A Topic and Concept Integrated Model for Thread Recommendation in Online Health Communities

Mingda Li
Department of Computer Science
New Jersey Institute of Technology
ml456@njit.edu

Weiting Gao
Martin Tuchman School of Management
New Jersey Institute of Technology
wg97@njit.edu

Yi Chen
Leir Research Institute
Martin Tuchman School of Management
New Jersey Institute of Technology
yi.chen@njit.edu

ABSTRACT

Online health communities (OHCs) provide a popular channel for users to seek information, suggestions and support during their medical treatment and recovery processes. To help users find relevant information easily, we present CLIR, an effective system for recommending relevant discussion threads to users in OHCs. We identify that thread content and user interests can be categorized in two dimensions: topics and concepts. CLIR leverages Latent Dirichlet Allocation model to summarize the topic dimension and uses Convolutional Neural Network to encode the concept dimension. It then builds a thread neural network to capture thread characteristics and builds a user neural network to capture user interests by integrating these two dimensions and their interactions. Finally, it matches the target thread's characteristics with candidate users' interests to make recommendations. Experimental evaluation with multiple OHC datasets demonstrates the performance advantage of CLIR over the state-of-the-art recommender systems on various evaluation metrics.

KEYWORDS

recommender systems, online health community, discussion forum, thread recommendation, neural network, Latent Dirichlet Allocation

ACM Reference Format:
Mingda Li, Weiting Gao, and Yi Chen. 2020. A Topic and Concept Integrated Model for Thread Recommendation in Online Health Communities. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411933

1 INTRODUCTION

Social media becomes more and more popular for people to learn about health matters. Seeking health information is the third most popular online activity (after using email and the search engine), where people seek advice, connect with experts and individuals with similar experience, discuss questions and concerns about treatment options, and so on [29]. According to a March 2018 Pew survey [36], health-related Facebook pages have more than 14.2 million followers in the past year. Online Health Communities (OHCs) are another type of social media designated for healthcare. An online health community is defined as an asynchronous online message board system that contains multiple message boards, each of which typically focuses on a single disease. OHCs provide a convenient channel for patients and caregivers to learn more about their current condition, share concerns about health problems, exchange information, as well as seeking and offering various kinds of peer support [34].

However, a big challenge that online users face when using Web for health matters is information overload. A study conducted by the Pew Internet & American Life Project reports that 36% of non-Internet users don't use Internet because they think it is not relevant to them. In addition, 32% of Internet users report difficulty in using Internet [49]. According to the "Law of Attrition" [12], users lose interest and stop using online health applications over time. Study shows that information overload makes it harder to keep users in OHCs [20]. Zhao's survival analysis finds that OHC users who start with a lot of information seekings may not keep engaged in the long run [44].

An effective recommender system that suggests relevant discussion threads to users is critical to address information overload problem and to help keep users engaged in OHCs. To develop a good recommender system, let us first analyze discussion threads. Table 1 shows a fragment of a sample discussion thread. We observe two dimensions in user interests in a post. One dimension is disease related, such as disease name, symptoms, diagnosis and treatment methods. This dimension represents the concepts being discussed. In this post, various terms are used, such as "breast cancer", "BC" and "mastectomy", to represent the cancer related concepts. The other dimension is topic related. Existing study shows top ten topics in breast cancer discussion forum including: emotion, lifestyle, diet, finance, family, communication with doctors, hair loss and regrowth, in addition to diagnosis, treatments and side effects [26] In this sample post, the discussion topic is related to finance.

There are many studies on content-based recommender systems, which apply natural language processing (NLP) techniques to extract user interests in either topic-dimension or concept-dimension. Some studies consider topic-dimension features using topic modeling techniques [2, 6, 42]. They propose to leverage Latent Dirichlet Allocation model (LDA) to discover latent topic distribution for matrix factorization-based collaborative filtering (CF). Other studies

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411933

Initial post:
I've been seriously considering having genetic testing done. I have a 13 year old daughter, and I'm very worried about her chances of getting breast cancer. Also, about a month ago, One of my cousins (who is 2 years younger than me) was diagnosed with BC. She was told she needed a mastectomy, and she opted for a double. Does anyone know how expensive the testing is, and if insurance covers it? Any info would be appreciated.
Reply post:
I too had the genetic testing done. Apparently, my insurance must have covered it because it was done a year ago and we never receive a bill for it.

Table 1: A Sample Thread on Breast Cancer

focus on concept-dimension and build content-based recommender systems using deep learning models. There are mainly two approaches. One is to encode the text using Gated Recurrent Units (GRU) or autoencoder and train neural network to predict user ratings for items [16, 27]. The other approach is Matrix Factorization (MF)-based recommender system. For instance, [22] integrates neural network (e.g., CNN) into probabilistic matrix factorization (PMF) to capture content. To the best of our knowledge, existing recommender sytems do not consider both topic dimension and concept-dimension of content.

In this paper, we present Convolution and LDA Integrated Recommendation (CLIR), a novel thread recommender system that models user interests in two dimensions: topic and concept, and also takes into consideration the interaction between the features from different dimensions. The topic-dimension is captured using Latent Dirichlet allocation (LDA) model [4], where multiple topics in a post are represented by a probability distribution. The concept-dimension is modeled by a Convolutional Neural Network (CNN) [9], where disease related words and phrases are encoded into a latent vector. To capture the interaction between these two dimensions, we create a user neural network and a thread neural network to integrate topic-dimension and concept-dimension features. In the user neural network, topic distribution and concept vectors are merged by fully connected layers to summarize the two-dimensional user interests based on the threads that the candidate user participated in the history. Similarly, in the thread neural network, the topic-dimension and concept-dimension features are integrated by fully connected layers to capture the content of target thread. Finally, we match content characteristics of the target thread and candidate users' interests with Multilayer Perceptron (MLP) in an interest matching network. The proposed CLIR system leverages content to identify relevant users, and is suitable for cold-start problem, where threads have no interaction history. The experimental evaluation demonstrates that CLIR outperforms the state-of-the-art content-based recommender systems on multiple OHCs datasets.

The contributions of our work include the following.

- To the best of our knowledge, this is the first attempt that combines the topic dimension and the concept dimension to capture user interests.
- A neural network model based on CNN and LDA is proposed to make thread recommendation, which captures both the topics dimension and the concept dimension to measure the relevance between the target thread's characteristics and candidate users' interests.
- Empirical evaluation on multiple OHC datasets demonstrates the effectiveness of CLIR.

The rest of the paper is organized as follows. We discuss the related work of OHC thread recommendation and content-based recommendation in Section 2. We formulate the thread recommendation problem in Section 3, and present the proposed thread recommender system CLIR in Section 4. The experimental results are presented in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

We first discuss the related work in thread recommendation in discussion forums, and then general content-based recommendation.

