top-M threads were actually interacted by the user (higher is better). The recall for the entire system can be summarized as the average recall value for all users.

Mean Reciprocal Rank (MRR) indicates where in the ranking the first relevant thread is returned by the system, averaged over all users. This measures the ability of the system to return a relevant thread at the top of the ranking. Let r_i be the rank of the highest ranking relevant thread for a target user i, then MRR is just the reciprocal rank, averaged over all target users, N_U:

$$MRR = \frac{1}{N_U} \sum_{i=1}^{N_U} \frac{1}{r_i}.$$

[4]http://resources.mpi-inf.mpg.de/impact/peopleondrugs

Normalized Discounted Cumulative Gain (nDCG) is well suited for evaluation of recommendation system, as it rewards relevant threads in the top ranked results more heavily than those ranked lower. nDCG is computed as:

$$nDCG_i = Z_i \sum_{j=1}^{M} \frac{2^{r(j)} - 1}{log(1+j)},$$

where Z_i is a normalization constant calculated so that a perfect ordering would obtain nDCG of 1; and each $r(j)$ is an integer relevance level (for our case, $r(j) = 1$ and $r(j) = 0$ for relevant and irrelevant recommendations, respectively) of result returned at the rank j ($j = 1, \cdots, M$). Then, $nDCG_i$ is averaged over all our target users. in this work, we use $nDCG@M$ $(M = 5, 10)$ for evaluation where M is the number of top-M threads recommended by our approaches.

Baselines. We compare our instantiated IATM+JNCTR with six baselines. Our complete model uses four signals overall: the user–thread interaction history, textual content of threads, user profiles, and the user–reported conditions. We chose baselines for their competitiveness, recency, and use of particular signals common to our model. Comparing among the various models can also be seen as assessing how important each form of evidence is in achieving quality recommendation. Table 3.3 summarizes how the baselines account for some subsets of the evidence in our model.

1. ***Collaborative Filtering (CF)***: This is the non-negative matrix factorization-based (NMF) method for collaborative filtering of [50].

2. The ***Author-Topic Model (AT)*** learns the author–topic distributions [81]. We estimate the thread–topic distributions from the learned word–topic distributions. These can be interpreted as the user and thread latent topic matrices for our task, respectively. We empirically set the hyperparameters $\alpha = 0.1$, $\beta = 0.01$, and the number of iterations and latent topics to 2,000 and 50, respectively.

3. ***Collaborative Topic Regression (CTR)*** [101] is the basis for JNCTR, but without the

Method	User-Thread Interaction	User Docs	Thread Docs	User Reported Conditions
1. CF (NMF) [50]	✓			
2. AT [81]		✓	✓	
3. CTR [101]	✓		✓	
4. IATM		✓	✓	✓
5. CAR [79]	✓			✓
6. AT + JNCTR	✓	✓	✓	
7. IATM + JNCTR	✓	✓	✓	✓

Table 3.3: Signals considered by comparative methods.

individual interest- (condition-) specific instances, and with LDA (instead of IATM) as the input. We tune the parameter settings, $b = 0.01$, $\lambda_u = 0.01$, $\lambda_v = 0.1$ to yield its optimal results.

4. ***IATM alone*** can also generate recommendations. Unlike the configuration described in Section 3.2.1, we consider the entire user- and thread-topic matrices obtained from IATM, and use them for the recommendation task. We empirically set the hyperparameters β=0.1, and α_u=α_v=5, and set the number of topics for each medical condition to 3 and the number of iterations to 2,000.

5. ***Context Aware Recommendation (CAR)*** uses Factorization Machines for the recommendation [79]. We use the libFM package (http://www.libfm.org). We create the test set following the sampling policy described in [65]. This models user–reported conditions as the context for each interaction.

6. ***AT + JNCTR***: In this experiment, we replace the first stage of our pipeline with Author–Topic model (AT) [81]. This model directly contrasts with our full model to see the comparative difference when using IATM over AT.

For our IATM+JNCTR model, we obtain the optimized matrices from the second stage of the pipeline and obtain the final prediction after combining the ratings as described in Section 3.2.3. We keep the optimal settings for IATM. For JNCTR, we empirically optimize the hyperparameters, $\lambda_u = 0.01$, $\lambda_v = 0.1$, and $b = 0.1$ which are estimated from grid search.

Method	MRR	nDCG	
		@5	@10
1. CF (NMF) [50]	0.179	0.180	0.194
2. AT [81]	0.023	0.033	0.036
3. CTR [101]	0.186	0.178	0.193
4. IATM	0.113	0.059	0.064
5. CAR [79]	0.092	0.081	0.087
6. AT+JNCTR	0.213*	0.221*	0.254*
7. **IATM+JNCTR**	**0.327***	**0.329***	**0.361***

Table 3.4: MRR and nDCG scores obtained by in-matrix prediction. "*" denotes the difference between the best baseline ("3. CTR") and our methods ("6. AT+JNCTR") and ("7. IATM+JNCTR") is significant for $p < 0.005$.

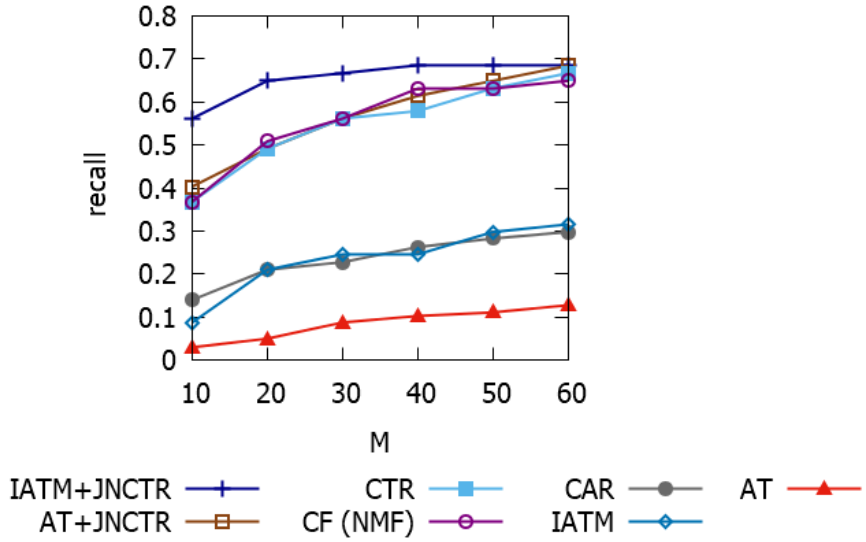


Figure 3.5: Recall scores at various M top ranks for the HBD dataset.

In-Matrix Setting: We report results using 5-fold cross validation. We split users with 5+ threads into a training set (80%) and a test set (the remaining 20%). Users with fewer than five interactions always appear in the training set. For these main results, we use a *warm-start* (*i.e.,* in-matrix) setting, ensuring that each user or thread in the test set has been observed at least once in the training set.

Temporal Setting: Apart from the 5-fold cross validation, where a randomly selected 20% split is used as test, we also report results for a temporal experiment. In this setting, for each user, the last 20% of her interacted threads are kept for test. Similar to the in-matrix setting, we make sure that all the user and threads appear at least once in the training data.

Results. Figure 3.5 shows the recall@M ($M = 10, 20, \ldots, 60$) for comparative methods for the in-matrix setting. In both datasets, the IATM+JNCTR pairing achieves the highest recall. CTR and CF (NMF) give comparable performance, which is consistent with [101]. We observe that, in the HBD dataset, while the recall scores converge when $M \geq 60$, our IATM+JNCTR method outperforms the others with a significant margin for lower values of M (*i.e.,* more important ranks). This indicates that our pipeline can rank relevant items high in the recommendation list. This phenomena is also depicted in Table 3.4 where we present the MRR, nDCG@$\{5, 10\}$ scores for all methods. The AT model alone works poorly in both datasets, indicating that it is insufficient to consider only user and thread documents. However, when paired with JNCTR (namely, AT+JNCTR), AT significantly improves recommendation accuracy, which factors in the user–item interaction. With IATM+JNCTR performing best, we conclude that considering the user documents and the user reported interests enhances the user–item interaction history for best recommendation accuracy. We perform a paired t-test to verify whether the obtained results are statistically significant or not. As shown in Table 3.4, we observe that, in the HBD dataset, our full IATM+JNCTR outperforms all the other baselines in both MRR and nDCG@$\{5, 10\}$.

Table 3.5 shows the recall@60, MRR, and nDCG@$\{5, 10\}$ scores in the temporal setting. We find a similar trend in recall scores at various top ranks compared to the in-matrix setting. We observe that IATM+JNCTR outperforms the others with statistical significance in this setting. Unlike the in-matrix setting, AT+JNCTR achieves statistically significant improvement only in the HBD dataset in MRR and recall@60. This indicates the robustness of our model in both of the randomized 5-fold and temporal settings.

3.4 Discussion

Aside from the main in-matrix results, there are several important research questions that merit deeper investigation. As shown in Figure 3.4, our IATM+JNCTR pipeline handles cold start by incorporating prior knowledge. Our research questions (RQ) are:

Method	Recall	MRR	nDCG	
	@60		@5	@10
1. CF (NMF) [50]	0.483	0.166	0.136	0.160
2. AT [81]	0.114	0.021	0.027	0.032
3. CTR [81]	0.517	0.211	0.178	0.205
4. IATM	0.286	0.142	0.055	0.064
5. CAR [79]	0.348	0.183	0.132	0.152
6. AT+JNCTR	0.549*	0.256*	0.176	0.202
7. IATM+JNCTR	**0.674***	**0.340***	**0.289***	**0.318***

Table 3.5: Recall@60, MRR, and nDCG@$\{5, 10\}$ scores for temporal prediction. "*" denotes the difference between the best baseline (Row 3) and our methods (Rows 6–7) are significant for $p < 0.005$.

RQ1: How does it perform with cold-start *documents* (*i.e.,* newly-introduced threads)?

RQ2: How well can the IATM+JNCTR pairing explain its recommendations?

RQ3: How well does the IATM+JNCTR pairing capture users' interests for specific symptoms and treatments?

RQ4: Does it actually recover the users' implicit interest in specific conditions?

In the following, we answer each of these RQs.

RQ1: Out-of-matrix Thread Recommendation. It is important for a newly-posted thread (usually some form of question) to receive quality answers. To simulate this, we partition all threads evenly among five folds. For each fold, we form a submatrix from the threads which are not within this fold and the corresponding users. We treat this submatrix as training data and learn user–topic and thread-topic distributions from the same. We ensure that none of the in-fold threads occurs in the training data. In the test phase, for all the in-fold threads, we consider the textual content of the query and user profile (*i.e.,* "About Me" text and user-reported conditions) of the user to estimate the topic distributions from the model learned during training. The task is to correctly predict the set of in-fold threads each user will actually interact with.

Table 3.6 shows the MRR and nDCG@$\{5, 10\}$ obtained by the relevant comparative methods for this out-of-matrix thread recommendation task. IATM+JNCTR again achieves

Method	MRR	nDCG	
		@5	@10
2. AT [81]	0.025	0.029	0.036
3. CTR [101]	0.131	0.098	0.114
4. IATM	0.094	0.068	0.081
6. AT+JNCTR	0.164*	0.146*	0.172*
7. **IATM+JNCTR**	**0.221***	**0.234***	**0.263***

Table 3.6: MRR and nDCG scores obtained for out-of-matrix (thread) recommendation. "*" denotes statistical signficance between the best baseline (Row 3) and our methods (Rows 6 and 7) at $p < 0.005$. CF and CAR do not work in this setting.

the best scores. In general, performance degrades compared to the in-matrix setting shown in Table 3.4, due to the harder nature of the task. It is interesting that prior evidence in the form of user profiles and the user-reported conditions significantly help, in the absence of observed user–item interactions. Examining the relative performance of IATM, CTR, and AT+JNCTR, we argue that the user-reported conditions further improve recommendation accuracy, compared against the evidence from user profiles.

RQ2: Transparency of Recommended Threads. While CF-based recommendation algorithms work well in terms of prediction accuracy, their latent factors make it difficult to justify the recommendation to the user [116, 29]. User participation (measured through metrics like clickthrough rate) alleviates this difficulty if items are recommended to a user with semantic explanation. Our IATM+JNCTR adds transparency by providing users with the context when recommending a thread. We learn the user–condition distribution (γ) in the first phase of our model, which is used downstream to combine recommended threads from different condition-specific sub-spaces. While presenting the combined list of threads to the user, the recommendation system can disclose which sub-space a particular thread belongs to. When a thread exists in multiple sub-spaces, we can select condition $c = \text{argmax}_c\, \gamma_{ic}$ and present it as the context recommending it to a user i, as "recommended due to your interests in c". Table 3.7 presents sample threads recommended by our pipeline. Note that it can identify the context (*i.e.*, the condition) for recommending the threads.

