KETCH: Knowledge Graph Enhanced Thread Recommendation in Healthcare Forums

Limeng Cui
The Pennsylvania State University
University Park, PA, USA
lzc334@psu.edu

Dongwon Lee
The Pennsylvania State University
University Park, PA, USA
dongwon@psu.edu

ABSTRACT

Health thread recommendation methods aim to suggest the most relevant existing threads for a user. Most of the existing methods tend to rely on modeling the post contents to retrieve relevant answers. However, some posts written by users with different clinical conditions can be lexically similar, as unrelated diseases (e.g., Angina and Osteoporosis) may have the same symptoms (e.g., back pain), yet irrelevant threads to a user. Therefore, it is critical to not only consider the connections between users and threads, but also the descriptions of users' symptoms and clinical conditions. In this paper, towards this problem of thread recommendation in online healthcare forums, we propose a knowledge graph enhanced Threads Recommendation (KETCH) model, which leverages graph neural networks to model the interactions among users and threads, and learn their representations. In our model, the users, threads and posts are three types of nodes in a graph, linked through their associations. KETCH uses the message passing strategy by aggregating information along with the network. In addition, we introduce a knowledge-enhanced attention mechanism to capture the latent conditions and symptoms. We also apply the method to the task of predicting the side effects of drugs, to show that KETCH has the potential to complement the medical knowledge graph. Comparing with the best results of seven competing methods, in terms of MRR, KETCH outperforms all methods by at least 0.125 on the MedHelp dataset, 0.048 on the Patient dataset and 0.092 on HealthBoards dataset, respectively. We release the source code of KETCH at: `https://github.com/cuilimeng/KETCH`.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Personalization**.

KEYWORDS

Recommendation system, online health forum, graph neural networks, medical knowledge graph

ACM Reference Format:

Limeng Cui and Dongwon Lee. 2022. KETCH: Knowledge Graph Enhanced Thread Recommendation in Healthcare Forums. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532008

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532008

Table 1: Example threads on Diabetes (taken from MedHelp)

Query	Reply post
How low do I need to be in terms of blood sugars before I treat? I have been experiencing low blood sugars (sometime as low as 1.5 mmol/L)	You need to work out how much 1 g of glucose will raise you.
My DR. will allow me either premixed insulin70/30 or Lantus + humalog, which best? Decide insulin preference.	Premixed insulin requires you to eat on a fixed schedule, and you cannot easily change your insulin dose.

1 INTRODUCTION

The wide usage of social media has promoted the creation and sharing of user-generated contents. Health information seeking behavior has become more common as social media empowers the public by providing access to vast knowledge and improving their decision making [3]. According to the health tracking survey[1] from the Pew Research Center, information seeking on health subjects is one of the top online activities. Online health forum is a type of social media, where users share medical information by posting contents or comments in a collaborative and social manner. An online health forum such as Patient[2] usually contains several channels, each of which is related to a single medical condition or drug. However, with the increasing volume of posts on online health forums, users inevitably face the *information overload* problem, which can prevent users from obtaining needed information in a timely manner. Researchers find that online health forum users who have to initially spend substantial effort in seeking information may not stay engaged in the long run [35]. Therefore, thread recommendation is critical in online health forums. Given a user post (i.e., query on the left hand side) in Table 1, our goal is to return relevant solved threads (on the right hand side) to the user.

Unlike the thread recommendation problem in other domains such as online courses [21, 39] and community question and answer [40], however, *health thread recommendation has its unique challenges.* First, as the symptoms experienced by patients with different clinical conditions are often overlapping, it is difficult to capture each user's underlying cause (i.e., the disease or procedure). Many traditional approaches such as tensor factorization and topic model struggle to capture user intents behind, as they essentially use word co-occurrences to determine relevant answers [4, 32, 38, 39, 41].