Thread Recommendation in Discussion Forums. Existing work that recommends relevant threads to target users in OHC are content based, considering either concept-dimension or topic-dimension when measuring the similarity between the candidate thread and the threads that a target user participates in the past [15, 16, 20]. Kishaloy Halder et al. [16] proposes a neural network architecture for thread recommendation using bi-directional Gated Recurrent United (GRU), capturing the concept dimension of user interests. The same research group also proposes another thread recommender system that captures topic dimensions [15]. It considers user's self reported medical conditions and user profiles as user interests, their symptoms as topics, then applies jointly normalized collaborative topic regression to capture the interaction between users and symptoms. Jiang et al. proposes to use heterogeneous healthcare information network to represent the concept dimension of OHCs data [20]. It extracts node-based and path-based features and trains a binary classification model for thread recommendation.

There are also studies on recommender systems for Community Question and Answering (CQA) forum such as Quora and Stackoverflow [17, 28, 40], or for Massive Open Online Courses (MOOCs) forum [21, 24, 32]. [40], [17], and [24] extract topic dimension features from thread content for recommendation. [40] considers a comprehensive set of features and applies the Learning-to-Rank algorithm to formulate recommendation as a ranking problem. [17] develops point process and neural network based algorithms for recommendation. [24] proposes a point-process based technique to make timely and topical thread recommendations.

Other factors are also explored in the literature. [21] considers social networks in recommendation, which is effective, but not available in cold-start setting. [32] uses sequential recommendation tasks to capture dynamic item sets and drifting user preferences. However, the method explores all possible sequences, which is not feasible for thread recommendation in OHC due to high sparsity. [28] considers not only users' expertise but also their willingness to answer a question.

Content-based Recommendation. In general, content-based recommendation profiles user interests based on their favorite content in the history and recommends relevant items according to users' interests. The items can be products [5, 7, 10, 33], articles [3, 22, 42] or events [46, 47].

Early work on content-based recommendation uses the collaborative filtering technique, which makes recommendations based on similarities between users and similarities between items [21]. Collaborative filtering may suffer from high sparsity of data (i.e. a lot of unknown relevance scores of user and item pairs) and cold start problem (new item or new user that does not appear in the training data). Matrix Factorization based models are also commonly used in recommender systems [5, 42, 46]. However, it often suffers from cold-start problem.

Topic models are also used in recommender systems. It regards documents as mixtures of latent topics with certain distributional properties [15]. The latent features of content is captured using LDA and its variants [4, 30, 38]. Wang et al. [42] proposes an extension of LDA, collaborative topic regression, which generates item latent factors by adding an offset latent variable to the document-topic distribution, for scientific article recommendation.

Recent studies use various deep learning models to capture concepts of content in recommendation. Among them, recurrent neural network (RNN)-based recommender systems [3] and convolutional neural network (CNN)-based recommender systems [14, 35] are two most frequently used approaches. There are also recommender systems that leveraging other deep learning methods, for example, multilayer perceptron [18], autoencoder [27], deep semantic similarity model [11], restricted boltzmann machine [13], neural autoregressive distribution estimation [48], and generative adversarial network [43]. Also there are studies that combine deep learning models with traditional recommendation techniques like matrix factorization (MF) [22]. [5, 22, 23, 46] propose recommender systems that combine Matrix Factorization based method and CNN. [8, 41] propose two RNN based deep learning models combined with traditional machine learning techniques.

In general, existing works on thread-recommendation or content-recommendation consider either topic or concept information of the item, but do not capture both dimensions, nor the interaction between two dimensions. To the best of our knowledge, this work is the first that considers both topic-dimension and concept-dimension in item description (thread content) and user interests, and is the first that combines the techniques of topic modeling and deep learning in recommender systems.

3 PROBLEM STATEMENT

We start with presenting the data model of OHC data. Then we define the OHC thread recommendation problem.

We model OHC data as two related sets: a set of threads $\mathcal{T}$ and a set of users who participate in threads $\mathcal{U}$. Each thread $t_j \in \mathcal{T}$ has a sequence of posts. We refer the first post in a thread as the initial post, and its author as the initial author of this thread. A user $u_i \in \mathcal{U}$ participates in a thread by either writing the initial post or reply a post in the thread. Figure 1 illustrates the OHC data structure. A reply relationship between two posts is represented by a solid line with an arrow pointing to the post being replied. A dashed line connects a post with its author.

A recommendation scenario includes users and items. In thread recommendation, threads are the items. In a thread, the initial

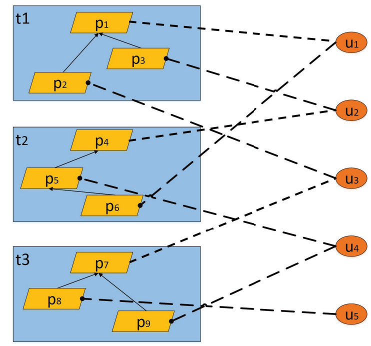

Figure 1: Data Structure of Online Health Community

Threads \ Users	u_1	u_2	u_3	u_4	u_5
t_1	0	1	1	0	0
t_2	1	0	0	1	0
t_3	0	0	0	1	1

Table 2: User Thread Interaction Matrix Presenting User Participation in Threads

post is considered as the item description and the reply action presents implicit relevance feedback from the users. For a thread, we consider all the users who participate in the thread discussions as the indication of their interests to this thread. We use an interaction matrix of the threads and users to capture the relevance relationship. For instance, Table 2 shows the interaction matrix of the threads and the users in Figure 1, where label 1 indicates relevance (i.e. a user replies the thread) and 0 means irrelevance.

For a given target thread, we recommend candidate users who are considered to be interested in the thread. We model this as a reply action classification problem. Given a target thread and a candidate user, we estimate the probability whether the candidate user will reply the target thread based on the history data of known user-thread reply relationships. The probabilities are used to create a ranked list of candidate users for recommendation.

*Definition 3.1 (**Problem Statement**).* Let $\mathcal{T}$ be a set of discussion threads, and $\mathcal{U}$ be a set of users in the history. The thread recommendation model is trained on the set of known user-thread reply relationships, represented as user-thread pairs with labels: $L(u_i, t_j) \in \{-1, 1\}$, for $\forall u_i \in \mathcal{U}, \forall t_j \in \mathcal{T}$, where 1 indicates that u_i replies in thread t_j, otherwise -1. The model predicts unknown label $L(u_i, t_j)$ for a candidate user $u_i \in \mathcal{U}$ and a target thread t_j, $t_j \notin \mathcal{T}$. That is, we focus on cold start problem, where the target thread does not appear in the training set.