RQ3: Significance of Discovered Symptoms and Treatments. A challenge in achiev-

Recommended Thread	Candidate Conditions
I have been suffering from lupus lately having red dots all over my face. Anyone else having the same?	**1. Systemic Lupus Erythematosus** 2. Spinal Stenosis
Suffering from degenerative spine, spinal stenosis, severe scoliosis..	**1. Spinal Stenosis** 2. Systemic Lupus Erythematosus
Is there anyone with ALS who doesn't catch colds anymore?	**1. ALS** 2. Dysautonomia
has anyone stopped rytary and gone back to stalevo or something like that?	**1. Parkinson's Disease** 2. Vitamin B12 Deficiency

Table 3.7: Recommended threads for sample users. The explaining condition chosen by IATM+JNCTR is bolded.

ing quality recommendation is to appropriately learn the topics even when overlapping words appear among several conditions. Our IATM+JNCTR leverages the user-reported conditions and learns the appropriate word distribution. Table 3.8 shows the top words discovered by our model for several conditions. Note that, while there are few common words across conditions, — *ALS*, *Epilepsy*, and *Multiple Sclerosis* all list *brain* among top keywords — our method can distinguish among these conditions.

It is important to analyze the condition-specific topics learned by our approach. Since different users express various levels of interest towards particular aspects (symptoms or treatments) of a condition, it is necessary to capture these aspects to achieve quality recommendation. Table 3.9 presents some condition-specific topics discovered by our pipeline. From simple observation, one can see correspondences for *Diabetes*, Topic 0 lists affected body parts and associated difficulties, Topic 1 discusses diets, and Topic 2 relates to human physiology — having words such as *blood, insulin*. In the scenario where a user has *Diabetes* and is interested in managing the condition through her diet, our model can recommend threads that would match her interests at this topical level.

In the case of serious terminal diseases, such as *cancer*, psychological and spiritual words, such as *god* and *luck*, appear in the top words as topics. Consider the following posts by cancer patients:

"*Would love to talk to anyone with ovarian cancer..really believe faith can play a huge role in recov-*

Eye & Vision	ALS	Parkinson's Disease	Diabetes	Cancer	Epilepsy	Multiple Sclerosis
eye	als	neurologist	carb	cancer	seizure	copaxone
vision	reflexes	pd	sugar	chemo	seizures	lesions
drops	muscle	nervous	insulin	radiation	keppra	mri
cataracts	amyotrophic	tremors	glucose	cells	hope	immune
red	nervous	scan	levels	kidney	meds	brain
reduce	irregular	shaking	eat	scan	brain	help
laser	brain	facial	diet	prayers	care	celebrex
opthalmologist	feel	control	exercise	god	pain	scoliosis
omeprazole	weight	tissue	test	luck	alcohol	breathing

Table 3.8: Example of the top words for certain medical conditions learned by our IATM+JNCTR model.

ering and also positive attitude..I wish this disease didn't exist"

"*Sending all my positive energy to you..Feel free to reach out if you need anything*".

Users with similar conditions often participate in health forums for such emotional support rather than informational need [83, 68, 109]. Our model can capture this phenomenon as a topic for certain conditions.

Topic 1	Topic 2	Topic 3
legs	carbs	blood
shake	sugar	insulin
feet	eat	high
walking	diet	glucose

(a) Diabetes

Topic 1	Topic 2	Topic 3
neurologist	feel	shaking
brain	help	cold
mri	hope	tension
disorders	people	dizziness

(b) Parkinson's Disease

Topic 1	Topic 2	Topic 3
leg	pain	disc
heart	feel	cervical
muscles	issues	spine
body	help	brain

(c) ALS

Topic 1	Topic 2	Topic 3
brain	warm	instruction
feeling	burning	breathing
painful	lesions	remembering
walker	harder	recall

(d) Multiple Sclerosis

Topic 1	Topic 2	Topic 3
cancer	treatment	prayer
lump	chemo	god
lymph	radiation	afraid
growth	stage	doctor

(e) Cancer

Table 3.9: Example of condition-specific topics (*i.e.,* symptoms and treatments) discovered by our IATM+JNCTR model.

# held-out conditions	Perfect recall
1	0.64
2	0.45
3	0.39

Table 3.10: Unreported conditions recovered by the IATM. Perfect recall denotes to the fraction of cases where it can recover all the held-out conditions.

RQ4: Predicting Implicit Conditions. In IATM, recall that we sample both interest (condition) and topic (symptom or treatment) for each word as described in Section 3.2.1. As a result, along with word- and thread-topic distribution, the model also learns the user–condition distribution γ. Although it is used later on for recommendation in our pipeline, it can also serve to predict implicit conditions. For an example culled from our dataset, a user reports *Multiple Sclerosis* as a condition he is afflicted with in his profile. However, from all of the posts that he interacts with, our model estimates the γ distribution to be $\{\text{Multiple Sclerosis} : 0.8, \text{Asthma} : 0.2\}$. In this case, the unreported, implicit condition "Asthma" is predicted by our model. We argue that this is a desirable nature of our model.

To quantitatively evaluate the capability of our model to predict the missing condition, in a separate experiment, we omit 1 to 3 conditions for each user for 1/5 of the users during training. We train our model and obtain the γ distribution for all users. We then evaluate how many cases our model can recover all of the missing conditions, *i.e.,* whether it achieves perfect recall. Table 3.10 reports our findings, indicating that our model can correctly predict over 60% of the cases in the single missing conditions. Unsurprisingly, performance degrades as the number of missing condition increases. However, gradually, even in the three missing conditions, our model can predict 39% of the cases.

3.5 Conclusion

We have systematically investigated how to best utilize each user's participation in online discussion forums to recommend relevant threads. Our IATM+JNCTR model leverages the user-reported clinical conditions to distinguish lexically similar yet different threads, addi-

tionally accounting for each user's specific, latent preferences for particular treatments and symptoms. In our experiments on warm- and cold-start scenarios, involving both users and threads, our framework demonstrated significant improvements over the current state-of-the-art methods. Deeper analysis reveals that IATM+JNCTR's modeling of latent conditions and user profiles are key to achieve competitive performance.

As our framework is general and language independent, we believe that it could be useful in other domains, including community question answering and scholarly paper recommendation. We hope the research community will apply our model to other scenarios to validate its modeling capabilities.

When we look at the forum posts written by the users in a chronological order, we observe that their interests evolve over time [27]. Models of time-varying user preferences in the recommendation domain generally assume that users evolve according to a "global clock" [59, 48], whereas interests of users participating in discussion forums progress according to their own personal timeline. In the future, it would be interesting to study how to capture this evolving trend with Recurrent Neural Networks to improve the thread recommendations inspired from techniques such as language modelling.

Chapter 4

Cold Start Thread Recommendation

4.1 Introduction

Online discussion forums are continuously growing as new threads are created frequently. While this enables users to ask questions to a large community, ensuring that the members find questions relevant to their interests, is key to getting them answered. This is a challenging matching problem, due to the huge number of threads, and the large number of active members in the community. Traditional Recommendation systems can be helpful in bridging this gap by suggesting users with relevant interests and expertise for a discussion thread. However their modelling capabilities hit a bottleneck in case of completely new item.

In this chapter, we systematically study this problem and work towards building a system that recommends incoming threads (with no user interaction observed whatsoever) to relevant users for participation. Recommendation systems mostly use past interaction history of a user or item to solve the matching problem. Even though this strategy can model users, given the threads they responded to in the past, it will fail on new threads. They have no interaction history to facilitate predictions – this is a form of cold start. For such a thread, the system needs to use its textual content in order to find potentially interested users.

We view this cold start thread recommendation from a different perspective – as one of supervised eXtreme Multi-Label Classification (XMLC). XMLC has been applied for text classification in domains where a document can have multiple tags among several thou-

sands of possible tags (e.g., Wikipedia page categorization or product categorization in e-commerce). Recently deep learning approaches have been proposed for this area for better text understanding and handling the large label space efficiently [55].

We propose a novel neural network architecture for this recommendation task. Inspired by the success of Recurrent Neural Networks (RNNs) on a range of natural language processing tasks, we apply stacked bidirectional RNN for encoding the raw textual content of a post. We consider the multi-label prediction task as multiple, individual binary classifications where the correlation among labels (i.e., users) is exploited by the model.

We hypothesize that users can be subdivided into clusters in a latent space depending on their interests. Users belonging to the same cluster are likely to have similar preferences and vice versa. In the literature, we find similar observations in different contexts around recommendation systems [112]. Inspired from this, we introduce a novel, cluster-sensitive attention (CSA) mechanism. It allows a post text to be encoded differently for different clusters using cluster-specific attention weights. This lets the network focus on parts of the text that might be more important for the set of clustered users while predicting their participation interest. Assuming similarity of preferences among users, and learning text encoding per cluster (as opposed to every individual user), helps us in addressing the scalability of the extreme multi-label task by reducing the parameter space. Additionally, it also helps in alleviating the sparsity issue, as the limited amount of evidence per user could easily lead to overfitting in such complex model architecture, otherwise.

From our results over multiple datasets, we find that our CSA-based XMLC model outperforms standard content-based recommendation algorithms as well as state-of-the-art XMLC models significantly.

Our approach is best geared towards providing cold start recommendation for a discussion forum scenario. To the best of our knowledge, this is also the first attempt at solving the generic cold start recommendation problem from the extreme multi-label classification perspective. To summarize, our contributions are the following:

- We formulate the well-known cold start recommendation problem as an Extreme Multi-Label Classification task.

- We propose a neural architecture using a novel cluster sensitive attention mechanism to cater to the varying interests of users.
- We show the effectiveness and generalization ability of our approach through a set of carefully-designed experiments, over multiple datasets. Additionally, we validate our problem formulation by comparing our model with traditional recommendation algorithms.

4.2 Background

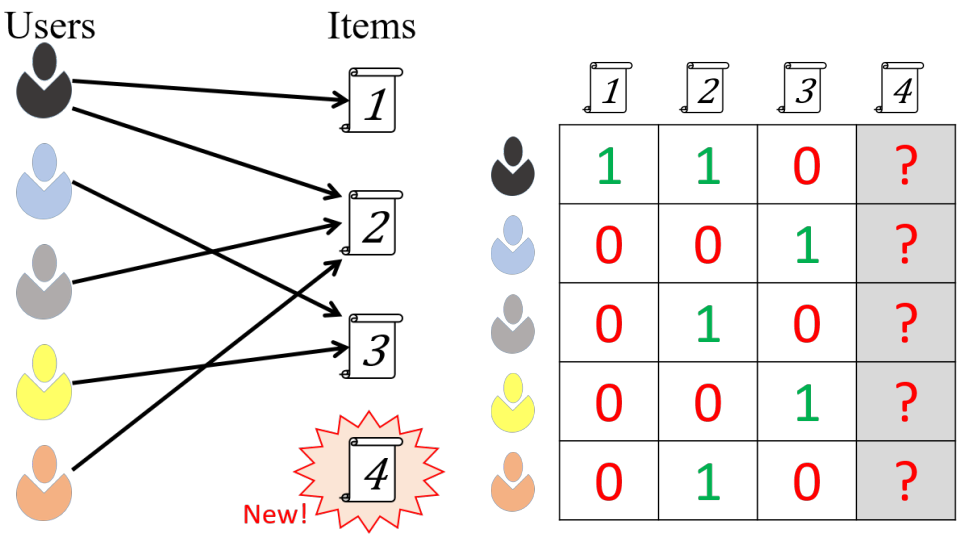
We have given a brief overview of the cold start problem faced by the traditional recommendation systems earlier in Section 2.2. Here, we elaborate the underlying technical difficulty to understand the challenge. We thereafter describe the approaches for extreme multi-label classification. We then conclude this section by connecting these two parts in a formal problem statement.

4.2.1 Cold Start Recommendation Problem

The two primary elements in a recommendation scenario are users and items. The user–item interaction forms a bipartite graph (Figure 4.1a) where a directed edge from a user to an item represents that the user has interacted in some way with the item (e.g 'like', 'comment', 'retweet' etc). The corresponding interaction matrix is shown in Figure 4.1b. In the widely used latent factor models, the user and item are represented in a low-dimensional (D) space - user is denoted by a latent vector $\mathbf{u}_i \in \mathbb{R}^D$, and item by $\mathbf{v}_j \in \mathbb{R}^D$. The prediction r_{ij} is formed by an inner product of these two vectors,

$$r_{ij} = \mathbf{u}_i^T \mathbf{v}_j$$

In non-negative matrix factorization (NMF) based approaches, the latent vectors are initialized randomly and can be learned using a regularized squared error loss in terms of $\mathbf{u}_i$ and $\mathbf{v}_j$, where $i \in \{1, \cdots, U\}$ and $j \in \{1, \cdots, V\}$; U, and V are the number of users and



(a) Interaction Graph. (b) Interaction Matrix.

Figure 4.1: Illustration of the cold start problem. (a) Edges represent user interactions. Item 4 has no interaction. (b) In interaction matrix: '1' ⇒ interaction, '0' ⇒ no interaction.

items respectively.

Let's consider the dynamics when a newly-created item '4' is introduced. In the interaction matrix, since the column for item '4' is entirely unobserved, it is also referred as *out-of-matrix* item recommendation [102, 100]. As no ground truth value of r_{ij} for $j = 4$ is available, the model will *not* be able to learn the correct representation of $\mathbf{v}_{j=4}$, giving rise to cold start problem. This is a significant limitation of an NMF based recommender system in a forum context where new threads are posted quite frequently needing user participation.