[1]https://www.pewresearch.org/internet/2019/08/22/social-activities-information-seeking-on-subjects-like-health-and-education-top-the-list-of-mobile-activities/
[2]https://patient.info

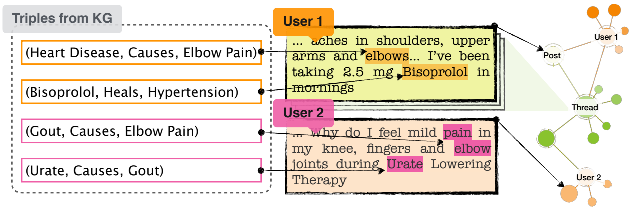


Figure 1: A heart health post from user 1, and lexically similar but unrelated post for gout from user 2 in another thread.

Current methods often adopt user profile data, such as subscription to a sub-forum related to a disease, as complementary information for recommendation [10, 22]. However, such information may not always be available in all forums. Second, existing health forum thread recommendation methods only consider the linear interactions between users and threads. To be specific, they only consider user-specific word usage patterns instead of learning a representation of user preference, while the latter is the core of personalized recommendation. The recommended threads should not only be lexically or semantically similar to user posts, but should also be of interest to users. For example, as shown in Table 1, though two users on the left both ask about diabetes medications before a meal, one is related to the concerns about low blood sugar and the other is about changing the medication. Thus, the two users interact with different threads on the right according to their own preferences. Considering this fact, it is far from enough to just measure the lexical and semantic similarity between user posts and threads from user-thread pairs. Instead, we should model the high-order interactions between them. Above all, we need to (1) explore the underlying cause behind the lexically similar user queries to recommend relevant answers; and (2) model the high-order interactions between users and threads to better characterize user preference.

To address the above two challenges, we introduce a medical knowledge graph to reveal potential causes behind a user query, and adopt the message passing idea to get better representations of user preference on threads. To be specific, a medical knowledge graph constructed from research papers and reports, can bridge the gap in users' posts by providing auxiliary information. From the example in Figure 1, we can infer that after being expanded by the triples from a knowledge graph such as (*Gout*, *Causes*, *Elbow Pain*), one post is related to heart health while the other is related to gout disease. Such latent clinical conditions can help provide more accurate answers to users. To get user preference on threads, we use the graph neural network (GNN) and model the interactions between users and threads. The users, posts, and threads are three types of nodes in a graph such that they are linked through their associations as shown in Figure 1. In other words, user nodes are connected to the thread nodes and post nodes in which they engage, and post nodes are connected to the corresponding thread nodes. The representation of a user node is learned by aggregating the message from their neighboring threads and posts, which reveals their interests. To our best knowledge, KETCH is the first method to model the high-order interactions between users and threads in online health forums.

In this paper, we propose a novel knowledge graph enhanced method for threads recommendation in online health forums, which can better capture connections between users and threads preference from both contextual and graph view. We propose a knowledge-enhanced text encoder to incorporate auxiliary information into the representations of threads and posts. We build a unified graph for user and thread, and compute node embeddings in the graph. The main contributions are threefold:

- We study a novel problem of thread recommendation in online health forums by leveraging knowledge graph to better capture user intents.
- We design a novel graph neural network based method KETCH (Knowledge Graph Enhanced Thread reCommendation in Healthcare Forums). In particular, we introduce the medical knowledge graph to capture user intents, and adopt the message passing idea to significantly enhance the learning of user and thread representations.
- We conduct extensive experiments on three real-world medical forum datasets to demonstrate the superiority of our method over several state-of-the-art methods. The reported results show that KETCH achieves a relative improvement of 0.125 on MedHelp dataset, 0.048 on Patient dataset and 0.092 on HealthBoards dataset comparing with the best results in terms of MRR. The case study shows the potential of KETCH in providing trustworthy and high-quality information.

2 RELATED WORK

This section briefly reviews two related topics: thread recommendation in online health forums and graph neural networks.

2.1 Thread Recommendation in Online Forums

A typical forum thread contains a query in its first post, and a discussion around it in subsequent posts. Thread recommendation is a special retrieval task with the first post as a query and the rest posts as a document [4]. It has been widely applied in online courses such as MOOCs [16, 21, 39] and community question and answer such as Quora and Reddit [13, 25, 30, 40, 42]. Existing methods can be broadly grouped into two categories: (1) the ones based on the topic similarity between the candidate threads and threads that user interacted in the past; and (2) the ones based on the network structure of users and threads.