4 PROPOSED METHOD

In this section, we first introduce the architecture of CLIR and then discuss the details of each module.

4.1 System Architecture

Figure 2 presents the architecture of CLIR. The input of the system is a target thread t_j, and a candidate user u_i along with her historical

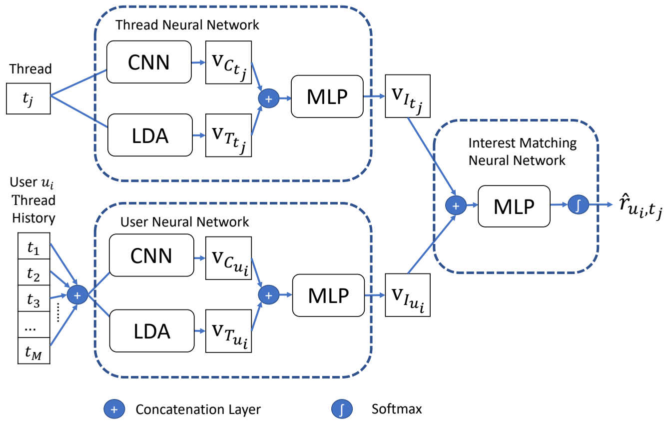

Figure 2: Overall Architecture of CLIR

threads before t_j, denoted as $\{t_1, t_2, \cdots, t_M\}$, from the oldest to the the most recent. In other words, target thread t_j is initiated after thread t_M. During the training time, CLIR builds a User Neural Network and a Thread Neural Network for all users and all historical threads. The User Neural Network captures the interests of each user based on their historically participated threads (presented by $\mathbf{v}_{I_{u_i}}$). The Thread Neural Network captures the characteristics of the target thread (presented by $\mathbf{v}_{I_{t_j}}$). Finally, the Interest Matching Neural Network calculates the likelihood ($\hat{r}_{u_i,t_j}$) of candidate user u_i to reply target thread t_j by matching $\mathbf{v}_{I_{t_j}}$ and $\mathbf{v}_{I_{u_i}}$.

As we discussed in Section 1, the characteristics of a thread is reflected in two dimensions. One dimension is the topics being discussed, such as diagnosis, lifestyle, family, insurance, cost, and so on. The other dimension is the concepts. In OHC, concepts are typically related to the medical conditions, such as disease names, treatments, and side effects. While both are important, there are two major differences between these two dimensions. First, typically there is only one topic in a single sentence; whereas a sentence may contain several words or phrases related to concepts. Second, a topic is typically captured by global features of a sentence; whereas a concept is captured by local features of the sentence, such as individual key terms. For example, consider a sentence, "Chemo costs lots of money." The term "Chemo" captures the medical condition of a patient. On the other hand, the topic of this sentence, "finance" is only captured by considering many words in the sentence or the whole sentence. Based on these observations, we propose to use different methods to capture the topic-dimension and the concept-dimension features, to be discussed in the Section 4.2 and 4.3, respectively.

A thread consists of a sequence of posts. In order to attract helpful replies, typically the initial post of each thread contains rich information for discussions, where a patient describes a problem and seeks for information and/or help. On the other hand, many reply posts contain irrelevant contents, like wishes, prays, or jokes, which may create noise for extracting thread characteristics. Therefore, we use the initial post's content to represent the thread's content in the discussions below.

4.2 LDA for Topics

We propose to summarize the topics discussed in OHC using a statistical topic modeling approach. As a statistical generative model, LDA (Latent Dirichlet allocation) is used to discover latent topics in documents as well as the words associated with each topic [4, 31, 39]. It is used for cancer-related dictionary construction and topic extraction in OHC [45]. In this study, we use LDA to generate topic distribution vectors to capture thread topics and user interest topics.

Given a set of threads $\mathcal{T}$, we train the LDA model. Suppose we aim to summarize L topics. After the training process, each thread $t_j \in \mathcal{T}$ is associated with a topic vector:

$$\mathbf{v}_{T_{t_j}} = (p_1^j, p_2^j, \ldots, p_L^j), \quad s.t. \sum_{l=1}^{L} p_l^j = 1 \tag{1}$$

Each dimension p_l^j of $\mathbf{v}_{T_{t_j}}$ stands for the probability of thread t_j belonging to the l^{th} topic, where $l \in \{1, 2, \ldots, L\}$. The sum of all dimensions is 1.

For a user $u_i \in \mathcal{U}$, the topic vector $\mathbf{v}_{T_{u_i}}$ is presented as the following.

$$\mathbf{v}_{T_{u_i}} = ({p'}_1^i, {p'}_2^i, \ldots, {p'}_L^i) \tag{2}$$

Each dimension ${p'}_l^i$ stands for the probability of user u_i's interest belonging to the l^{th} topic, where $l \in \{1, 2, \ldots, L\}$. To compute it, recall that user u_i participated in threads $\{t_1, t_2, \ldots, t_M\}$ before the target thread. Therefore, we add up the probabilities of these M historical threads belonging to the l^{th} topic, and then normalize the sum to be $[0, 1]$.

$${p'}_l^i = \frac{\sum_{j=1}^{M} p_l^j}{\sum_{l=1}^{L} \sum_{j=1}^{M} p_l^j} \tag{3}$$

Here p_l^j is the probability of thread t_j ($j = 1, 2, ..M$) belonging to the l^{th} topic. For example, consider a user with two historical threads, whose topic vectors are $(0.1, 0.9)$ and $(0.2, 0.8)$, respectively, assuming that there are two topics. The user's topic vector is calculated as $(0.15, 0.85)$, where $0.15 = \frac{0.1+0.2}{0.1+0.9+0.2+0.8}$ and $0.85 = \frac{0.9+0.8}{0.1+0.9+0.2+0.8}$.

4.3 CNN for Concepts

Now we discuss how to capture concepts in thread discussions. Different words in different threads may have similar meanings or potential connections. For example, a breast cancer forum user initiates a thread asking about the treatment "chemo". Another thread mentions purchasing "wig" to address the issue of hair loss, a side effect of chemo. Traditional bag-of-word model can only capture the similarity of documents with the co-occurrence of the same word, but fail to capture the semantic relationships among different words. To address this problem, word embedding techniques are proposed to capture the semantic relationships among words [37]. A pretrained feature vector is used to represent each word such that related words have their vectors close to each other in the vector space. Since OHC data is domain sensitive, we train the word vectors of OHC forum data using Keras package in Python [1].