4.2.2 Extreme Multi-label Classification

Extreme multi-label classification (XMLC) refers to the task of assigning each item its most relevant subset of labels from an extremely large collection of class labels. The fundamental difference between multi-label classification and traditional binary or multi-class classification task is that in multi-class classification only one among the possible labels applies to an item, whereas in multi-label classification the labels can be correlated with each other or have a subsuming relationship, and multiple labels can apply for an item (e.g., 'politics' and 'White House' for news articles, 'electronics', 'Samsung' and 'smartphone' for products, 'Eiffel tower' and 'vacation 2017' for an image).

In this setting, an instance can be considered as a pair $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is the feature vector for an item, and $\mathbf{y}$ is the label vector i.e., $\mathbf{y} \in \{0,1\}^L$, L is the number of labels. Given n such training instances, a classifier is trained which can predict the label vector for an unseen test item. Since the label space L can be extremely large, it suffers from scalability, and sparsity issues. Properly exploiting the correlation among labels can help in alleviating them.

Problem Statement: We approach the cold-start thread recommendation problem from an XMLC task perspective, where given a *new* thread, using only its textual features we try to predict the set of interested users. We formalize the problem statement as,

Given a piece of text $t \in T$, find a mapping $f : T \rightarrow \{0,1\}^U$ where T is the set of all items. f would give us a probability score for each of the U labels given t,

$$f(t) = P(r_i = 1|t)$$

where $i \in \{1, \cdots, U\}$, and r_i is the label corresponding to i^{th} user.

4.3 Proposed Method

We propose a neural network architecture (Figure 4.2) to predict the subset of users interested in a new thread from the extremely large set of users in the forum community. As a newly-created thread has only a single post, we use the terms *thread* and *post* interchangeably.

4.3.1 Text Encoding

The network takes as input a post text p consisting of a sequence of words $(w_1, w_2, \ldots, w_n)$.

We first embed each word in a lower-dimensional space so that a post is now represented as a sequence of word vectors $\{\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_n}\}$ where $\mathbf{q_i} \in \mathbb{R}^d$. We initialize the word vectors using pre-trained GloVe embeddings [75] but tune it during training to capture domain

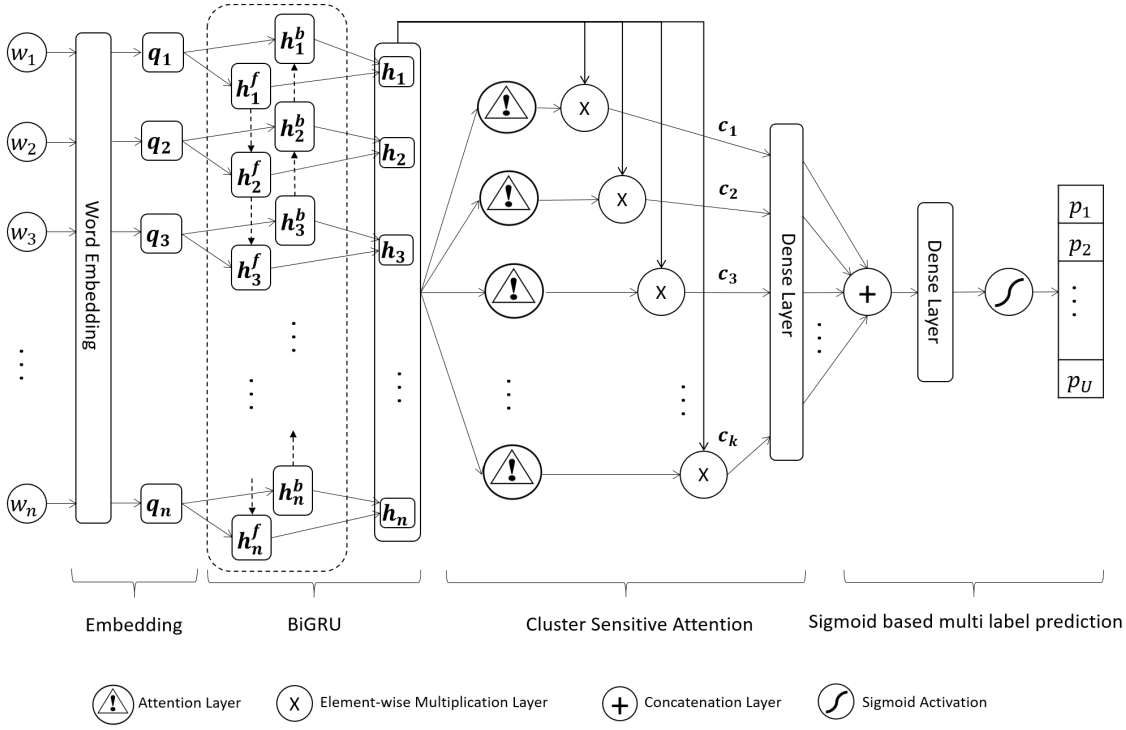


Figure 4.2: Overall model architecture of the XMLC-based recommendation system.

specific semantics.

The post is then encoded using bi-directional RNNs. The input to the bi-directional RNN is the embedded word sequence of a post $\{\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_n}\}$ and the output is a sequence of vectors $\mathbf{h}^p = \{\mathbf{h_1}, \mathbf{h_2}, \cdots, \mathbf{h_n}\}$ where $\mathbf{h_i} \in \mathbb{R}^g$ denotes the encoded representation of the post.

An RNN reads the sequence of word vectors $\{\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_n}\}$ from left to right in the forward pass and creates a sequence of hidden states $\{\mathbf{h_1^f}, \mathbf{h_2^f}, \cdots, \mathbf{h_n^f}\}$, where $\mathbf{h_i^f}$ is computed as:

$$\mathbf{h_i^f} = RNN(\mathbf{q_i}, \mathbf{h_{i-1}^f}) \tag{4.1}$$

where RNN is a function. Due to vanishing (and conversely, exploding) gradients, the basic RNN cannot learn long-distance temporal dependencies with gradient-based optimization [9]. To deal with this, extensions to the basic RNN have been proposed that incorporate a memory unit to remember long term dependencies. We use one such variant named Gated Recurrent Unit (GRU) [21] instead of the basic RNN in our model.

In the backward pass, a GRU reads the input sequence in reverse order and returns a sequence of hidden states $\{\mathbf{h_n^b}, \mathbf{h_{n-1}^b}, \cdots, \mathbf{h_1^b}\}$. The forward and backward hidden states are then concatenated to create the encoded hidden state of a word $\mathbf{h_i} = [\mathbf{h_i^f}; \mathbf{h_i^b}]$ considering all its surrounding words.

We use a stack of such bi-directional GRUs where the output of a GRU layer is fed as input to the GRU at next level. This increases the expressive power of the network by capturing higher-level feature interactions between different words. The output sequence from the final bi-directional GRU layer is the representation of the post text $\mathbf{h}^p$. In our experiments, we have used a stack of two bi-directional GRUs. We also experimented with adding more layers, but that did not lead to much improvements in our results.

4.3.2 Cluster Sensitive Attention

I have been recommended to undergo tracheotomy and put in a PEG. I am wondering how many days I'll have to stay in the hospital? Will I have a hard time adjusting afterwards? Does the hose need to be connected while transferring? Will the equipments take up a lot of room? How do you call for help? I am unable to talk or move. What type of tube would you suggest? I have been a member of the ALS community for some time now. It is nice to read the way some people think and face ALS, it gives me courage.

The above is an illustrative post (synthetically modified for anonymity) in an ALS forum. The patient is about to undergo a surgical procedure (tracheotomy) and has queries regarding the procedure, recovery time and the after effects. Furthermore, since the procedure creates a hole in the neck to provide an air passage to the windpipe, it disrupts the normal eating and speaking abilities of a person. The patient therefore has additional questions regarding the best feeding tubes and ways of communicating with others. Given its complexity and detailed information need, individual users are unlikely to be able to answer all parts of it. Instead, we envision that users with different backgrounds and experience could address specific parts; e.g., someone having experience with a PEG (Percutaneous Endoscopic Gastrostomy) could answer the queries regarding it, while someone else could

help clear the user's concerns regarding the procedure and recovery.

Put succinctly, different users may be interested in disparate parts of a (new) post.

This motivates us to build a component in our network that can help focus on parts of a post for different users. To achieve this, we need an *attention* mechanism that can give different weights to words of the post and generate an encoded text representation using the weighted words, thus focusing on important parts.

Given the encoded text representation of a post p, as $\mathbf{h}^p = \{\mathbf{h_1}, \mathbf{h_2}, \cdots, \mathbf{h_n}\}$ from the bi-directional GRU component, the attention mechanism [4, 54] weights each of the hidden states of the words i.e. $\mathbf{h_i}$. For each $\mathbf{h_i}$, we compute a weight a_i for its corresponding word w_i and get an attention vector $\mathbf{a} = \{a_1, a_2, \cdots, a_n\}$ as:

$$a_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}, \text{ where} \tag{4.2}$$

$$e_i = tanh(\mathbf{W_i}\dot{\mathbf{h}}_\mathbf{i} + b_i) \tag{4.3}$$

where $\mathbf{W_i}$ is a weight matrix of dimension $1 \times g$, and b_i is the bias term. The text representation with attention is then computed as:

$$\mathbf{c} = \sum_{i}^{n} a_i \mathbf{h_i} \tag{4.4}$$

Note that a single attention layer is insufficient, since the attention weights$(\mathbf{a})$ should not be general but should instead be dependent on different users' interests. Naïvely, to achieve per-user attention, we need U such attentions. This will significantly expand the number of parameters to be estimated to an extremely large value ($U \times n \times g$), which is infeasible to train due to the scalability issue. Additionally, in most datasets not enough data-points are available for all users, to reliably learn the individual attention vectors.

We assume that, since the forums are topical, the users can be softly clustered in a finite number of clusters depending on their interests. The number of clusters k, would be much smaller than U (i.e. $k \ll U$). Therefore, instead of learning U different attention vectors,

we only need to learn k such vectors. This reduces the parameter space hugely. We call this as cluster sensitive attention mechanism. From the same hidden text representation $\mathbf{h}^p$, we learn k different attention weight vectors $\mathbf{a^1}, \mathbf{a^2}, \cdots, \mathbf{a^k}$. Thereafter, by using the different attention weights on $\mathbf{h}^p$, we get a cluster sensitive encoding of the post text p ($\mathbf{C}^p = \mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_k}$).

4.3.3 Multi-label Prediction

For a post p, we concatenate the k text encodings and feed through a fully connected layer with U output neurons. For each of the output neuron (corresponding to each user), the fully connected layer learns the weights for its k inputs (corresponding to the different text encodings).

$$\mathbf{z}_p = tanh(\mathbf{W}.\mathbf{C}^p + \mathbf{b}) \tag{4.5}$$

where $\mathbf{W}$ and $\mathbf{b}$ are weight and bias matrices respectively and $tanh$ is an element-wise non-linear activation function. The output of this feed-forward layer $\mathbf{z}_p \in \mathbb{R}^U$ is then passed through a *sigmoid* activation function to scale each of its element value in the range [0,1]. The model is trained using binary cross-entropy as the loss function which is defined as,

$$\mathcal{L} = -\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{U}\left(y_{ij}.\log(\sigma(z_{ij})) + (1-y_{ij})\log(1-\sigma(z_{ij}))\right) \tag{4.6}$$

where σ denotes the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, z_{ij} is j^{th} element in $\mathbf{z}_i$, and y_{ij} is the ground truth value for j^{th} user (label) and i^{th} post. Our network is end-to-end trainable and is optimized with Adam optimizer [45].

4.4 Experiments

To evaluate the generalization ability of our model, we experimented with multiple datasets from different domains involving users, and some form of textual items in a recommendation scenario. We also present a comparison with some of the well-known content based recommendation systems, as well as the state-of-the-art XMLC approaches to show its ef-

fectiveness.

4.4.1 Dataset

Table 4.1: Dataset Statistics

Dataset	#users	#threads		Avg #word in thread		Avg #user per thread		Sparsity
		train	test	train	test	train	test	
1. Epilepsy	1506	1644	412	147	168	7.39	9.29	99.49%
2. ALS	3182	6466	1617	148	135	9.85	9.75	99.69%
3. Fibromyalgia	5669	8576	2144	203	233	9.02	9.14	99.84%
4. Stackoverflow	69,631	20,137	5035	93	99	6.81	7.29	99.99%

We used the following datasets in our experiments.

- **[1-3] Health Forum:** a popular online health discussion forum website where users can post a thread asking something related to their disease. Other relevant users reply in the threads to share their experiences with it. The website consists of subforums for different diseases. We used three subforum datasets i.e., 'Epilepsy,' 'ALS', and 'Fibromyalgia' for the experiments. We removed threads which have replies from lesser than 4 users or greater than 100 to get rid of extremely off-topic or survey threads.

- **[4] Stackoverflow:** is a CQA website for programming related questions. We obtained a data-dump from Kaggle[1]. We have used all the questions posted during $2008 - 2010$ to form the dataset. We have removed all the code snippets (encapsulated within the tags '⟨code⟩⟨/code⟩') from the question texts.

The dataset statistics are presented in Table 4.1. We observe that the number of labels (i.e., users) is quite large in the stackoverflow dataset. This leads to extremely high sparsity (99.99%) as well. We will describe in Section 4.4.5 how this affects the recommendation accuracy compared to the others.