Methods in the first category learn the embeddings of users based on their posts, and the recommend threads according to the post content. Halder et al. [10] propose an interest-aware topic model to align user's self-reported medical conditions and treatments for recommendation. CLIR [22] uses LDA and CNN to match the candidate thread's characteristics with users' interests to make recommendations. Hansen et al. [13] quantify both discussion post similarities and social centrality measures on a Stack Overflow dataset. Halder et al. [11] use BiGRU to capture the concept dimension of user interests for recommending threads to users. Above methods model user preferences on the global level of a post. However, it is intuitive that a user is interested in a particular part of a post such as symptoms according to their own medical conditions. In this work, we will model the user preference based on their interactions with threads and learn user-specific thread representations for recommendation.

The second group of methods learns the interactions between the users and threads. Jiang et al. [14] present the forum data as a heterogeneous information network to extract the path-based features of users and threads. Kardan et al. [16] map users' interactions to s social network and analyze the network to discover similar users for recommendation. However, these methods only consider the user-thread pair (two nodes with an edge) rather, while ignoring the high-order relations, which connect two nodes with multiple hops. Our model learns the high-order user-thread connectivity, which is essential for recommendation.

2.2 GNN based Recommendation

Graph Neural Networks (GNNs) refer to the neural network models which can operate on arbitrarily structured graphs [6, 7, 20, 31]. Due to their powerful capability of representation learning, GNNs have been widely applied in various domains such as computer vision [15, 28, 34], drug discovery[43], and chemistry [7, 9, 17]. Several extensions have been proposed to model the interactions between users and items for recommendation. These methods generally can be divided into two categories according to the use of side information. General recommendation methods without side information take the user-item bipartite graph as input, and propagate the information of nodes in the network. For example, a general inductive framework GraphSAGE [12] learns embeddings by sampling and aggregating features from a node's neighbors for citation recommendation.

For side information based approaches [8, 36, 37], researchers incorporate social information of users, such as the influence of friends. DiffNet [36] extends GraphSAGE by adding users' social relationships and behaviors. GraphRec [8] deploys a graph attention network on the users' social network graph and the user-item bipartite graph separately, and concatenates the learned node embeddings for social recommendation. Besides user's social relationship, DANSER [37] further considers the item-to-item relationship. Knowledge Graph (KG) can also be used as side information. Researchers take advantage of the rich information from KG to capture the potential connectivities between nodes. Wang et al. proposed KAGT [33] for product recommendation, which integrates the user-item bipartite graph and the KG into one, and adopts the attention mechanism to fully exploit the relationships between entities.

Hence in this paper, we introduce the KG into thread recommendation, aiming to improve recommendation performance in online health forums, and provide trustworthy and high-quality information simultaneously.

3 PROBLEM FORMULATION

In this section, we describe the notations and formulate the thread recommendation problem in online healthcare forums.

Given a forum thread set C, a post set $\mathcal{P}$ and a user set $\mathcal{U}$, a user $u_i \in \mathcal{U}$ published a post $p_k \in \mathcal{P}$ in thread $t_j \in C$. Based on this dataset, we treat the problem of thread recommendation as a specialized retrieval task with the first post of a thread as a query and each thread in our existing thread archive as a document. We estimate the probability of whether the user will reply to the target thread.

We denote $\mathcal{U}$ as a set of users $u_i \in \mathcal{U}$ with $i \in \{1, \ldots, N_u\}$, C as a set of forum threads $t_j \in C$ with $j \in \{1, \ldots, N_t\}$ and $\mathcal{P}$ as a set of posts $p_k \in \mathcal{P}$ with $k \in \{1, \ldots, N_p\}$. In order to model the complex interactions among the three types of nodes, we define the user-thread-post heterogeneous graph as follows.

Definition 1. ***User Graph**: The user graph is an undirected graph denoted as $\mathcal{G}_h = (\mathcal{U} \cup C \cup \mathcal{P}, \mathcal{I})$, where $\mathcal{I}$ is the set of interactions. When an interaction exists between two nodes (e.g., a user publishes a post), there will be an interaction to link the two nodes in the graph, otherwise none.*

The medical knowledge graph describes the entities collected from the medical literature, as well as relations (e.g., *Causes*) among entities. For example, $(Arthritis, Causes, Back\ Pain)$.