[1] https://keras.io

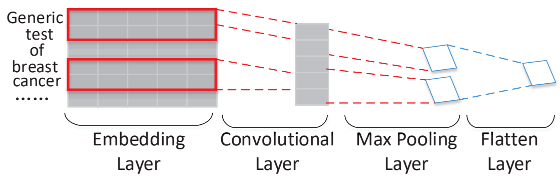

Figure 3: Model Architecture of Thread Convolution Model

The architecture of the thread convolution model is shown in Figure 3. To capture the pattern of different thread concepts, we first transform each English word into a low dimensional, continuous and real-valued vector. That is, the first layer of this neutral network is the embedding layer. The meaning of a longer expression comes from the meanings of its constituents and the rules used to combine them. To capture the potential semantic pattern in longer expressions, e.g., phrases or sentences, we use Convolutional Neural Network (CNN) [25] to extract the local features of threads in a fixed-width sliding window. We discuss the details below. Suppose the r^{th} word of an input thread is represented by a S-dimensional word embedding vector $\mathbf{x}_r \in \mathbb{R}^S$. Let $\oplus$ stand for the concatenation operation. $\mathbf{x}_{r:r+q}$ refers to the concatenation of words, $\mathbf{x}_r, \mathbf{x}_{r+1}, \ldots, \mathbf{x}_{r+q}$. Thus the input thread is represented as $\mathbf{x}_{1:R} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \ldots \mathbf{x}_R$ where R is the number of words in the input thread.

Taking $\mathbf{x}_{1:R}$ as input, the convolutional layer produces a feature map $\mathbf{c} = [c_1, c_2, \ldots, c_{R-H+1}]$. Each feature c_k in $\mathbf{c}$ is calculated by applying the filter $\mathbf{w} \in \mathbb{R}^{H \times E}$ on one of the sliding window positions per Equation 4.

$$c_k = g(\mathbf{w} \times \mathbf{x}_{k:k+H-1} + b) \tag{4}$$

Here H is the length of the sliding window and E is the number of filters. $b \in \mathbb{R}$ presents a bias term and g is a non-linear function such as the hyperbolic tangent ($tanh$) function.

Then a max pooling operation is applied over the feature map $\mathbf{c}$ to generate the concept vector $\hat{\mathbf{c}}$.

$$\hat{\mathbf{c}} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_{\frac{R-\hat{H}+1}{\hat{H}}}] \tag{5}$$

Each dimension $\hat{c}_{\hat{k}}$ is calculated as the max value in each sliding window, where $\hat{H}$ is the max pooling window size. Such values are considered to represent the important features in the feature map [26].

$$\hat{c}_{\hat{k}} = max\{c_{\hat{k}\times\hat{H}-\hat{H}+1}, c_{\hat{k}\times\hat{H}-\hat{H}+2}, \ldots, c_{\hat{k}\times\hat{H}}\} \tag{6}$$

Now we define the CNN model as a function $\mathcal{F}_{CNN}()$, whose input is the content of a thread and output is thread concept vector $\hat{\mathbf{c}}$ per Equation 5 and 6. That is, given a thread $t_j \in \mathcal{T}$, its concept vector $\mathbf{v}_{C_{t_j}}$ is calculated in Equation 7.

$$\mathbf{v}_{C_{t_j}} = \mathcal{F}_{CNN}(t_j) \tag{7}$$

For each user $u_i \in \mathcal{U}$, we generate the concept interest vector $\mathbf{v}_{C_{u_i}}$ by concatenating the content of historical threads that she participated in and feeding it to the convolution model.

$$\mathbf{v}_{C_{u_i}} = \mathcal{F}_{CNN}(t_1 \oplus t_2 \oplus \cdots \oplus t_M) \tag{8}$$

Here $t_1, t_2, \ldots, t_M$ are historical threads that user u_i participated in, and $t_1 \oplus t_2 \oplus \cdots \oplus t_M$ presents the concatenation of their content.

4.4 Thread Neural Network

We have discussed for each thread $t_j \in \mathcal{T}$, how to summarize its topics using LDA as a topic vector $\mathbf{v}_{T_{t_j}}$ (Equation 1) and capture its concepts using CNN as a concept vector $\mathbf{v}_{C_{t_j}}$ (Equation 7). Now we discuss how to build the Thread Neural Network that generates thread interest vector $\mathbf{v}_{I_{t_j}}$ to summarize thread characteristics from the topic vector $\mathbf{v}_{T_{t_j}}$ and concept vector $\mathbf{v}_{C_{t_j}}$.

We observe that the same word or phrase may have different meanings among forums that focus on different diseases, and indicate different user interests under different topics. For example, under breast cancer topic, "hair loss" means candidate user is interested in knowing about side effect. However, under a skin disease forum, it implies user interests in symptoms. In other words, "hair loss" interacts with "cancer" and enhances the topic of "side effect". For the latter case, "hair loss" interacts with "skin" and emphasizes topic "symptom". Multi-layer perception (MLP), which is composed of three or more layers of non-linearly-activating nodes, is suitable to simulate the interactions between different concepts and topics. Thus we integrate the topic-dimension and the concept-dimension using fully connected layers to capture the interaction between these two dimensions. In this layer, a MLP with Relu activation function is adapted to merge the topic vectors and concept vectors. Given a thread $t_j \in \mathcal{T}$, its thread interest vector $\mathbf{v}_{I_{t_j}}$ is generated by feeding the concatenation of thread topic vector $\mathbf{v}_{T_{t_j}}$ and concept vector $\mathbf{v}_{C_{t_j}}$ (Equation 1 and 7, respectively) to the MLP layer in the Thread Neural Network.

$$\mathbf{v}_{I_{t_j}} = Relu(\mathbf{w}_I \times [\mathbf{v}_{C_{t_j}} \oplus \mathbf{v}_{T_{t_j}}] + \mathbf{b}_I) \tag{9}$$

where $\mathbf{w}_I$ is a weight matrix and $\mathbf{b}_I$ is a bias vector.

4.5 User Neural Network

Now we discuss how to build the User Neural Network which summarizes each user $u_i \in \mathcal{U}$'s interests by generating user interest vector $\mathbf{v}_{I_{u_i}}$ from user topic vector $\mathbf{v}_{T_{u_i}}$ (Equation 2) and user concept vector $\mathbf{v}_{C_{u_i}}$ (Equation 8).

Similar as the Thread Neural Network, we use a MLP layer to capture the interaction between the topic-dimension and concept-dimension to generate the user interest vector $\mathbf{v}_{I_{u_i}}$:

$$\mathbf{v}_{I_{u_i}} = Relu(\mathbf{w'}_I \times [\mathbf{v}_{C_{u_i}} \oplus \mathbf{v}_{T_{u_i}}] + \mathbf{b'}_I) \tag{10}$$

where $\mathbf{w'}_I$ and $\mathbf{b'}_I$ are the weight matrix and the bias vector for user interest vector $\mathbf{v}_{I_{u_i}}$. $\mathbf{v}_{C_{u_i}}$ and $\mathbf{v}_{T_{u_i}}$ are calculated by Equation 8 and 2, respectively.