[1] https://www.kaggle.com/stackoverflow/stacksample/data

4.4.2 Metrics

In our setting, the label set is huge with very high sparsity. Therefore we do not use overall accuracy as our evaluation metric and only aim to evaluate the positive instances i.e. the users who actually participated in a thread. To ensure participation, the ranking quality of the recommended list of users should be evaluated and commonly used metrics for such evaluation include the Mean Reciprocal Rank, precision at top M, Normalized Discounted Cumulated Gains at top M, and Recall at top M. Even though *precision at top M* is usually used for evaluating XMLC methods, it is not appropriate in our case. This is due to the fact that the labels are implicit user feedback. A negative instance could imply that the user is actually not interested in the thread but could also imply that the user had not seen it (and could have been interested in).

We use the following three metrics to evaluate the recommendation quality of the competing methods

- **Mean Reciprocal Rank (MRR)** indicates the position of the first relevant user in the ranked list. This measures the ability of a system in identifying an interested user at the top of the ranking. Let r_t be the rank of the highest ranking relevant user for a test thread t. MRR is just the reciprocal rank, averaged over all threads in test set, n:

$$MRR = \frac{1}{n}\sum_{t=1}^{n}\frac{1}{r_t}$$

- **Recall@M** considers how many top-M users actually interacted with the thread (higher is better). Recall for the entire system is computed as the average recall value for all threads in test data.

- **Normalized Discounted Cumulative Gain (NDCG@M)** is well suited for evaluation of recommendation system, as it rewards relevant results ranked higher in the returned list more heavily than those ranked lower. NDCG@M for a thread t is com-

puted as:

$$NDCG_t = Z_t \sum_{j=1}^{M} \frac{2^{r(j)} - 1}{log(1+j)}$$

where Z_i is a normalization constant calculated so that a perfect ordering would obtain NDCG of 1; and each $r(j)$ is an integer relevance level (for our case, $r(j) = 1$ and $r(j) = 0$ for relevant and irrelevant recommendations, respectively) of result returned at the rank $j \in \{1, \cdots, k\}$. Then, for each M value, $NDCG_t$ is averaged over all (n) threads in the test set to get the overall NDCG@M.

In our evaluation, we experiment with $M = \{5, 10, 30, 50, 100\}$ to determine the quality of recommendation at different thresholds of the ranked list.

4.4.3 Baselines

We compare our model with the following competing methods:

CVAE [52] : was proposed to tackle cold start problem using a Bayesian generative model. It reportedly outperforms many state-of-the-art recommendation systems by considering both rating and textual content using deep learning.

CTR [102] : takes LDA [14] discovered topic distributions as input along with the user-item interaction matrix. This has proven to be a very solid baseline for cold start problem and we use it as a representative of traditional recommendation algorithms.

CNN-Kim [44]: constructs a document vector with its constituent word embeddings, and then convolutional filters are applied to this feature maps. The features pass through a max-over-time pooling layer to construct the document representation. For prediction, the document representation is fed to a fully-connected layer with L softmax outputs, corresponding to the L labels.

XML-CNN [55]: introduces some advancements over CNN-Kim. It adopts a dynamic max pooling scheme, a bottleneck layer and a loss function more suitable for multi-label prediction. It has reportedly outperformed many traditional XMLC models over several datasets.

BiGRU-2: is a baseline implemented by us which uses a stack of two Bidirectional GRU layers for text representation. This is essentially equivalent to our model without the CSA component.

4.4.4 Experimental Settings

Pre-processing for CTR is done as per the recommendations in the paper. We remove all the stopwords and compute tf-idf scores for all the words in all the documents in the training set and retain the top 8000 words to form the vocabulary. Thereafter LDA is run with 100 topics and LDA discovered document-, and word-topic distributions are provided to CTR. For CVAE we used the implementation provided by the authors[2].

For the CNN based models (CNN-Kim and XML-CNN), we used rectified linear units as activation functions, and one-dimensional convolutional filters with window sizes of $2, 4, 8$. The number of feature maps for each convolutional filter was 128. For XML-CNN the dropout rate was $p = 0.5$, and hidden units of the bottleneck layer was 512 as suggested by the authors [55].

For the baseline BiGRU-2 and the proposed model, we set the number of neurons for the GRUs to 128, and number of clusters (k) to 100. A dropout layer with 0.3 dropout rate is used after the fully connected layer. To deal with the highly imbalanced class distribution, we use normalized class weights to weigh the sparse positive training examples more. All the deep learning models are implemented using Keras library[3] with Theano[4] as the backend.

[2] https://github.com/eelxpeng/CollaborativeVAE
[3] https://keras.io/
[4] https://github.com/Theano/Theano

4.4.5 Results

Table 4.2, 4.3, and 4.4 show the performance of different methods on the four datasets in terms of MRR, Recall@M, and NDCG@M respectively.

Firstly, we note that all the XMLC models outperform the widely used off-the-shelf recommendation algorithms comfortably in most cases. However the same does not hold true for the off-the-shelf text classifier as CNN-Kim's scores are not always better. This empirical proof works as a validation of our approach of posing cold start recommendation problem as an XMLC task.

Moreover, we observe that our model outperforms the baselines consistently in all datasets. We achieve a relative performance gain of $4.5\% - 21.7\%$ (depending on the dataset) in terms of MRR compared to current state-of-the-art for XMLC i.e., XML-CNN.

We find the performance of the models to be consistent in terms of both Recall, and nDCG@M. From the NDCG scores we conclude that our model is able to correctly identify interested users and places them near the top of the list in most cases. For $M = 100$, we achieve a relative performance gain of $7.79\% - 16.19\%$ in terms of NDCG compared to XML-CNN. We observe similar trends in case of recall, with a relative performance gain of $3.23\% - 15.39\%$. We would like to mention that in our setting, the recall values at larger M values are equally important as the lower ones – quite unlike the traditional case, where a recommended list of items are presented to every user. Since it is infeasible for a user to look through more than the first $5 - 10$ items, the objective is to have better recall, and NDCG scores for small M (e.g., $5-10$). However for a new item, we are trying to identify the set of interested users who would be notified individually. Typically the recommendation engines try to notify as many interested users as possible to ensure sufficient user-engagement. For this reason, we argue that our model would be more appropriate as it consistently achieves higher recall, and NDCG scores for large M values compared to the state-of-the-art for XMLC. Although CVAE uses both rating and textual content, we observe that it struggles to provide accurate recommendation in our scenario. It was reported to outperform other methods when it has seen the test item *at least* once [52]. However in our case, the test item

is *never* seen during training – we believe this makes it challenging for CVAE to perform well.

In absolute terms, the performance of all the competing methods degrade drastically in case of the stackoverflow dataset because of extremely high sparsity (99.99%) and huge label space ($\sim 70K$). However relatively speaking, our model fairs well compared to the others with better (in most cases) or very close scores (in few cases) in terms of all the metrics.

Ablation Study: The choice of baselines allows us to do two ablation studies. Firstly, we observe that BiGRU encoding of text works much better compared to XML-CNN which uses CNN to encode the text. We believe that the long sequential nature of posts is better captured with a recurrent network rather than fixed length convolution filters. Finally, recall that the BiGRU-2 is primarily our model without the CSA component. This allows to us to an ablation study between our model variants with/without it. We observe that the attention mechanism achieves a relative performance improvement of upto 6.33% over the BiGRU-2 model in MRR, 3.40% in Recall@100, and 4.67% in NDCG@100 respectively. Moreover, the attention mechanism consistently scores better than BiGRU-2 for larger values of M. This study quantitatively validates the hypothesis of having the CSA component in our model.

Table 4.2: Comparison of Mean Reciprocal Rank (MRR) of different methods for the four datasets.

Dataset	Methods					
	CVAE	CTR	CNN-Kim	XML-CNN	BiGRU-2	Our Model
1.Epilepsy	0.159	0.443	0.536	0.551	0.631	**0.671**
2.ALS	0.201	0.275	0.270	0.293	0.297	**0.306**
3.Fibromyalgia	0.304	0.435	0.669	0.668	0.740	**0.773**
4.Stackoverflow	0.003	0.032	0.025	0.029	0.047	**0.050**

Table 4.3: Comparison of Recall@M of different methods across four datasets

Dataset	Metric	Method					
		CVAE	CTR	CNN-Kim	XML-CNN	BiGRU-2	Our Model
1. Epilepsy	recall@5	3.69	17.46	17.23	**22.76**	22.64	22.65
	recall@10	7.22	27.67	22.93	**34.67**	29.22	29.26
	recall@30	21.14	43.83	44.63	49.08	50.99	**51.21**
	recall@50	29.62	50.86	52.45	53.69	59.47	**59.80**
	recall@100	42.44	59.93	65.77	63.67	68.23	**69.37**
2. ALS	recall@5	4.17	7.05	6.19	6.51	7.63	**9.23**
	recall@10	7.07	12.08	10.09	11.44	**14.65**	13.89
	recall@30	17.04	25.00	22.15	23.56	30.18	**31.84**
	recall@50	24.07	32.46	31.27	30.61	36.32	**36.55**
	recall@100	35.77	44.14	43.82	43.14	48.14	**49.78**
3. Fibromyalgia	recall@5	8.24	14.58	23.01	22.11	25.63	**25.97**
	recall@10	14.93	27.18	34.77	33.88	35.18	**37.38**
	recall@30	32.83	54.39	58.04	61.83	62.39	**63.06**
	recall@50	42.43	63.91	67.83	68.92	69.17	**72.04**
	recall@100	55.02	72.31	76.37	75.74	77.98	**78.19**
4. Stackoverflow	recall@5	0.02	0.59	0.46	0.51	0.66	**0.86**
	recall@10	0.06	1.14	0.73	0.97	1.15	**1.30**
	recall@30	0.16	2.73	1.84	2.42	**2.94**	2.80
	recall@50	0.31	4.02	2.74	3.43	4.03	**4.11**
	recall@100	0.69	**6.36**	4.43	5.35	6.09	6.33

Table 4.4: Comparison of NDCG@M of different methods across four datasets

Dataset	Metric	Method					
		CVAE	CTR	CNN-Kim	XML-CNN	BiGRU-2	Our Model
1. Epilepsy	NDCG@5	3.80	19.44	19.52	25.80	27.80	**29.52**
	NDCG@10	5.95	25.80	24.26	33.08	31.96	**33.72**
	NDCG@30	12.26	33.38	34.10	39.91	41.78	**43.91**
	NDCG@50	15.38	36.02	38.49	41.70	45.14	**47.01**
	NDCG@100	19.50	38.97	42.18	44.88	47.93	**50.17**
2. ALS	NDCG@5	5.28	8.59	8.02	8.21	9.16	**10.24**
	NDCG@10	7.26	11.94	10.62	11.41	**13.71**	13.42
	NDCG@30	12.12	18.18	16.38	17.40	21.05	**22.49**
	NDCG@50	14.90	21.08	19.88	20.13	23.54	**23.86**
	NDCG@100	18.90	25.00	24.14	24.39	27.53	**28.34**
3. Fibromyalgia	NDCG@5	10.29	17.27	28.97	28.57	32.38	**33.71**
	NDCG@10	14.72	25.43	33.67	36.32	38.44	**41.05**
	NDCG@30	23.53	38.82	48.23	50.19	51.09	**54.03**
	NDCG@50	27.29	42.50	52.04	52.98	54.53	**57.36**
	NDCG@100	31.46	45.36	54.95	55.32	57.52	**59.63**
4. Stackoverflow	NDCG@5	0.02	0.64	0.54	0.59	1.01	**1.22**
	NDCG@10	0.04	0.98	0.70	0.87	1.31	**1.48**
	NDCG@30	0.09	1.68	1.19	1.52	2.09	**2.12**
	NDCG@50	0.14	2.14	1.51	1.88	2.47	**2.59**
	NDCG@100	0.26	2.86	2.03	2.46	3.11	**3.27**

4.5 Conclusion

We have addressed cold start thread recommendation in online forums, which is an important task to ensure user engagement. We recommend newly-posted threads to interested users in the community for participation. Mainstream recommendation systems cannot use collaborative filtering to address this phenomenon as there is no interaction history for such items.

We have applied an alternative approach utilizing extreme multi-label classification. In particular, we proposed a novel neural network architecture consisting of stacked bi-directional GRUs for text encoding, coupled with cluster-sensitive attention to address scalability, and sparsity.

Specifically, leveraging our insight that sets of users display different levels of interest within a long post text, the cluster-sensitive attention incorporates user interests by learning multiple attention layers for attending to different parts of a text. This cluster-sensitive attention layer also helps us in addressing the sparsity issues usually associated with extreme multi-label classification approaches, by exploiting the correlation between users within clusters. Thorough experimental evaluation show that the proposed model outperforms existing content based recommendation systems, deep learning based text classification systems, as well as state-of-the-art multi-label classification approaches.

In the future, we plan to model how interests of community members change over time. Not all users will remain interested in the same topic over a long course of time as their experiences and expertise change. Also, we encourage the research community to try our approach in other domains where recommending new items to interested users is the priority such as news articles, tweet recommendation, social media news feed generation and so on.