Definition 2. ***Medical Knowledge Graph**: Let $\mathcal{G}_m = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ be a medical knowledge graph, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ are the entity set, relation set and subject-relation-object triple set respectively. The triples are presented as $\{(e_h, e_r, e_t)|e_h, e_t \in \mathcal{E}, e_r \in \mathcal{R}\}$, which describes a relationship r from the head node e_h to the tail node e_t.*

Each post contains $|p|$ words, $p = \{w_1, w_2, \ldots, w_{|p|}\}$. We perform entity linking to build the word-entity alignment set $\{(w, e)|w \in \mathcal{V}, e \in \mathcal{E}\}$, where (w, e) means that word w in the vocabulary $\mathcal{V}$ can be linked to an entity e in the entity set. To capture the correlations of posts and entities in a medical knowledge graph, we define the post-entity bipartite graphs as follow.

Definition 3. ***Post-Entity Bipartite Graph**: The post-entity bipartite graph is denoted as $G_{pe} = (\mathcal{P} \cup \mathcal{E}, \mathcal{L})$, where $\mathcal{L}$ is the set of links. The link is denoted as $\{(p, Contains, e)|p \in \mathcal{P}, e \in \mathcal{E}\}$. If a post p contains a word that can be linked to entity e, there will be a link between them, otherwise none.*

The medical knowledge graph and the post-entity bipartite graph together can better capture latent user intent and unspecified medical conditions. Consider a user saying they have *Elbow Pain* during *Urate* lowering therapy. Two triples from the knowledge graph $(Urate, Causes, Gout)$ and $(Gout, Causes, Elbow\ Pain)$ provide a potential link between *Elbow Pain* and *Urate* and point out that the user may suffer from *Gout*. As such, it makes more sense to recommend this user with a post suggesting *Colchicine* as a treatment due to the triple $(Colchicine, Heals, Gout)$. Conversely, if the words between two different posts are not reachable in a knowledge graph, the two posts may be irrelevant. For example, although both "Colchicine" and "Bisoprolol" co-occur with "elbow pain" a lot, there is no strong connection between the two entities themselves from a medical perspective. However, existing recommendation methods may regard "Colchicine" and "Bisoprolol" as related. Hence, we argue that incorporating a medical knowledge graph can provide useful complementary information and yield higher accuracy in medical thread recommendation.

With the above notations and definitions, now we formulate the thread recommendation task in online healthcare foums as follows:

Problem 1 (Thread Recommendation in Online Healthcare Forums). *Given a thread set C, and a set of users $\mathcal{U}$, the goal is to assign a relevance score to each thread t in the collection C to a user u based on the history of user-thread relationships. The recommended* *threads should not only match the post contents but also fit the user's preference.*

4 THE PROPOSED METHOD: KETCH

Our proposed framework, KETCH, consists of three components, which is shown in Figure 2: 1) a knowledge-guided text encoder, which learns the entity embeddings in KG and incorporates the information into the thread textual representation; 2) a user preference encoder, which propagates the information between the user and thread nodes to learn user-specific preferences; 3) a prediction layer, which takes a user's post query as input and returns a set of threads in the archive with relevance scores. Then, we introduce the details of each module.

4.1 Knowledge-enhanced Text Encoding

In this section, we propose Knowledge-guided Text Encoding Layers to guide the embedding of user posts. To fully utilize the medical knowledge graph for healthcare forum post embedding, motivated by previous work [5], we leverage the inherent directional structure of a medical database to learn the entity embedding. To propagate the information from knowledge graph to user posts, we incorporate the Post-Entity Bipartite Graph and Medical Knowledge Graph into a unified relational graph, and add a set of self-loops (edge type 0) denoted as $S = \{(e, 0, e) | e \in \mathcal{E}\}$, which allows the state of a node to be kept. Hence, the new knowledge graph is defined as $\mathcal{G} = \{\mathcal{E}', \mathcal{R}', \mathcal{T}'\}$, where $\mathcal{E}' = \mathcal{E} \cup \mathcal{P}$, $\mathcal{R}' = \mathcal{R} \cup \{Contains, 0\}$ and $\mathcal{T}' = \mathcal{T} \cup \mathcal{L} \cup \mathcal{S}$.