4.6 Interest Matching Neural Network

The Interest Matching Neural Network measures the similarity between user interest ($\mathbf{v}_{I_{u_i}}$ produced by the User Neural Network) and characteristics of thread ($\mathbf{v}_{I_{t_j}}$ generated by the Thread Neural Network).

Let us first consider an example: a candidate user is looking for threads relevant to the side effects of "chemo". An ideal thread to recommend should not only discuss the "side effect" symptoms, but also possible recovery methods for "hair regrowth". The connections between the "chemo" features expressed in the user interest vector and the features presenting "side effect" and "hair regrowth" in the thread characteristics vector should be captured to increase the similarity between candidate user's interest and target thread's characteristics.

We propose to capture the interactions between the user interest and the thread characteristics using fully connected layers. Specifically, we propose to use a neural network instead of Matrix factorization (MF) for interest matching. A neural network can capture the interaction between multiple features in a user vector and multiple features in a thread vector using a weighted sum. On the other hand, MF restricts each feature in a user vector to interact with a single feature in a thread vector. We propose to apply an MLP with a weight matrix $\mathbf{w}_M$ and a bias vector $\mathbf{b}_M$. Then a Relu activation function is adopted to learn the non-linear combination of the user interest features and thread characteristics features. The output vector $\mathbf{v}_M$ contains the interest matching information.

$$\mathbf{v}_M = Relu(\mathbf{w}_M \times [\mathbf{v}_{I_{u_i}} \oplus \mathbf{v}_{I_{t_j}}] + \mathbf{b}_M) \tag{11}$$

Here t_j is a target thread, which has a later timestamp than all historical threads that user u_i has participated in. $\mathbf{v}_{I_{t_j}}$ is thread t_j's interest vector generated by Equation 9 and $\mathbf{v}_{I_{u_i}}$ is user u_i's interest vector per Equation 10.

The probability of a user u_i replying a target thread t_j, $\hat{r}_{u_i,t_j}$, is calculated by passing $\mathbf{v}_M$ through a softmax activation function. Target thread t_j is recommended to a list of candidate users $u_i \in \mathcal{U}$, ranked by their $\hat{r}_{u_i,t_j}$ scores.

$$\hat{r}_{u_i,t_j} = softmax(\mathbf{v}_M) \tag{12}$$

4.7 Model Training

The model is trained on a set of known user-thread reply relationships. For each pair of candidate user u_i and target thread t_j in the training set, CLIR computes $\hat{r}_{u_i,t_j}$ (Equation 12), the predicted probability of user u_i replying t_j, based on the content of t_j and the content of historical threads $t_1, t_2, \ldots, t_M$ that u_i participated in. The training goal is to minimize the sum of the sum-of-squared-error terms between the actual reply action r_{u_i,t_j} and the predicted probability $\hat{r}_{u_i,t_j}$, and the L_2 regularized terms of the user interest vector $\mathbf{v}_{I_{u_i}}$ and the thread interest vector $\mathbf{v}_{I_{t_j}}$. Specifically, the loss function is defined as follows.

$$\mathcal{L} = \sum_{u_i,t_j} (r_{u_i,t_j} - \hat{r}_{u_i,t_j}) + \lambda' \sum_{u_i} \|\mathbf{v}_{I_{u_i}}\|^2 + \lambda \sum_{t_j} \|\mathbf{v}_{I_{t_j}}\|^2 \tag{13}$$

In Equation 13, λ' and λ stand for the L_2 regularization hyper-parameters for user interest vector $\mathbf{v}_{I_{u_i}}$ and thread interest vector $\mathbf{v}_{I_{t_j}}$, respectively. These two regularized terms are used for relieving over-fitting.

5 EXPERIMENTS

We evaluated the performance of CLIR on multiple real-world OHC datasets and compared it with the state-of-the-art recommender systems for discussion thread or content recommendation.

5.1 System Implementation

Using lda [2] package in Python, we trained the LDA model with 20 topics by fitting the topic model with 1,500 iterations. The topic number is consistent with the OHC data analysis proposed in literature [44]. For the input of user neural network, each topic probability is calculated based on Equation 3.

We implemented CLIR using Keras [3] library with TensorFlow [4] [1] as the backend. We trained word vectors in the thread convolution model for each dataset using the the top 20,000 frequent words in the vocabulary. The word vectors' dimension is set to be 128. We applied 32 filters with a sliding window of size 5. The pooling size of max pooling layer is 2 (1D pooling). This setting is shown to be effective for OHC data analysis in the literature [26]. The fully connected layers in user neural network, thread neural network, and interest matching network are followed by dropout layers with 0.25 dropout rate.

5.2 Datasets

To cover different disease types and different data characteristics, we use the following datasets for performance evaluation in our experiments.

Cancer Survivor Network (CSN) [5] is an online community for cancer patients, cancer survivors, their families and friends. We chose the breast cancer forum, which is the largest subforum in CSN, as test dataset. We collected 321 thousand posts with 1.9 million sentences, which are included in 25,208 discussion threads that are publicly available in this forum.

MedHelp [6] is created in 1994. Now it has over 12 million people discussing their health and medical problems and looking for information each month. The threads of one of its largest subforums, Heart Disease Community, are collected for our experiment. There are 22,826 threads including 51,408 posts with 7.7 million sentences.

HealthBoards [7] is a long-running support group website. We collected the threads from the Epilepsy and Fibromyalgia subforums. For Epilepsy, we collected 4,343 posts from 1,296 threads with around 0.5 million sentences. We also collected 13524 posts from 3121 threads with 2.5 million sentences from Fibromyalgia.

For the threads that are collected, we consider the ones that are related to patient decision making processes using existing work [26] since those threads contain rich information and suggestions related to medical-related problems. While threads that are not related to patient decision making may be jokes, advertisements, and fundraising campaigns, which brings noises. Following existing work [16], we removed threads which have less than 3 or greater than 100 replies to get rid of extremely off-topic threads or survey threads.

The statistics of the datasets are shown in Table 3. We collected data of five adjacent months and then split the data to the training set and the test set based on timestamp. Specifically, the threads that are initiated during the first four months are used for training the model. The threads initiated in the last month are used for testing

[2] https://pypi.org/project/lda/
[3] https://keras.io/
[4] https://github.com/tensorflow/tensorflow
[5] http://csn.cancer.org
[6] https://www.medhelp.org/
[7] https://www.healthboards.com/

the model's ability of recommending to relevant users. Since target threads do not appear in the training data, they are *cold start* threads.