Chapter 5

Beyond Threads: Identifying Helpful Posts

5.1 Introduction

We have presented how a robust thread recommendation system for the discussion forum users can be built so far. Although this recommendation step is important to find information from the huge pool of threads, unfortunately this is not enough to provide a pleasant reading experience to the users. Due to the open nature of the forums and the various expertise level of users, the posts in the discussion threads vary in helpfulness. To address this, some websites provide options for users such as "Upvote" (`reddit`, `stackoverflow`), "Highlight" (`coursera`), etc. Such feedback is helpful for identifying important posts among the many. However, such feedback rarely comes immediately when new posts are created, affecting their visibility to the users [90]. In this chapter, we would devise technology to proactively identify such helpful posts as they arrive, in a *helpfulness prediction task*, as it would enable users to efficient relevance assessment.

We observe that there is a key structural difference between online discussion forums and Community Question Answering (CQA) websites. Figure 5.1 shows the distribution of normalized helpful votes for the top-5 posts across a popular discussion forum (`reddit`), and a CQA website (`stackoverflow`[1]). In CQA, the vote distribution decays exponentially, indicating that usually there is a single correct answer with the largest number of

[1] `https://www.kaggle.com/stackoverflow/stacksample/data`

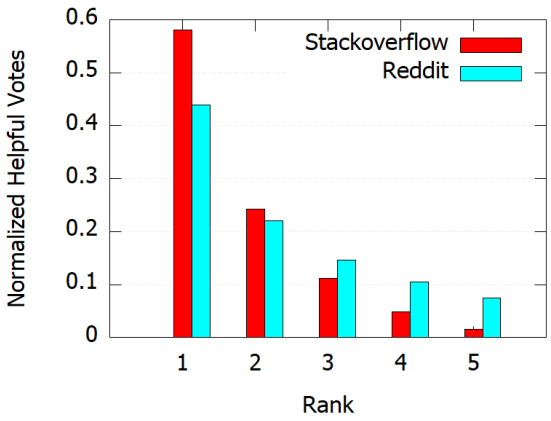


Figure 5.1: The helpful vote distribution for the top-5 posts across an online discussion forum (`reddit`), and the `stackoverflow` CQA website. The helpful votes decay at a slower rate for `reddit` compared to focused CQA.

votes [71]. In contrast, votes for less helpful posts in discussion forums decay at a much lower rate, suggesting that discussion forum threads are more open-ended.

Table 5.1 shows a sample thread from `reddit` to understand the dynamics of online discussion. We observe the following two major differences compared to threads in CQA domain: (1) The first post (hereafter, *original post*) is not necessarily a question, but can also be personal anecdotes or new findings on a certain topic, attracting more discussion. (2) Instead of searching for a single relevant answer in CQA, discussion forum users find a post as helpful, when it introduces some *relevant* (with respect to the original post) and *novel* (*i.e.*, not presented in the earlier posts within the same thread) information. Motivated by these observations, we address the helpfulness prediction by considering both the target post and its preceding posts.

We propose a novel neural architecture to predict the helpfulness of a post in a discussion thread. Our approach consists of two components: (1) modeling the *relevance* of a post and (2) determining the *novelty* with respect to the sequence of preceding posts. It combines the output from both components to predict the overall post helpfulness. As recurrent neural networks (RNNs) have shown good performance in sequence modeling tasks [21, 96], we apply it to our architecture to model the (i) sequence of words in the post text, and the (ii) sequence of posts in a thread. Our model significantly outperforms other state-of-the-art models across experiments on five varied and large forum datasets. Our main contributions

Table 5.1: A sample discussion thread from `reddit`. Helpful votes are provided by the website users.

Order	Post Text	Helpful?
Original post	I was working yesterday..and my back was bent over and when I got up I felt like I strained my back but now my mind is linking it to my kidney..	Yes
1	I have this and my doc has told me it's muscular and physio might help..	Yes
2	Kidney pain is usually constant and doesn't change when you move, or get better when you change position, from how I understand it..you'll be fine :)	Yes
3	If it happens only when you move there is a big chance it's a muscle spasm, this happens after some physical activities.	No

are:

- We reveal the key differences between posts in CQA and online discussion forums;
- We analyze the confounding factors behind the perceived helpfulness of posts in discussion forums. We observe that both *relevance* and *novelty* play important roles in determining the helpfulness of a post;
- We propose a novel neural network architecture to predict the helpfulness by using textual content of a target post as well as sequence of posts preceding it in the thread;
- We compare our model with current neural network classifiers and analyze the factors that influence our model's performance.

5.2 Methods

We propose a neural network architecture shown in Figure 5.2a to model post helpfulness. The architecture is end-to-end trainable, adaptable to different domains.

Our model takes advantage of the hierarchical nature of the thread and post's parent–child relationship. This phenomenon is common to several applications of natural language processing. We are inspired by the hierarchical recurrent encoder decoder approach from context-aware query generation [93], which was adopted across other downstream applica-

tions such as conversations on Twitter [35, 46, 97], dialogue systems [85], and for thread and post modeling in MOOC discussion forums [17].

The model comprises two components to analyze a target post's thread *relevance* and *novelty* with respect to its past k posts.

5.2.1 Text Encoder

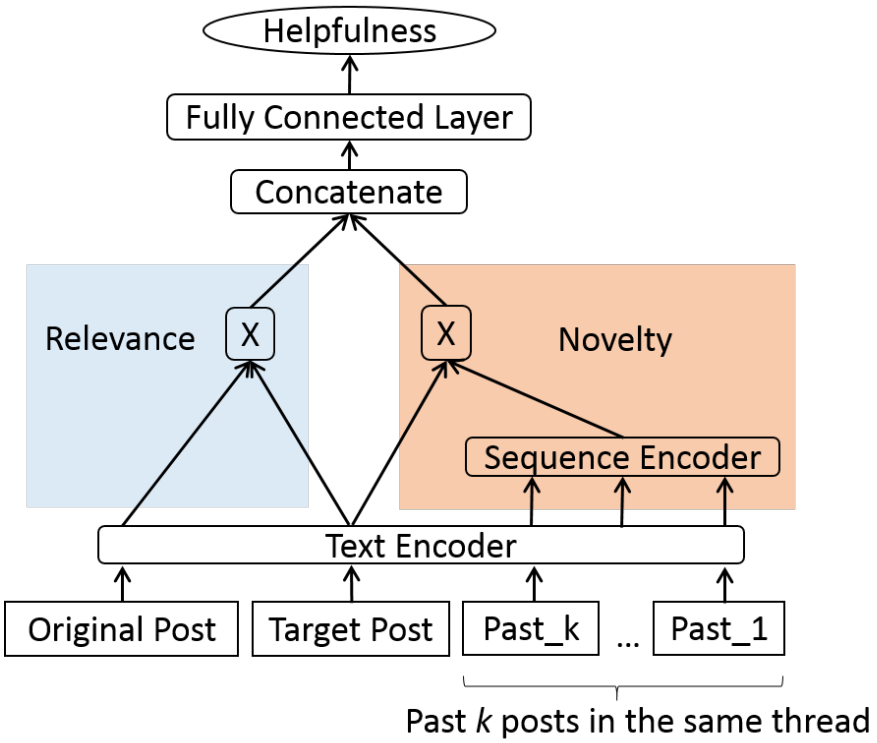
This component takes a post text p which consists of words $(w_1, w_2, \ldots, w_n)$ as input and encodes it to a tensor ($\mathbf{h}^p$) in two steps. We first use a word embedding initialized with GLoVe[2] to transform all the words from the post text into finite d-dimensional vectors, *i.e.*, $w_i \mapsto \mathbb{R}^d$. Our experimental results on multiple datasets show that the coverage of GLoVe varies between $68 - 76\%$. To estimate the embeddings for the out-of-vocabulary words and reflect the domain dependence, we keep the embedding vectors trainable. In the second step, the sequence of words are provided to a gated recurrent unit (GRU) layer [21] to obtain a sequence of hidden vectors $(\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n)$, where $\mathbf{h}_i \in \mathbb{R}^g$, and g is the output dimension of the GRU encoded tensor. The latent vector is defined as follows:

$$\mathbf{h}_i = \text{GRU}_{\text{text}}(\mathbf{h}_{i-1}, w_i).$$

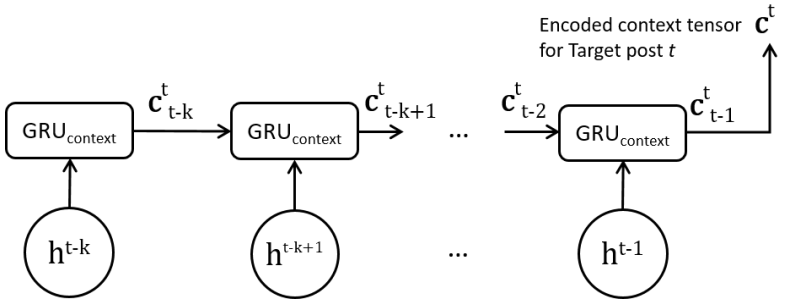
The last vector in the sequence, *i.e.*, $\mathbf{h}_n$ is considered as the encoded representation of a post text (*cf* Figure 5.2c). For a post p, the GRU_{text} encoded representation is denoted as $\mathbf{h}^p$. We use a dropout layer after the GRU to prevent overfitting. Note that, there is only a single text encoder in our model. All the textual inputs – the target post, original post, and each of the past posts in the thread – are encoded using a single text encoder, since as all of them are essentially textual posts of similar nature.

Alternative Architectures. We also tried stacking additional GRUs in our experiments, but we did not observe accuracy improvements. We also tried to replace GRU with LSTM, resulting in similar performance at the cost of much longer training time due to the larger number of parameters.

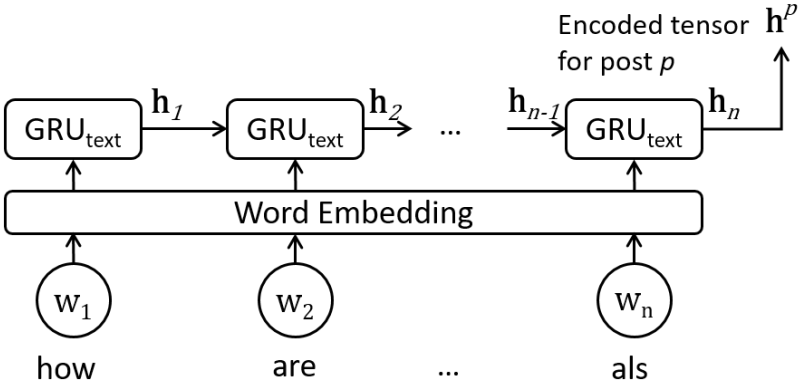
[2] `http://nlp.stanford.edu/data/glove.6B.zip`



(a) Overall network



(b) Sequence Encoder



(c) Text Encoder

Figure 5.2: Our neural architecture and its components. (a) Overall network architecture. The shaded component on left side captures relevance with respect to the original post; the right measures the novelty compared against the past k posts. (b) Unrolled layout for the Sequence Encoder ($\text{GRU}_{\text{context}}$). (c) Unrolled layout for the Text Encoder (GRU_{text}).

5.2.2 Modeling Post's Relevance

The left component of Figure 5.2a captures the relevance of a target post with respect to the original post. It obtains two GRU encoded tensors: one for the target post $\mathbf{h}^t$, and the other one for the original post $\mathbf{h}^o$. It computes the similarity between these two tensors, defined as:

$$\mathbf{r}_t = \mathbf{h}^t \otimes \mathbf{h}^o,$$

where $\otimes$ denotes the element-wise multiplication. We also experimented with element-wise difference and cosine similarity, but found that the multiplication operation works best. Our relevance modeling component is inspired from the architecture for answer sentence selection model [114].

5.2.3 Modeling Post's Novelty

In Figure 5.2a, the right component models the target post's novelty compared to the past k posts from the same thread. It takes the encoded tensors for the target post $\mathbf{h}^t$ as input, as well as the past k posts $(\mathbf{h}^{t-k}, \mathbf{h}^{t-k+1}, ..., \mathbf{h}^{t-1})$.

We first encode the context of the discussion by modeling the sequence of the past k posts. In order to achieve this, we use another GRU (labeled as Sequence Encoder in Figure 5.2a) to transform the sequence of k post tensors to a single context tensor $\mathbf{c}^t$ of equal dimension g. Each timestep i of this is defined as follows:

$$\mathbf{c}_i^t = \text{GRU}_{\text{context}}(\mathbf{c}_{i-1}^t, \mathbf{h}^{t-i}).$$

Similar to GRU_{text}, the last vector in the sequence, *i.e.*, $\mathbf{c}_{t-1}^t$ is considered as the context representation $\mathbf{c}^t$ (as shown in Figure 5.2b).

To determine the novelty of the target post, we compute its similarity $\mathbf{n}_t$ with the discussion thread context represented by its context tensor:

$$\mathbf{n}_t = \mathbf{h}^t \otimes \mathbf{c}^t.$$

Importantly, instead of considering all the previous posts in the thread, we limit the context to the past k posts for two reasons:

1. Users may not recall the entire context of discussion while reading a post appearing much later in a long-running thread.