Knowledge graph embedding is a way to map the relations and entities into a semantic space, while preserving the structure information of a graph. Here, we use TransR [24] to learn the embeddings. In TransR, for each triple (e_h, e_r, e_t), the entity embeddings $\mathbf{e}_h, \mathbf{e}_t \in \mathbb{R}^d$ are mapped into relation e_r's space via a projection matrix $\mathbf{W}_r \in \mathbb{R}^{d \times m}$. The relation embedding $\mathbf{e}_r \in \mathbb{R}^m$ builds a translation between projected entities by optimizing the score function, which is defined as:

$$f_r(h, t) = \mathbf{W}_r \mathbf{e}_h + \mathbf{e}_r - \mathbf{W}_r \mathbf{e}_t \tag{1}$$

The training of TransR considers maximizing the discrimination between correct triples and incorrect ones:

$$\mathcal{L}_{KG} = \sum_{(e_h, e_r, e_t) \in \mathcal{T}} \sum_{(e'_h, e_r, e'_t) \in \mathcal{T}^-} \lambda + f_r(e'_h, e'_t) - f_r(e_h, e_t) \tag{2}$$

where $(e'_h, e_r, e'_t) \in \mathcal{T}^-$ are incorrect triples constructed from correct triples $(e_h, e_r, e_t) \in \mathcal{T}$ by replacing entities and λ is a configurable margin.

Knowledge Propagation Net: Considering an entity e_h, we adopt the message passing idea [2] to model the message transferred from its neighboring nodes to the entity through the set of triples $\mathcal{T}_h = \{(e_h, e_r, e_t) | (e_h, e_r, e_t) \in \mathcal{T}'\}$:

$$\mathbf{e}^{(l)}_{\mathcal{T}_h} = \sum_{(e_h, r, e_t) \in \mathcal{T}_h} \delta^r_{ht} \mathbf{e}^{(l)}_t \tag{3}$$

where l is the depth and δ^r_{ht} indicates how much information being propagated from e_t to e_h in terms of relation r. Higher l will allow the information to propagate to higher-hop neighbors. We will explore how l will influence the performance in Section 5.4.2. δ^r_{ht} is calculated as follows:

$$\delta^r_{ht} = g(\mathbf{W}_r \mathbf{e}_h + e_r, \mathbf{W}_r \mathbf{e}_t) \tag{4}$$

where $g(\cdot)$ is a similarity function to measure the similarity of vectors. Different similarity functions can be applied here, e.g. cosine function.

We normalize the similarity scores across all triples connected with e_h:

$$\delta^r_{ht} = \frac{\exp(\delta^r_{ht})}{\sum_{(h,r,t') \in \mathcal{T}'} \exp(\delta^r_{ht'})} \tag{5}$$

Post Encoder: We use BiGRU [1] to encode the text sequence from both directions of words. Specifically, given the word embeddings $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{|p_k|}\}$ of a post p_k, the post embedding is computed as follows:

$$\begin{aligned} \overrightarrow{\mathbf{p}}_t &= \mathrm{GRU}(\overrightarrow{\mathbf{p}}_{t-1}, \mathbf{v}_t) \\ \overleftarrow{\mathbf{p}}_t &= \mathrm{GRU}(\overleftarrow{\mathbf{p}}_{t-1}, \mathbf{v}_t) \end{aligned} \tag{6}$$

We concatenate the forward hidden state $\overrightarrow{\mathbf{p}}_t$ and the backward hidden state $\overleftarrow{\mathbf{p}}_t$ as $\mathbf{p}_t = [\overrightarrow{\mathbf{p}}_t, \overleftarrow{\mathbf{p}}_t]$, which captures the contextual information of the post centered around word $\mathbf{v}_t$.