5.3 Metrics

For each thread in the test set, each system ranks all users who have replied to at least one thread in the training data, and recommends the thread to the top M users. There may be users in the test set who have not replied to any thread in the training set. None of the comparison system nor CLIR is designed to make recommendations to such users, since their interests are unknown. Thus such users are excluded from the evaluation. The ground truth is determined by whether a user replied a thread, as implicit user feedback. If a user replied a thread, it means that the user is interested in the thread. Such a user-thread pair is considered as a positive instance. On the other hand, the fact that a user did not reply to a thread does not necessarily mean that the user is not interested in the thread. Instead, maybe it is because the user did not read the thread (but would have been interested in the thread if she reads it). We follow existing studies [16] [15] and evaluate system performance using Recall at top M, Normalized Discounted Cumulative Gain at top M, and Mean Reciprocal Rank.

Recall@M is the fraction of the users who indeed replied to a given thread that are recommended in the top M user list output by a system.

$$Recall@M = \frac{\#\ of\ replied\ users\ in\ top\ M\ list}{\#\ of\ replied\ users} \tag{14}$$

Normalized Discounted Cumulative Gain (*nDCG@M*) [19] is one of the most commonly used rank-aware metrics to evaluate recommender systems. It assigns different weights to results based on their relevance, and penalizes a system heavily if a result with higher relevance is ranked lower by the system. The score of the top M results output by a system $DCG@M$ is normalized by the ideal Discounted Cumulative Gain $iDCG@M$, which is score of the list of relevant results (ordered by their relevance) in the data up to position M.

$$nDCG@M = \frac{DCG@M}{iDCG@M} \tag{15}$$

where $DCG@M$ and $iDCG@M$ are calculated per Equation 16 and Equation 17.

$$DCG@M = \sum_{j=1}^{M} \frac{2^{r_j} - 1}{log_2(1+j)} \tag{16}$$

$$iDCG@M = \sum_{j=1}^{|REL_M|} \frac{2^{r_j} - 1}{log_2(1+j)} \tag{17}$$

Equation 16 calculates the DCG score on the ranked user list output by a system, whereas Equation 17 calculates the DCG score on the ground truth of user list in the order of their relevance, up to position M, REL_M. $r_j \in \{0, 1\}$ is the relevance of the user at position j. Here the relevance is binary, a user is either relevant (replied to the given thread) or is irrelevant (did not reply).

To measure the recommendation quality of the system-generated top M ranking list, we evaluated the system performance using Recall and nDCG with $M \in \{5, 10, 30, 50, 100\}$.

Mean Reciprocal Rank (*MRR*) The reciprocal rank of a recommender system is the multiplicative inverse of the rank of the first relevant result: 1 if the first relevant result in the system-generated ranking list is in the first place, $\frac{1}{2}$ for the second place, and $\frac{1}{3}$ for third place, and so on.

We calculated each of the above three metrics for every target thread and reported the average over all target threads.

5.4 Comparison Systems

Three state-of-the-art recommender systems using deep learning models are used for comparison purpose.

ConvMF [22] is a novel context-aware recommendation model that integrates convolutional CNN into probabilistic matrix factorization (PMF). ConvMF enhances the rating prediction accuracy by capturing the contextual information of documents. It has good recommendation performance in MovieLens and Amazon product review datasets.

TR-XMLC [16] considers the thread recommendation problem as one of supervised eXtreme Multi-Label Classification (XMLC) problem and addresses the problem by proposing a neural network architecture adapting bi-directional Gated Recurrent Units. It outperforms both the state-of-the-art recommender systems and other XMLC methods in thread recommendation on multiple discussion forums, such as Epilepsy, Fibromyalgia and Stackoverflow.

CVAE [27] is a collaborative variational autoencoder system that jointly models the generation of item content while extracting the implicit relationships between items and users collaboratively. It shows better performance than many state-of-the-art content-based recommender systems on datasets of academic articles and their citations.

5.5 Results and Analysis

The evaluation results of different models on each dataset are shown in Table 4 and 5. There are several points worth discussions. First, ConvMF does not work in this evaluation setting. It uses probabilistic matrix factorization (PMF), which requires a target thread have at least one user replied in the training set [22]. However, in our setting, target threads are cold-start threads, which do not have any user reply information. ConvMF fails to make any recommendation under this setting. Since it has 0 on all evaluation metrics, its performance is not shown in the tables.

Second, for *Recall@M* metrics (in Table 4), CLIR always outperforms other methods on Recall@5, Recall@10 and Recall@30, but sometimes has inferior performance than CVAE or TR-XMLC on Recall@50 or Recall@100. This indicates that most of the correct predictions of CLIR are located at the top of its ranking list. In contrast, CVAE and TR-XMLC may do better in including more relevant items down in the ranking list. A possible reason is that both CVAE and TR-XMLC not only capture user interests in their models but also social relationships when making thread recommendation. For instance, two users commonly reply to the same threads have similar user vectors. Indeed, in OHC, some users reply to their friends' posts to show their support, even though the

	CSN	Medhelp	Epilepsy	Fibromyalgia
#users	435	204	69	79
#threads	940	689	106	79
#user-thread pairs for training	329,295	108,732	6279	4345
#user-thread pairs for testing	79,605	31,824	1035	1896
sparsity	0.985	0.995	0.92	0.963

Table 3: Statistics of Four OHC Datasets

Datasets	Models	Metrics				
		Recall@5	Recall@10	Recall@30	Recall@50	Recall@100
CSN	TR-XMLC	0.092	0.093	0.170	0.253	0.594
	CVAE	0.053	0.090	0.258	0.452	**0.680**
	CLIR	**0.096**	**0.190**	**0.375**	**0.461**	0.523
MedHelp	TR-XMLC	0.243	0.288	0.434	0.438	0.693
	CVAE	0.046	0.075	0.091	0.219	0.718
	CLIR	**0.541**	**0.702**	**0.752**	**0.756**	**0.852**
Epilepsy	TR-XMLC	0.010	0.120	0.450	**0.659**	**1.000**
	CVAE	0.039	0.056	0.086	0.368	**1.000**
	CLIR	**0.098**	**0.170**	**0.451**	0.570	**1.000**
Fibromyalgia	TR-XMLC	0.180	0.307	0.629	0.770	**1.000**
	CVAE	0.097	0.118	0.218	0.712	**1.000**
	CLIR	**0.325**	**0.325**	**0.644**	**0.781**	**1.000**