2. Users often arrive at a discussion thread through search engine queries. Since long-running threads are paginated, a user may land on a page in the middle of a discussion thread, thus also missing the previous context.

We find empirical evidence for these hypotheses later in our experiments (see Section 5.4). In tuning our model, we observed that increasing the context length beyond a threshold does not yield improvements.

5.2.4 Final Helpfulness Prediction

We combine the relevance tensor ($\mathbf{r}_t$) and novelty tensor ($\mathbf{n}_t$) and feed through a fully connected layer to make the final post helpfulness prediction:

$$\mathbf{x}_t = \mathbf{r}_t \oplus \mathbf{n}_t,$$

$$p(y|\mathbf{x}_t) = sigmoid(\mathbf{W} \cdot \mathbf{x}_t + \mathbf{b}),$$

where $\oplus$ denotes concatenation; $\mathbf{x}_t$ is the concatenated tensor; y is the output label (0 or 1); $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and bias vector, respectively, learned for the fully connected layer. We use binary cross-entropy loss to train the model, optimizing with Adam [45].

Alternative Architectures. We also investigate ensemble architectures. We fed the relevance and novelty tensors through two separate fully connected layers and obtain the binary predictions from both components concurrently, then merging the two predictions via a final fully connected layer for obtaining prediction. This approach fared worse compared to our concatenation-based model, possibly as our final concatenation model can exploit the non-linear interactions between the two components. The actual post content is never presented to the fully connected layer so that it generalizes well. The final layer only gets to see the

relevance, and novelty vectors. This ameliorates the creation of overfitted (post-based or thread-based) features for the helpfulness prediction task.

5.3 Experiments

We first describe the datasets, evaluation metrics, and baseline models before our main results. We conducted additional experiments to answer specific research questions about our model.

5.3.1 Datasets

We experiment with five real-world online discussion forums (Table 5.2) to validate model effectiveness. We remove threads that have less than two posts.

1–2. **Reddit** is a popular platform for discussions on a wide-variety of topics on the web. We use a large number of discussion threads from a `reddit` data dump[3]. To diversify the datasets in terms of average thread length, we set different thresholds, and created two datasets: *Reddit_10+* (≥ 10 posts) and *Reddit_3+* (≥ 3 posts). Along with a chronologically ordered set of posts, `reddit` also has "Upvote" counts for every post.

3–4. **Coursera** is a large MOOC platform, providing a discussion forum for the course participants. We select two courses with the largest number of posts: "Matrix-001" and "Android Apps 101-001" from a MOOC dataset [16]. Course participants can "vote" for a post if they find it helpful. We refer to these datasets as *Matrix* and *Android Apps*, hereafter.

5. **Travel Stack Exchange**[4] is one of many QA websites in the Stack Exchange community. We use a data dump[5] of the website and refer to it as *Travel* dataset. In Travel Stack Exchange, a user can "Upvote" a post if she deems it helpful. Although not strictly a discussion forum, the threads in this forum appear to be less objective (by our vote distribution analysis, similar to Figure 5.1), compared to other CQA sites like `stackoverflow`.

[3] `https://files.pushshift.io/reddit/comments/`

[4] `https://travel.stackexchange.com/`

[5] `https://archive.org/download/stackexchange/travel.stackexchange.com.7z`

Table 5.2: Dataset statistics.

Dataset	# Posts	# Threads	Avg # Posts / Thread	Avg # words / Post
1. Reddit_10+	200,006	9,744	20.52	29.45
2. Reddit_3+	200,016	28,763	6.95	30.58
3. Android Apps	11,643	2,077	5.60	56.53
4. Matrix	10,159	2,484	4.08	65.30
5. Travel	30,116	10,250	2.93	163.43

5.3.2 Post Annotation and Evaluation Metrics

We use the user-provided feedback in form of "mark as helpful", "like", "upvote" actions as a proxy of the actual helpfulness of a post. We had considered using human annotation to obtain ground truth labels, but judged this as problematic since such annotations will require specific domain expertise; *e.g.*, in a MOOC on philosophy, marking forum posts as helpful requires both domain knowledge and context within the time frame of the course to judge helpfulness. Vote counts vary widely across posts and threads, (*i.e.*, 0 to 3,100 for the `reddit` dataset), making it infeasible to formulate the task as a regression problem. Following by prior published research [20, 57], we model it as a binary classification problem, and use the 80^{th} percentile expected value of helpful vote count across all the posts as the boundary between the two classes. We assume that a post is *helpful* if it has received more helpful votes than the 80^{th} percentile, and *not helpful* otherwise. Instead of using the helpful vote counts across all posts, one could also consider the 80^{th} percentile within a thread to divide the posts in it in two classes.

Since our goal is to predict the helpful posts and the class distribution is inherently skewed from our definition, we evaluate the model performance in terms of prediction accuracy for only the positive, helpful class. We evaluate using standard precision, recall, and F_1 score across all datasets.

5.3.3 Baselines

We compare our model with the following state-of-the-art neural text classification methods:

1. BiLSTM [94]: a stack of two layers of Bidirectional LSTM encoders on post text.

2. Stacked LSTM [56]: a stack of two layers of LSTM encoders on the post text.

3. LSTM with Attention [80]: an LSTM layer with hierarchical attention[6].

4. Answer Sentence Selection [114]: a CNN model pioneered in a TREC QA[7] task.

5. Our Model (Novelty based): only the novelty component of our model.

We do not include traditional feature-based models as part of our reported baseline portfolio, as in our study, neural models have outperformed them as well, which is corroborated in recent studies [44]. Additionally, such approaches are fragile, as we experiment with datasets from multiple domains with various discussion styles, and extracting hand crafted features for each is non-trivial and labour intensive.

5.3.4 Training

We used the Keras[8] library with TensorFlow[9] as the backend for model implementation. We split the dataset 80:10:10 for train, validation, and test, respectively, and perform 5-fold cross validation. We tuned the hyper-parameters via grid search on the validation set for all the models.

The rest of the parameters used follow standard values from the recent literature. We set word embedding dimension (d) to 100, vocabulary size to 100K, hidden dimension of GRU (g) to 128, batch size to 512, the dimension of the final fully connected layer to 128, and use 70% dropout. For the CNN-based Answer Sentence Selection baseline, we tuned the number and size of filters (128 and 3, respectively). The maximum length of post text was set according to average post length (in the training split) for each dataset.

[6] https://gist.github.com/cbaziotis/7ef97ccf71cbc14366835198c09809d2
[7] http://trec.nist.gov/data/qa.html
[8] https://keras.io
[9] https://www.tensorflow.org/

Table 5.3: (P)recision, (R)ecall and F_1 comparison of model performances across our five datasets representing three domains. Our model outperforms other state-of-the-art neural text classifiers consistently. Ablation study with Answer Selection, and Novelty-based model shows that modelling both relevance, and novelty is important.

Model	1. **Reddit_10+**			2. **Reddit_3+**			3. **Android Apps**			4. **Matrix**			5. **Travel**		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
BiLSTM [94]	0.23	0.23	0.23	0.23	0.22	0.22	**0.36**	0.32	0.34	0.29	0.35	0.32	0.28	**0.31**	0.29
Stacked LSTM [56]	0.24	0.21	0.22	0.23	0.20	0.21	0.34	0.29	0.31	0.32	0.29	0.31	0.23	0.26	0.25
LSTM w/ Attention [80]	0.24	0.21	0.23	0.24	0.21	0.22	0.34	0.27	0.30	0.30	0.36	0.33	0.25	0.26	0.25
Answer Selection [114]	0.28	0.27	0.27	0.31	0.32	0.32	0.28	0.21	0.24	0.33	0.34	0.33	0.30	**0.31**	0.31
Our Model (Novelty-based)	**0.53**	0.38	0.44	**0.42**	0.27	0.33	0.33	0.24	0.28	**0.43**	0.27	0.33	**0.47**	0.27	**0.34**
Our Model (full)	0.48	**0.53**	**0.51**	0.41	**0.39**	**0.40**	0.35	**0.40**	**0.38**	0.37	**0.37**	**0.37**	0.37	**0.31**	**0.34**

5.3.5 Results

Table 5.3 shows the comparison of model performance over the five datasets. We observe that our full model consistently outperforms others in terms of F_1 across all datasets. Our novelty-based model gives the second best score in all datasets except for *Android Apps*. Comparing our novelty-based model against answer selection model, we observe that the helpfulness of a post depends on both its relevance to the original post and the novelty with respect to earlier posts in the same thread. The evaluation scores obtained by the state-of-the-art neural text classification models strongly support this observation. They consistently make less accurate prediction compared to the relevance- and/or novelty-based models. Among them, BiLSTM or LSTM with Attention model achieves the best performance, dependent on the dataset. We discuss the confounding factor affecting performance in Section 5.4.

We also observe that the prediction is more accurate when there is sufficient context to learn the dynamics of the discussion forums. In *Reddit_10+*, and *Reddit_3+*, where both datasets average about 20 and 7 posts per thread respectively, we obtain F_1 score of 0.40 to 0.51. In the other datasets, where the average thread length is much shorter ($\sim$ 3 to 5), we obtain relatively low F_1 score of 0.34 to 0.38. Our model is more accurate in `reddit` datasets where threads are longer on average, indicative of more open-ended discussion centered on the original post.

Table 5.4: Illustration of different corner cases for helpfulness prediction. The target post needs to be both relevant to the original post, and novel compared to the previous posts in the thread in order to be helpful.

Original Post	**My fiancée and I are looking for a good Caribbean cruise in October and were wondering which islands are best to see and which Cruise line to take?..**	**I've had bouts of heart burn & this time its sticking around for a while. I ate something really spicy on Tuesday night & its Thursday & Im having heart burn on & off... Please help**	**In a few weeks' time, I will be visiting the US for 14 days. Coming from the EU, roaming is very expensive, so I am considering getting a temporary SIM card..**
Previous Posts	Friends I am staying with are travelling with Royal Carribean on a cruise in October. They are starting from Miami..	You're probably fine. People get heartburn from time to time.. Eat bland food for a few days and that inflammation should subside..	There are many options you can have as far as mobile phone data prepaid plans are concerned. Since you need coverage along the route..
	The Princess Cruise line has a Caribbean cruise in the fall. It may start in November rather than October but could be suitable for your needs..	Heartburn can last a few days and its not always spicy food that triggers it. I assume youre concerned it might be a heart attack. If that was it you would know it by now.	You may want to check your existing phone plan. For example, quite a few providers in the UK offer free or cheap roaming with data included..
	There are plenty of options for the Caribbean in October regardless of it being in hurricane season..	Heartburn doesn't JUST occur from spicy food. If you're having it over multiple days, it could simply be other food. Fatty foods in particular cause it.	If your main goal is price, MetroPCS has no-contract 30 month plans which have unlimited calling US numbers, unlimited SMS, and unlimited data in the US..
Target Post	If you like to dress up and eat high-end food, the cruise line you want is not the one that caters to honeymooners on a tight budget or to families with small kids. If you like things to be..	Stay calm. Drink lots of water. Do you have an antacid you could take? Try to avoid spicy, acidic, caffeine, alcohol for a while..	`willmyphonework.net` is good for checking a phone's compatibility with the various networks. Suggestion before departure, print-out a list of the carriers your phone will work with hard copy is the way to go here..
Helpful?	Yes	No	No

5.3.6 Case Study

We now highlight a few corner cases successfully handled by our model.

Table 5.4 shows three target posts along with the original posts and their previous posts from different datasets. In the first case, we observe that the target post introduces some relevant and novel information into the thread, and thus our model predicts it as helpful.

In the second example, we find that the target post is quite similar to some of the previous posts. Since it introduces less novelty in the discussion, our model predicts the target post as unhelpful, although relevant to the discussion topic. In the third example, the target post seems to be novel compared to the previous posts but it deviates from discussion topic in the original post. Hence, our model does not predict it as helpful.

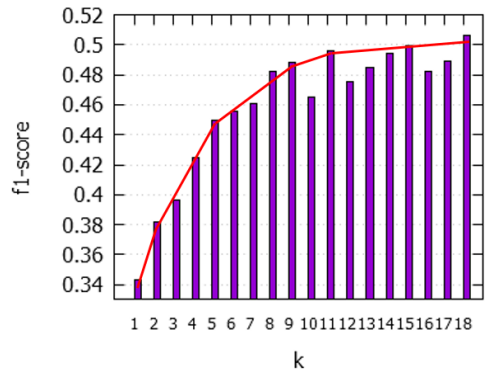
These observations indicate that our model treats each of the two qualities of a target post, *i.e.*, relevance with the original post, and novelty compared to the previous discussion individually as necessary but not sufficient conditions. A target post needs both relevance and novelty so that our model predicts it as helpful.

5.4 Discussion

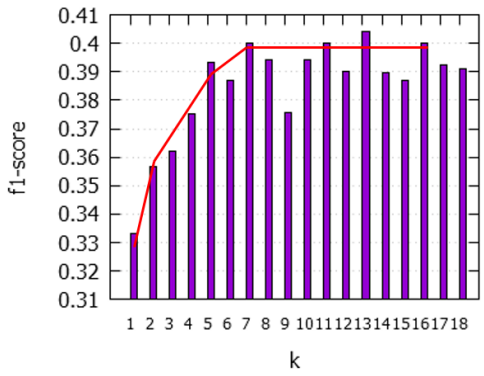
We now answer the following research questions (RQ) to further analyze prediction of helpful posts:

RQ1: How does the past context length influence model performance?