Different words in the same sentence may have different informativeness in representing users and threads. For example, in sentence "I'm with what has now become a chronic ectopic heart problem", the word "ectopic" is more important than the word "chronic" in representing this problem. Thus, we use the attention mechanism over word representations to learn informative post representations by aggregating the important word embeddings. The attention weight of the t-th word is computed as follows:

$$\begin{aligned} \mathbf{u}_t &= \tanh(\mathbf{W}_w \mathbf{p}_t + \mathbf{b}_w) \\ \alpha_t &= \frac{\exp(\mathbf{u}_t^{\mathrm{T}} \mathbf{u}_w)}{\sum_{t=1}^{|p_k|} \exp(\mathbf{u}_t^{\mathrm{T}} \mathbf{u}_w)} \end{aligned} \tag{7}$$

where $\mathbf{W}_w$, $\mathbf{b}_w$ and $\mathbf{u}_w$ are trainable parameters, $\mathbf{u}_t$ is a hidden representation of $\mathbf{p}_t$ obtained by feeding the hidden state $\mathbf{p}_t$ to a fully embedding layer.

The final representation of the post p_k is the aggregation of the contextual word representations weighted by their attention weights: $\mathbf{c}_k = \sum_{t=1}^{|p_k|} \alpha_t \mathbf{p}_t$.

Knowledge-aware Attention: To incorporate the graph structure into the textual representations, we replace the $\mathbf{u}_w$ in Eq. 7 by $\mathbf{u}'_w$ to get the final attention function:

$$\mathbf{u}'_w = \gamma \mathbf{u}_w + (1 - \gamma) \mathbf{W}_g \mathbf{e}_p \tag{8}$$

where $\mathbf{e}_p$ is the node embedding of a post p obtained from the Knowledge Propagation Net, W_g is a learnable transformation matrix and $\gamma \in [0, 1]$ is a trade-off parameter that controls the relative importance between two terms.

Thread Encoder: The thread encoder is used to learn representations of threads from the post representations. Similar to post encoder, we use BiGRU to encode each post. Given the post embeddings $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{|t_j|}\}$ in a thread t_j, we capture the contextual information in the post-level to learn the post representations $\mathbf{s}_k$ from the learned post vector $\mathbf{c}_k$.

Intuitively, not all posts can equally contribute to the representation. For example, in a thread, some ask for clarifications about the

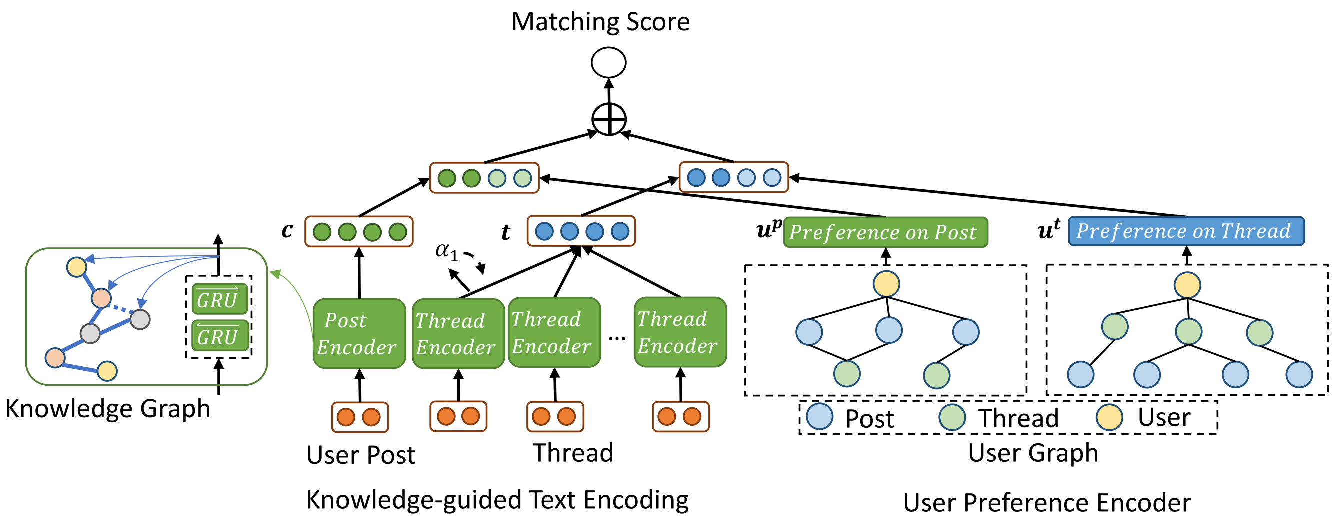


Figure 2: The framework of KETCH.