Table 4: Recommendation Results of Recall@M with Different Methods

Datasets	Models	Metrics					
		nDCG@5	nDCG@10	nDCG@30	nDCG@50	nDCG@100	MRR
CSN	TR-XMLC	0.113	0.152	0.160	0.193	0.243	0.393
	CVAE	0.122	0.135	0.248	0.312	0.325	0.204
	CLIR	**0.228**	**0.238**	**0.285**	**0.317**	**0.340**	**0.405**
MedHelp	TR-XMLC	0.128	0.143	0.192	0.193	0.243	0.118
	CVAE	0.034	0.044	0.048	0.078	0.171	0.057
	CLIR	**0.467**	**0.529**	**0.546**	**0.547**	**0.564**	**0.568**
Epilepsy	TR-XMLC	0.029	0.071	0.184	0.225	0.327	0.149
	CVAE	0.022	0.031	0.041	0.121	0.274	0.056
	CLIR	**0.067**	**0.098**	**0.203**	**0.236**	**0.340**	**0.150**
Fibromyalgia	TR-XMLC	0.140	0.194	0.206	0.341	0.397	0.229
	CVAE	0.083	0.090	0.124	0.287	0.332	0.178
	CLIR	**0.230**	**0.233**	**0.261**	**0.342**	**0.410**	**0.314**

Table 5: Recommendation Results of nDCG@M and MRR with Different Methods

contents are not necessarily highly relevant to their interests. CLIR makes content-based recommendation, matching user interests with target thread content. Such cases impact the quality of the interest latent vectors generated by CLIR.

Third, CLIR demonstrates significantly better performance than all comparison systems on both *nDCG@M* and *MRR* metrics (Table 5), across different datasets and for different M values. The advantage is especially big for small Ms. As we discussed in Section 5.3, these two metrics evaluate the ranking performance of a system. This indicates that CLIR is more likely to rank relevant users high, and thus has the ability of identifying interested users to the target thread. In particular, TR-XMLC achieves a better performance than CVAE, leveraging the attention layer that identifies important words by giving them higher weights. CLIR considers not only words in the concept dimension, but also abstract topics in the topic dimension, thus better capturing the semantics of the content and achieving the best performance, to be illustrated in a case study. Furthermore, the performance advantage of CLIR is especially large for highly sparse data, such as Medhelp. There are many ad-hoc users, who may initiate threads asking for information, but rarely reply to other threads. CLIR well captures the user interests based on their recently initiated threads through concept and topic dimensions and makes good recommendation. On the other hand, TR-XMLC and CVAE consider user social relationships based on their replies to the same threads. Since there are limited common replies in the highly sparse data, these system often fail to capture users' interests to target threads.

CLIR considers both the concept and the topic dimensions, while TR-XMLC and CVAE consider only the concept-dimension of user interests. To gain insight about their recommendation, Table 6 shows a case study, with threads from CSN breast cancer forum. The first row shows a target thread in the evaluation. The other two rows show fragments of the threads that two users have participated in history. User A indeed replied to the the target thread, and user B didn't, in the data. The main topics and the terms representing concepts of each thread are summarized in the last two columns in the table. The recommendations made by the comparison systems with regard to these two users are presented in Table 7.

CLIR is the only system that makes the correct recommendation. Both user A and user B participated in historical threads with concepts relevant to the target thread, "shoulder pain". All three threads are relevant in terms of concept, shoulder pain. In terms of topics, all three threads discuss symptoms. On the other hand, the historical thread that user A participated in has additional topics on finance, which is also a main topic of the target thread. Thus, the target thread is more relevant to user A's interest than user B's. CLIR is the only system that considers both concept-dimension and topic-dimension, and is able to rank user A higher than user B.

From the evaluation results and the case study, we can see that with the integration of neural network and topic modeling, the proposed CLIR model captures both topic-dimension and concept-dimension of user interests and achieves overall better recommendation performance, especially on its top ranked results.

6 CONCLUSIONS

In this paper, we study thread recommendation problem in OHC. To address the problem, we consider both the topic-dimension and the concept-dimension to capture the content of a thread. We use LDA to summarize the topic-dimension and leverage CNN to encode the concept-dimension of threads. Then we build a thread neural network and a user neural network to generate the latent vectors to represent thread characteristics and user interests, respectively. Then, user interest vectors and thread characteristics vectors are matched using an interest matching network. Experiment results of multiple OHC datasets demonstrate the effectiveness of the proposed method. In future, we will incorporate social relationship features to further improve recommendation quality.

7 ACKNOWLEDGEMENT

This material is based on work partially supported by the Leir Foundation, a grant from the National Institutes of Health (UL1TR003017) and a Google Faculty Research Award.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*. 265–283.

[2] Deepak Agarwal and Bee-Chung Chen. 2010. fLDA: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining*. 91–100.

[3] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 107–114.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Honglong Chen, Jinnan Fu, Lei Zhang, Shuai Wang, Kai Lin, Leyi Shi, and Lianhai Wang. 2019. Deformable Convolutional Matrix Factorization for Document Context-Aware Recommendation in Social Networks. *IEEE Access* 7 (2019), 66347–66357.

[6] Xu Chen, Mingyuan Zhou, and Lawrence Carin. 2012. The contextual focused topic model. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 96–104.

[7] Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. 2018. ANR: Aspect-based neural recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 147–156.

[8] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 29–34.

[9] Judith E. Dayhoff. 1990. *Neural Network Architectures: An Introduction*. Van Nostrand Reinhold Co., New York, NY, USA.

[10] Aminu Da'u and Naomie Salim. 2019. Sentiment-Aware deep recommender system with neural attention networks. *IEEE Access* 7 (2019), 45472–45484.

[11] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.

[12] Gunther Eysenbach. 2005. The law of attrition. *Journal of medical Internet research* 7, 1 (2005), e11.

[13] Kostadin Georgiev and Preslav Nakov. 2013. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *International conference on machine learning*. 1148–1156.

[14] Yuyun Gong and Qi Zhang. 2016. Hashtag Recommendation Using Attention-Based Convolutional Neural Network.. In *IJCAI*. 2782–2788.

[15] Kishaloy Halder, Min-Yen Kan, and Kazunari Sugiyama. 2017. Health Forum Thread Recommendation Using an Interest Aware Topic Model. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1589–1598.

[16] Kishaloy Halder, Lahari Poddar, and Min-Yen Kan. 2018. Cold start thread recommendation as extreme multi-label classification. In *Companion Proceedings of the The Web Conference 2018*. 1911–1918.

[17] Patrick Hansen, Richard Junior Bustamante, Tsung-Yen Yang, Elizabeth Tenorio, Christopher Brinton, Mung Chiang, and Andrew Lan. 2019. Predicting the Timing and Quality of Responses in Online Discussion Forums. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1931–1940.