The number of posts across threads varies widely, making it difficult to estimate the optimal value for past context length (k in Section 5.2.3). To understand the effect of k on model performance, we vary k ranging from 1 to 18 and report F_1 for the *Reddit_10+*, and *Reddit_3+* datasets in Figure 5.3. Interestingly, we observe that, the performance stops improving after a certain number of posts in both cases: k=11 and k=7 for *Reddit_10+*, and *Reddit_3+*, respectively.



(a) Reddit_10+



(b) Reddit_3+

Figure 5.3: Model performance while varying context length k for *Reddit_10+*, and *Reddit_3+* datasets. F_1 stabilizes after a certain context length in both cases. Trend line in red.

Table 5.5: F_1 obtained by model variations with average of past post tensors as context tensor, compared to our $GRU_{context}$ based model.

Context Modeling	**Reddit_10+**	**Reddit_3+**	**Andriod Apps**	**Matrix**	**Travel**
Average	0.40	0.35	0.36	0.36	0.33
$GRU_{context}$	**0.53**	**0.40**	**0.38**	**0.37**	**0.34**

Setting too low a k limits the number of past posts the model gets to see, underfitting the data. Large k gives modest performance gains but incurs significant increase in training cost. As discussed in Section 5.2.3, the entire context might be redundant to determine target posts' helpfulness in long threads.

RQ2: Does the order of contextual posts matter?

To investigate whether the order of the past posts matters in determining the helpfulness of a target post, instead of modeling the past posts by $GRU_{context}$ layer, we just use the average of the past post tensors to get the context tensor. Table 5.5 shows the F_1 achieved by this variation compared to our model.

We observe that the model performance degrades when the order of the past posts is ignored and represented by an average. Crucially, we find that the datasets with longer threads suffer more compared to the ones with shorter threads. This observation indicates that the sequential nature of discussion is integral to model construction.

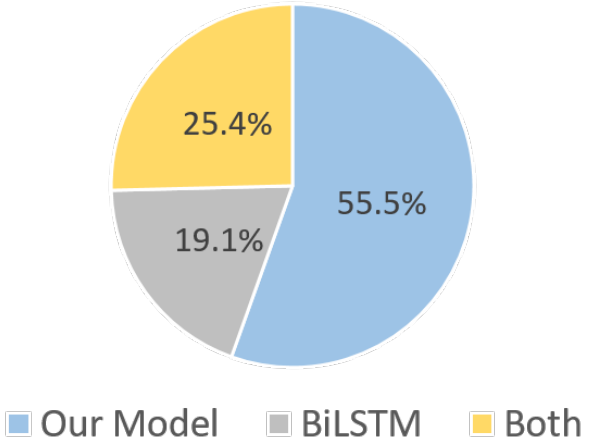


Figure 5.4: Correct prediction share of helpful posts for Reddit_all. Yellow: both models; blue: only our model; grey: only BiLSTM.

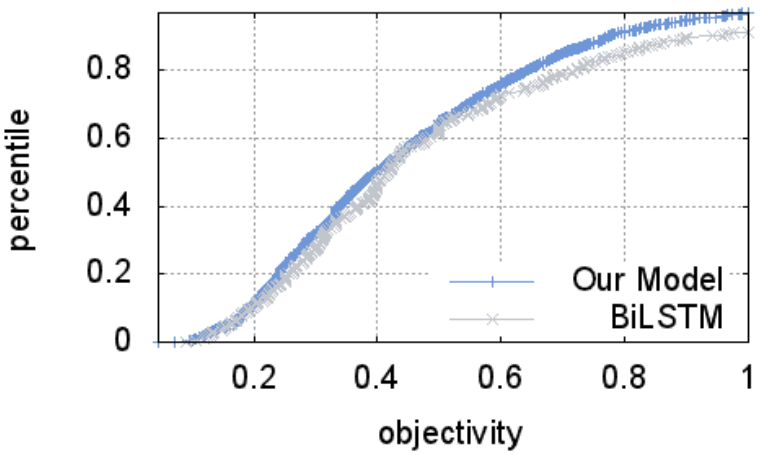


Figure 5.5: Thread objectivity score CDF. The blue curve shows threads where our model is correct and BiLSTM is not; vice versa for the grey.

RQ3: What factors influence performance among the text classification models and our model?

Table 5.3 shows that BiLSTM achieved better scores compared to the other neural text classification models. To better understand differences between BiLSTM and ours, we focus on the cases where one model is correct but not the other (as illustrated for *Reddit_10+* in Figure 5.4). While both models can predict the correct class in 25.4% cases (in yellow), in the other cases (blue and grey), they differ.

We study the objectivity of the posts where such differences were observed. Without

loss of generality, we define a metric called thread *objectivity* spread, in terms of the vote shares for the top-5 posts:

$$objectivity = \frac{max(vote(x)) - min(vote(x))}{\sum vote(x)},$$

where $x \in \{\text{top-5 posts}\}$ in the thread and $vote(x)$ gives the helpfulness score of post x. $objectivity$ is unit bound $[0, 1]$. While a high objectivity score indicates skewed helpfulness distribution in a thread, a low score indicates that there are multiple helpful answers in a thread; in other words, the thread is less objective in nature.

We analyze the cumulative distribution functions (CDFs) of objectivity spread scores for all threads belonging to the grey or blue wedge of Figure 5.4 (*cf.* Figure 5.5). We observe that the CDF for our model (blue) gives lower objectivity scores with 80^{th} percentile score of 0.64 for our model and 0.72 for BiLSTM, respectively. This indicates that our model performs better when the thread is more open-ended in nature.

5.5 Conclusion

We studied the problem of predicting helpfulness of posts in open-ended discussion forums. We found key differences in discussion forums compared to traditional CQA platforms: we observe that forum threads are often non-factoid and subjective in nature with many helpful answers. We hypothesize that post helpfulness crucially relies on two factors: (i) its relevance to the discussion thread and (ii) the novelty of the information introduced. We propose a generic and novel neural architecture using GRU encoders to embody this intuition. Our model outperforms state-of-the-art neural text classification baselines over a diverse set of forums representing three distinct domains. Through deeper analysis, we demonstrate that our model is able to encode the sequential nature of contextual posts, and capture the open-ended nature of discussion threads, thus achieving superior performance over other neural approaches. We plan to apply our work towards building a notification system for incoming helpful posts. In the current work, we addressed the information need

aspect present in the discussion forums in general. However, helpfulness might be conflated with other reasons e.g., humour, sentiment in certain domains. We would like to investigate those aspects in the future.

Chapter 6

Conclusion

Online discussion forums serve an important role by allowing people to learn from the collective wisdom of the community. However, the ever increasing number of new content being posted all the time makes it difficult for the forum users to find content suitable to their interest. In this thesis, we identify some of the key challenges in scaling the discussion forums and address some of them to better facilitate the discussions.

Firstly, we address discussion thread recommendation problem. We propose a probabilistic graphical model based solution which considers users' explicit and implicit interests to provide explainable recommendations. Using a two-stage framework called *IATM-JNCTR*, we show that the recommendation can be made aware of the users' underlying interests and cater threads based on that. Experiments with a large real-world discussion forum dataset show that our model is effective in addressing multiple issues associated with such thread recommendation scenario.

Next, we focus on the cold start problem in discussion forums stemming due to continuous influx of newly created threads. We treat it as an Extreme Multi-Label classification problem. We show that using a *cluster-sensitive attention* mechanism helps in dealing with long posts commonly found in open-ended discussion forums. We prove the effectiveness of our approach by experimenting with multiple large real-world forum datasets.

Finally, we propose solutions to improve the readability of individual threads. We propose an automatic solution that can identify the helpful posts out of many irrelevant or

repetitive posts within a thread. We perform experiments with discussion forum datasets from multiple domains and show the effectiveness of our models compared to off-the-shelf competitive methods. As our techniques are generic in nature, it could be applied to other domains involving text as long as the perceived helpfulness can be quantified semantically.

Facilitating discussion in online forums has a long way to go as the domain observes constant change in terms of technology, people, and all the happenings around the world. It would be interesting to study the temporal shifts in users' preferences, and if that can aid the recommendation process given user-thread interaction history for considerably long time. This thesis focuses on the information need of the forums users that drives their participation in these platforms. However, discussions in the online forums are often conflated with contemporary social behaviour such as sarcasm, trolling, memes, multimedia contents and so on. These open challenges would need research efforts from Machine Learning community such as NLP, vision as well as Social Sciences to understand human behavioural traits to make these platforms scale in an efficient and robust manner in the future. The lack of open feature-rich datasets in certain domains (*e.g.,* health) poses significant challenge in fostering research at scale. Such data are often restricted by legal clauses to preserve user-privacy as well as to serve the business models around health industry. The community has to come forward and think about new avenues to yield meaningful datasets while preserving important aspects like privacy, and public risk.

Bibliography

[1] D. Agarwal and B.-C. Chen. fLDA: Matrix Factorization through Latent Dirichlet Allocation. In *Proc. of WSDM*, pages 91–100, 2010.

[2] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proc. of WWW*, pages 13–24. ACM, 2013.

[3] M. Z. Asghar, A. Khan, F. M. Kundi, M. Qasim, F. Khan, R. Ullah, and I. U. Nawaz. Medical Opinion Lexicon: an Incremental Model for Mining Health Reviews. *International Journal of Academic Research*, 6(1):295–302, 2014.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] K. Balasubramanian and G. Lebanon. The landmark selection method for multiple output prediction. In *Proc. of ICML*, pages 283–290. Omnipress, 2012.

[6] T. Bansal, M. Das, and C. Bhattacharyya. Content Driven User Profiling for Comment-Worthy Recommendations of News and Blog Articles. In *Proc. of RecSys*, pages 195–202, 2015.

[7] A. Batenburg and E. Das. Emotional Coping Differences Among Breast cancer Patients from an Online Support Group: A Cross-Sectional Study. *Journal of Medical Internet Research (JMIR)*, 16(2):e28, 2014.

[8] A. Beloborodov, P. Braslavski, and M. Driker. Towards Automatic Evaluation of Health-Related CQA Data. In *Proc. of the International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 7–18. Springer, 2014.

[9] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[10] A. Beuchot and M. Bullen. Interaction and interpersonality in online discussion forums. *Distance Education*, 26(1):67–87, 2005.

[11] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *Proc. of NIPS*, pages 730–738, 2015.

[12] J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *Proc. of WWW*, pages 467–476, 2008.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[15] J. Carbonell and J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proc. of SIGIR*, pages 335–336, 1998.

[16] M. K. Chandrasekaran, M. Kan, B. C. Y. Tan, and K. Ragupathi. Learning Instructor Intervention from MOOC Forums: Early Results and Issues. In *Proc. of EDM*, pages 218–225, 2015.

[17] M. K. Chandrasekaran and M.-Y. Kan. When to reply? context sensitive models to predict instructor interventions in mooc forums. *arXiv preprint arXiv:1905.10851*, 2019.

[18] L. Charlin, R. S. Zemel, and H. Larochelle. Leveraging User Libraries to Bootstrap Collaborative Filtering. In *Proc. of KDD*, pages 173–182, 2014.

[19] X. Chen, M. Zhou, and L. Carin. The Contextual Focused Topic Model. In *Proc. of KDD*, pages 96–104, 2012.

[20] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can Cascades be Predicted? In *Proc. of WWW*, pages 925–936, 2014.

[21] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *Proc. of NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

[22] M. M. Cisse, N. Usunier, T. Artieres, and P. Gallinari. Robust bloom filters for large multilabel classification tasks. In *Proc. of NIPS*, pages 1851–1859, 2013.

[23] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and Diversity in Information Retrieval Evaluation. In *Proc. of SIGIR*, pages 659–666, 2008.

[24] E. Craig Sull. Tips for overcoming online discussion board challenges. *Faculty Focus*, September 2012.

[25] K. Halder, M.-Y. Kan, and K. Sugiyama. Health forum thread recommendation using an interest aware topic model. In *Proc. of CIKM*, pages 1589–1598. ACM, 2017.

[26] K. Halder, M.-Y. Kan, and K. Sugiyama. Predicting helpful posts in open-ended discussion forums: A neural architecture. In *Proc. of NAACL*, pages 3148–3157, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[27] K. Halder, L. Poddar, and M.-Y. Kan. Modeling Temporal Progression of Emotional Status in Mental Health Forum: A Recurrent Neural Net Approach. In *Proc. of WASSA*, pages 127–135, 2017.

[28] K. Halder, L. Poddar, and M.-Y. Kan. Cold start thread recommendation as extreme multi-label classification. In *Companion Proc. of WWW*, WWW '18, pages 1911–1918, 2018.

[29] X. He, T. Chen, M.-Y. Kan, and X. Chen. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *Proc. of CIKM*, pages 1661–1670, 2015.

[30] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proc of WWW*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[31] Z. He, M. S. Park, and Z. Chen. UMLS-Based Analysis of Medical Terminology Coverage for Tags in Diabetes-Related Blogs. *IConference 2016 Proceedings*, 2016.