medical case to be discussed and the followings provide complementary information to the problem. Hence, we leverage the attention mechanism to learn the weights to measure the importance of each post as follows:

$$\mathbf{u}_v = \tanh\left(\mathbf{W}_c \mathbf{s}_k + \mathbf{b}_c\right)$$
$$\beta_v = \frac{\exp(\mathbf{u}_k^{\mathrm{T}} \mathbf{u}_c)}{\sum_{k=1}^{|t_j|} \exp(\mathbf{u}_k^{\mathrm{T}} \mathbf{u}_c)} \tag{9}$$

where $\mathbf{u}_k$ is a hidden representation of $\mathbf{s}_k$ obtained by feeding the hidden state $\mathbf{s}_k$ to a fully embedding layer, and $\mathbf{u}_c$ is a trainable parameter to guide the extraction of the context.

The thread representation $\mathbf{d}$ learned from these posts is computed as: $\mathbf{d}_j = \sum_{k=1}^{|t_j|} \beta_k \mathbf{s}_k$.

4.2 User Preference Encoding

The user preference Encoder is to learn the representation of user preference by modeling their interactions with threads and posts in the User Graph. For the same health thread, different users may be attracted its different parts of the content according to their own preferences. In this section, we model individual preference on threads and posts in order to conduct personalized recommendation.

User preference on threads: The threads that a user engaged with reflect their interests. In our model, a user u_i's preference on threads $\mathbf{u}_i^t$ is modeled by aggregating the incoming message from all threads $\mathcal{N}_i$ that they reply. In the User Graph, each node is assigned to an initial representation $\mathbf{h}^{(0)}$. The message transferred from a thread $t_j \in \mathcal{N}_i$ to the user u_i in the $(l+1)$-th layer is:

$$\mathbf{m}_{t_j \to u_i}^{(l+1)} = \mathbf{W}_t^u \mathbf{h}_j^{(l)} \tag{10}$$

where $\mathbf{m}_{t_j \to u_i}^{(l+1)}$ denotes the message from thread t_j to user u_i, $\mathbf{W}_t^u$ is a learnable weight parameter which maps the thread vector into the user embedding space. After the message passing step, we accumulate the incoming message at every user node by summing over all neighbors $\mathcal{N}_i$:

$$\mathbf{u}_i^t = \sigma\left(\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{m}_{t_j \to u_i}^{(l+1)}\right) \tag{11}$$

where $\sigma(\cdot)$ denotes an activation function (we use LeakyReLU in this paper).

User preference on posts: The user preference on posts $\mathbf{u}_i^p$ can be learned similar to the user preference on threads, by aggregating the message passing from all the post written $\mathcal{P}_i$. The message that a post $p_t \in \mathcal{P}_i$ passes to a user u_i in the $(l+1)$-th layer is defined as:

$$\mathbf{m}_{p_t \to u_i}^{(l+1)} = \mathbf{W}_p^u \mathbf{h}_t^{(l)} \tag{12}$$

where $\mathbf{W}_p^u$ is a learnable weight parameter which maps the post vector into the user embedding space.

Similar to Eq. 13, the user preference on posts is the aggregation of all the neighbor post information, which is defined as:

$$\mathbf{u}_i^p = \sigma\left(\frac{1}{|\mathcal{P}_i|} \sum_{t \in \mathcal{P}_i} \mathbf{m}_{p_t \to u_i}^{(l+1)}\right) \tag{13}$$

4.3 Model Prediction

The user-specific representations of posts and threads are the concatenation of the representations learned from the knowledge-enhanced text encoder and user preference encoder: $\mathbf{p} = [\mathbf{u}^p, \mathbf{c}]$ and $\mathbf{t} = [\mathbf{u}^t, \mathbf{d}]$. Hence, given a post query p_k of user u_i, the existing threads can be recommended to the users through the similarity score, which is calculated by the dot product of the user-specific representation of post and thread: $(\mathbf{p}_k)^{\mathrm{T}} \mathbf{t}_j$.