[18] Wenyi Huang, Zhaohui Wu, Liang Chen, Prasenjit Mitra, and C Lee Giles. 2015. A Neural Probabilistic Model for Context Based Citation Recommendation.. In *AAAI*. 2404–2410.

[19] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[20] Ling Jiang and Christopher C Yang. 2016. Personalized Recommendation in Online Health Communities with Heterogeneous Network Mining. In *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*. IEEE, 281–284.

[21] A Kardan, Amir Narimani, and Foozhan Ataiefard. 2017. A Hybrid Approach for Thread Recommendation in MOOC Forums. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering* 11, 10 (2017), 2195–2201.

[22] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 233–240.

[23] Donghyun Kim, Chanyoung Park, Jinoh Oh, and Hwanjo Yu. 2017. Deep hybrid recommender systems via exploiting document context and statistics of items. *Information Sciences* 417 (2017), 72–87.

[24] Andrew S Lan, Jonathan C Spencer, Ziqi Chen, Christopher G Brinton, and Mung Chiang. 2018. Personalized Thread Recommendation for MOOC Discussion Forums. *arXiv preprint arXiv:1806.08468* (2018).

[25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[26] Mingda Li, Jinhe Shi, and Yi Chen. 2019. Analyzing Patient Decision Making in Online Health Communities. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 1–8.

	Thread content	Topic	Concept
Target thread	I made it through uterine cancer. Diagnosed and had hysterectomy and radiation. Am 9 months past surgery now, but found lump in my breast and having shoulder pain. Is this a sign, or am I looking for problems were there aren't any? I'm in a bad place with no health insurance at the moment, so can't just go to my oncologist and get her opinion. It's a sad thing, to me at least, that getting paid is more of an issue than getting healthy.	symptoms, finance	uterine cancer, hysterectomy, radiation, lump, breast, **shoulder pain**, oncologist
User A (replied)'s historically participated thread	...... But every day after I noticed shoulder hurting it continued to hurt and sometimes have the pins/needles in it. I started checking everyday and the swelling has not gone down since. My daughter in law found a clinic in bigger town about 35 miles north of here that would work with me on payments. I guess I should mention I work only very part time and have no insurance. She went ahead and prescribed me SMZ/TMP 800-160mg. I was on that for 2 weeks. I then was on 500mg of Penicillin for another 2 weeks ...... Has anyone here had the swollen glands in arm pit for months- even though have taken antibiotics?	finance, treatments, symptoms	**shoulder hurting**, swelling, SMZ/TMP, Penicillin, swollen glands, antibiotics
User B (not replied)'s historically participated thread	I had a bilateral mastectomy in October with immediate reconstruction, followed by tissue expanders, then the replacement surgery in January. I completed six weeks of physical therapy for cording on my left arm (ulnar nerve) in May. I still have intense and constant shoulder pain, but the cording is gone. It is worse in the mornings, but doesn't improve much over the day. I have been told I will not develop lymphedema because I only had sentinel node biopsy and no radiation. But what is this shoulder pain? I can't seem to get an answer. Has anyone had this experience, would massage therapy help? If so, what kind? Any help would be appreciated.	symptoms, treatment	mastectomy, reconstruction, **shoulder pain**, lymphedema, biopsy

Table 6: Examples of a Target Thread and Two Users

	Recommended as a Top 5 User
User A	CLIR
User B	CVAE, TR-XMLC

Table 7: Recommendation of the Example in Table 6

[27] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 305–314.

[28] Jakub Macina, Ivan Srba, Joseph Jay Williams, and Maria Bielikova. 2017. Educational question routing in online student communities. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 47–55.

[29] Racheli Magnezi, Dafna Grosberg, Ilya Novikov, Arnona Ziv, Mordechai Shani, and Laurence S Freedman. 2015. 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 (2015), 125–138.

[30] Jon D Mcauliffe and David M Blei. 2008. Supervised topic models. In *Advances in neural information processing systems*. 121–128.

[31] Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. (2002).

[32] Fei Mi and Boi Faltings. 2017. Adaptive Sequential Recommendation for Discussion Forums on MOOCs using Context Trees. In *Proc. Intl. Conf. Educ. Data Min.* 24–31.

[33] Dongmin Hyun1 Chanyoung Park Min-Chul, Yang2 Ilhyeon Song2 Jung-Tae Lee, and Hwanjo Yu. 2018. Review Sentiment–Guided Scalable Deep Recommender System. (2018).

[34] Tareq Nasralah, Cherie Bakker Noteboom, Abdullah Wahbeh, and Mohammad Aref Al-Ramahi. 2017. Online Health Recommendation System: A Social Support Perspective. (2017).

[35] Hanh TH Nguyen, Martin Wistuba, Josif Grabocka, Lucas Rego Drumond, and Lars Schmidt-Thieme. 2017. Personalized Deep Learning for Tag Recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 186–197.

[36] Kenneth Olmstead Paul Hitlin. 2018. The Science People See on Social Media. https://www.pewresearch.org/science/2018/03/21/the-science-people-see-on-social-media/

[37] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[38] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 248–256.

[39] Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of latent semantic analysis* 427, 7 (2007), 424–440.

[40] Rohan Tondulkar, Manisha Dubey, and Maunendra Sankar Desarkar. 2018. Get me the best: predicting best answerers in community question answering sites. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 251–259.

[41] Bartłomiej Twardowski. 2016. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 273–276.

[42] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[43] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 515–524.

[44] Xi Wang, Kang Zhao, and Nick Street. 2014. Social support and user engagement in online health communities. In *International Conference on Smart Health*. Springer, 97–110.

[45] Y.-C. Wang, R. Kraut, and J. M. Levine. 2012. To stay or leave? The relationship of emotional and informational support to commitment in online heath support groups. In *ACM 2012 Conference on Computer Supported Cooperative Work (CSCW)*.

[46] Zhibo Wang, Yongquan Zhang, Honglong Chen, Zhetao Li, and Feng Xia. 2018. Deep user modeling for content-based event recommendation in event-based social networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1304–1312.

[47] Minghua Xu and Shenghao Liu. 2019. Semantic-enhanced and context-aware hybrid collaborative filtering for event recommendation in event-based social networks. *IEEE Access* 7 (2019), 17493–17502.

[48] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A Neural Autoregressive Approach to Collaborative Filtering. In *International Conference on Machine Learning*. 764–773.

[49] Kathryn Zickuhr. 2013. Who's not online and why. https://www.pewresearch.org/internet/2013/09/25/whos-not-online-and-why-2/