[32] K. Hew and W. Cheung. *Student participation in online discussions: Challenges, solutions, and future research*. 05 2014.

[33] L. Hong and B. D. Davison. A Classification-based Approach to Question Answering in Discussion Boards. In *Proc. of SIGIR*, pages 171–178, 2009.

[34] D. J. Hsu, S. M. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *Proc. of NIPS*, pages 772–780, 2009.

[35] M. Huang, Y. Cao, and C. Dong. Modeling rich contexts for sentiment classification with lstm. *arXiv preprint arXiv:1605.01478*, 2016.

[36] A. Hutchinson. Reddit now has as many users as twitter, and far higher engagement rates. *Social Media Today*, Apr 2018.

[37] G. Jain, M. Sharma, and B. Agarwal. Spam detection on social media using semantic convolutional neural network. *International Journal of Knowledge Discovery in Bioinformatics (IJKDB)*, 8(1):12–26, 2018.

[38] M. Jenders, R. Krestel, and F. Naumann. Which Answer is Best?: Predicting Accepted Answers in MOOC Forums. In *Proc. of Workshop on Question Answering and Activity Analysis in Participatory Sites (Q4APS)*, pages 679–684, 2016.

[39] J. Jeon, W. B. Croft, J. H. Lee, and S. Park. A Framework to Predict the Quality of Answers with Non-Textual Features. In *Proc. of SIGIR*, pages 228–235, 2006.

[40] S. Kanthawala, A. Vermeesch, B. Given, and J. Huh. Answers to Health Questions: Internet Search Results Versus Online Health Community Responses. *Journal of Medical Internet Research (JMIR)*, 18(4):e95, 2016.

[41] A. Kapoor, R. Viswanathan, and P. Jain. Multilabel classification using bayesian compressed sensing. In *Proc. of NIPS*, pages 2645–2653, 2012.

[42] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. In *Proc. of RecSys*, pages 79–86, 2010.

[43] S. Kim. An Exploratory Study of User-centered Indexing of Published Biomedical Images. *Journal of the Medical Library Association (JMLA)*, 101(1):73–76, 2013.

[44] Y. Kim. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*, 2014.

[45] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2014.

[46] E. Kochkina, M. Liakata, and I. Augenstein. Turing at SemEval-2017 task 8: Sequential approach to rumour stance classification with branch-LSTM. In *Proc of (SemEval-2017)*, pages 475–480, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[47] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of SIGKDD*, pages 426–434. ACM, 2008.

[48] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. of SIGKDD*, pages 447–456. ACM, 2009.

[49] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of NIPS*, pages 556–562, 2001.

[50] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. In *Proc. of NIPS*, pages 556–562, 2001.

[51] J. Li, Y.-L. Theng, and S. Foo. Predictors of Online Health Information Seeking Behavior: Changes between 2002 and 2012. *Health Informatics Journal*, 22(4):804–814, 2016.

[52] X. Li and J. She. Collaborative variational autoencoder for recommender systems. In *Proc. of SIGKDD*, pages 305–314. ACM, 2017.

[53] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua. Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers. In *Proc. of SIGIR*, pages 283–292, 2013.

[54] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *ICLR*, 2017.

[55] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang. Deep learning for extreme multi-label text classification. In *Proc. of SIGIR*, pages 115–124. ACM, 2017.

[56] P. Liu, X. Qiu, Y. Zhou, J. Chen, and X. Huang. Modelling Interaction of Sentence Pair with Coupled-LSTMs. In *Proc. of EMNLP*, pages 1703–1712, 2016.

[57] C. Lo, J. Cheng, and J. Leskovec. Understanding Online Collection Growth Over Time: A Case Study of Pinterest. In *Proc. of WWW Industry Track*, pages 545–554, 2017.

[58] R. Magnezi, D. Grosberg, I. Novikov, A. Ziv, M. Shani, and L. S. Freedman. Characteristics of Patients Seeking Health Information Online via Social Health Networks versus General Internet Sites: A Comparative Study. *Informatics for Health and Social Care*, 40(2):125–138, 2015.

[59] Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, and M. Yoshikawa. Fast mining and forecasting of complex time-stamped events. In *Proc. of SIGKDD*, pages 271–279. ACM, 2012.

[60] J. D. Mcauliffe and D. M. Blei. Supervised Topic Models. In *Proc. of NIPS*, pages 121–128. 2008.

[61] A. Meier, J. E. Lyons, G. Frydman, M. Forlenza, and K. B. Rimer. How Cancer Survivors Provide Support on Cancer-Related Internet Mailing Lists. *Journal of Medical Internet Research (JMIR)*, 9(2):e12, 2007.

[62] T. Mihaylov and P. Nakov. Hunting for troll comments in news community forums. In *Proc. of ACL (Volume 2: Short Papers)*, volume 2, pages 399–405, 2016.

[63] A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. In *Proc. of NIPS*, pages 1257–1264, 2008.

[64] R. Nallapati, F. Zhai, and B. Zhou. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *Proc. of AAAI*, pages 3075–3081, 2017.

[65] T. V. Nguyen, A. Karatzoglou, and L. Baltrunas. Gaussian Process Factorization Machines for Context-aware Recommendations. In *Proc. of SIGIR*, pages 63–72, 2014.

[66] L. Nie, Y.-L. Zhao, M. Akbari, J. Shen, and T.-S. Chua. Bridging the Vocabulary Gap between Health Seekers and Healthcare Knowledge. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 27(2):396–409, 2015.

[67] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive language detection in online user content. In *Proc. of WWW*, pages 145–153, 2016.

[68] H. J. Oh, C. Lauckner, J. Boehmer, R. Fewins-Bliss, and K. Li. Facebooking for Health: An Examination into the Solicitation and Effects of Health-Related Social Support on Social Networking Sites. *Computers in Human Behavior*, 29(5):2072–2080, 2013.

[69] S. Oh and A. Worrall. Health Answer Quality Evaluation by Librarians, Nurses, and Users in Social Q&A. *Library & Information Science Research*, 35(4):288–298, 2013.

[70] S. Oh, Y. J. Yi, and A. Worrall. Quality of Health Answers in Social Q&A. *Proc. of AIST*, 49(1):1–6, 2012.

[71] A. Omari, D. Carmel, O. Rokhlenko, and I. Szpektor. Novelty based Ranking of Human Answers for Community Questions. In *Proc. of SIGIR*, pages 215–224, 2016.

[72] J. Palotti, L. Goeuriot, G. Zuccon, and A. Hanbury. Ranking Health Web Pages with Relevance and Understandability. In *Proc. of SIGIR*, pages 965–968, 2016.

[73] J. S. Pedro and A. Karatzoglou. Question Recommendation for Collaborative Question Answering Systems with RankSLDA. In *Proc. of RecSys*, pages 193–200, 2014.

[74] L. F. Pendry and J. Salvatore. Individual and social benefits of online discussion forums. *Computers in Human Behavior*, 50:211 – 220, 2015.

[75] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proc. of EMNLP*, 2014.

[76] L. Poddar, W. Hsu, and M. L. Lee. Author-aware Aspect Topic Sentiment Model to Retrieve Supporting Opinions from Reviews. In *Proc. of EMNLP*, pages 483–492, 2017.

[77] Y. Prabhu and M. Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proc. of SIGKDD*, pages 263–272. ACM, 2014.

[78] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. In *Proc. of EMNLP*, pages 248–256, 2009.

[79] S. Rendle. Factorization Machines with libFM. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):Article No. 57, 2012.

[80] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, and P. Blunsom. Reasoning about Entailment with Neural Attention. In *Proc. of ICLR*, 2016.

[81] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The Author-Topic Model for Authors and Documents. In *Proc. of UAI*, pages 487–494, 2004.

[82] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proc. of ICML*, pages 880–887, 2008.

[83] L. Schnitzler, S. K. Smith, H. L. Shepherd, J. Shaw, S. Dong, D. M. Carpenter, F. Nguyen, and H. M. Dhillon. Communication During Radiation Therapy Education Sessions: The Role of Medical Jargon and Emotional Support in Clarifying Patient Confusion. *Patient Education and Counseling*, 100(1):112–120, 2017.

[84] R. Seethamraju. Effectiveness of using online discussion forum for case study analysis. *Education Research International*, 2014, 2014.

[85] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proc. of AAAI*, 2016.

[86] A. Severyn and A. Moschitti. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proc. of SIGIR*, pages 373–382, 2015.

[87] C. Shah and J. Pomerantz. Evaluating and Predicting Answer Quality in Community QA. In *Procs. of SIGIR*, pages 411–418, 2010.

[88] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to Comment?: Recommendations for Commenting on News Stories. In *Proc. of WWW*, pages 429–438, 2012.

[89] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to comment?: Recommendations for commenting on news stories. In *Proc. of WWW*, pages 429–438. ACM, 2012.

[90] J. P. Singh, S. Irani, N. P. Rana, Y. K. Dwivedi, S. Saumya, and P. K. Roy. Predicting the "Helpfulness" of Online Consumer Reviews. *Journal of Business Research*, 70:346–355, 2017.

[91] M. D. Smucker, D. Kulp, and J. Allan. Dirichlet Mixtures for Query Estimation in Information Retrieval. Technical Report IR-445, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, 2005.

[92] I. Soboroff and D. Harman. Novelty Detection: The TREC Experience. In *Proc. of HLT-EMNLP*, pages 105–112. Association for Computational Linguistics, 2005.

[93] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proc. of CIKM*, pages 553–562. ACM, 2015.

[94] C. Sun, Y. Liu, C. Jia, B. Liu, and L. Lin. Recognizing Text Entailment via Bidirectional LSTM Model with Inner-Attention. In *Proc. of ICIC*, pages 448–457, 2017.

[95] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to Rank Answers on Large Online QA Collections. In *Proc. of ACL*, pages 719–727, 2008.

[96] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Proc. of NIPS*, pages 3104–3112, 2014.

[97] D. Tang, B. Qin, and T. Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proc. of EMNLP*, pages 1422–1432, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[98] A. Van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Proc. of NIPS*, pages 2643–2651, 2013.

[99] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[100] M. Volkovs, G. Yu, and T. Poutanen. Dropoutnet: Addressing cold start in recommender systems. In *Proc. of NIPS*, pages 4964–4973, 2017.

[101] C. Wang and D. M. Blei. Collaborative Topic Modeling for Recommending Scientific Articles. In *Proc. of KDD*, pages 448–456, 2011.

[102] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proc of SIGKDD*, pages 448–456. ACM, 2011.

[103] D. Wang and E. Nyberg. A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering. In *Proc. of ACL and IJCNLP*, pages 707–712, 2015.

[104] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proc. of SIGKDD*, pages 1235–1244. ACM, 2015.

[105] H. Wang, S. Xingjian, and D.-Y. Yeung. Collaborative recurrent autoencoder: recommend while learning to fill in the blanks. In *Proc. of NIPS*, pages 415–423, 2016.

[106] J. Wang, X. Hu, Z. Li, W. Chao, and B. Hu. Learning to Recommend Questions Based on Public Interest. In *Proc. of CIKM*, pages 2029–2032, 2011.

[107] X. Wang and Y. Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proc. of ACM International Conference on Multimedia*, pages 627–636, 2014.

[108] X.-J. Wang, X. Tu, D. Feng, and L. Zhang. Ranking Community Answers by Modeling Question-Answer Relationships via Analogical Reasoning. In *Proc. of SIGIR*, pages 179–186, 2009.

[109] Y.-C. Wang, R. Kraut, and J. M. Levine. To Stay or Leave?: The Relationship of Emotional and Informational Support to Commitment in Online Health Support Groups. In *Proc. of CSCW*, pages 833–842, 2012.

[110] J. Weston, A. Makadia, and H. Yee. Label partitioning for sublinear ranking. In *Proc. of ICML*, pages 181–189, 2013.

[111] S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The IBP Compound Dirichlet Process and its Application to Focused Topic Modeling. In *Proc. of ICML*, pages 1151–1158, 2010.

[112] B. Xu, J. Bu, C. Chen, and D. Cai. An Exploration of Improving Collaborative Recommender Systems via User-Item Subgroups. In *Proc. of WWW*, pages 21–30, 2012.

[113] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu. Detecting High-Quality Posts in Community Question Answering Sites. *Information Sciences*, 302(1):70–82, 2015.

[114] L. Yu, K. Moritz Hermann, P. Blunsom, and S. Pulman. Deep Learning for Answer Sentence Selection. In *Proc. of NIPS Deep Learning and Representation Learning Workshop*, 12 2014.

[115] M. Zhang, J. Tang, X. Zhang, and X. Xue. Addressing Cold Start in Recommender Systems: A Semi-supervised Co-training Algorithm. In *Proc. of SIGIR*, pages 73–82, 2014.

[116] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. In *Proc. of SIGIR*, pages 83–92, 2014.

[117] Y. Zhang and J. Schneider. Multi-label output codes using canonical correlation analysis. In *Proc. of AISTATS*, pages 873–882, 2011.

[118] T. C. Zhou, M. R.-T. Lyu, I. King, and J. Lou. Learning to Suggest Questions in Social Media. *Knowledge and Information Systems (KAIS)*, 43(2):389–416, 2015.

[119] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *Proc. of WWW*, pages 22–32, 2005.