Following other recommendation methods [33], We use pairwise learning to train the model. For each post-thread pair, we add an unrelated thread $\mathbf{t}_j'$ to create a training triple $(\mathbf{p}_k, \mathbf{t}_j, \mathbf{t}_j')$. The loss function is formulated as:

$$\mathcal{L} = \sum_{(\mathbf{p}_k, \mathbf{t}_j, \mathbf{t}'_j) \in O} - \ln \sigma \left((\mathbf{p}_k)^{\mathrm{T}} \mathbf{t}_j - (\mathbf{p}_k)^{\mathrm{T}} \mathbf{t}'_j \right) \tag{14}$$

where O denotes the set of triples for training.

4.4 Model Training

Finally, we combine the pairwise learning loss $\mathcal{L}$ with KG loss $\mathcal{L}_{KG}$ to form the final objective function as follows:

$$\mathcal{L}_{final} = \mathcal{L}_{KG} + \mathcal{L} + \eta \, \|\Theta\|_2^2 \tag{15}$$

where Θ is the model parameters, and η is a regularization factor.

During the training, we optimize $\mathcal{L}_{KG}$ and $\mathcal{L}$ alternatively. We use Adam [19] to optimize the loss function. Adam can compute individual adaptive learning rates for different parameters, which has been widely used.

5 EXPERIMENTS

In this section, we present the experiments to evaluate the effectiveness of the proposed method. Specifically, we aim to answer the following evaluation questions:

- **EQ1**: Is KETCH able to improve thread recommendation performance by incorporating the medical knowledge graph?
- **EQ2**: How effective are knowledge graph and knowledge-aware attention, respectively, in improving the thread recommendation performance?
- **EQ3**: Can KETCH provide trustworthy and high-quality information to users?

5.1 Datasets

As the medical knowledge graph, we use a public medical knowledge graph KnowLife[3]. It is constructed from Web sources found in specialized portals and literature. There are 25,334 entity names and 591,171 triples. We use seven relations in KnowLife, including *Causes*, *Heals*, *CreatesRiskFor*, *ReducesRiskFor*, *Alleviates*, *Aggravates* and *HasSideEffect*.

To evaluate the performance, we gathered a collection of medical threads from three well-known online medical forums as follows.

- **MedHelp**[4] is created in 1994, now has over 12 million users discussing their medical issues and looking for advice. We collected data from their Diabetes and Heart Disease Communities for our experiment. There are 5,234 threads including 98,731 posts.
- **Patient** is a subsidiary of EMIS Health, first launched in 1996. We collected 25,212 threads with 253,206 posts from the Diabetes and Heart Disease forums on their website.
- **Healthboards**[5] is a long-running social networking support group website, which consists of over 280 Internet message boards for patient to patient health support. We use the dataset collected by Mukherjee et al.[6] [26].

The detailed statistics of the datasets are shown in Table 2. Following the existing work [11, 22], we removed threads that have less than 3 or more than 100 posts. Given a user's post and a candidate thread, the user-thread pair is positive if the user replied to the thread before; otherwise, it is negative.

[3]http://knowlife.mpi-inf.mpg.de/
[4]https://www.medhelp.org/
[5]https://www.healthboards.com/
[6]http://www.mpi-inf.mpg.de/impact/peopleondrugs/

Table 2: Statistics of datasets

Dateset	# Threads	# Posts
Medhelp	5,234	98,731
Patient	25,212	253,206
HealthBoards	5,584	1,048,576

5.2 Baselines

We compare KETCH with representative and state-of-the-art forum thread recommendation algorithms, which are listed as follows:

- **ConvMF** [18]: ConvMF integrates convolutional neural network into probabilistic matrix factorization for review recommendation on Amazon and MovieLens. It captures contextual information of documents to enhance the rating prediction accuracy.
- **AMF** [39]: AMF uses a matrix factorization framework to make thread recommendations to users.
- **PVLM** [32]: The authors design several linguistic features and calculate the similarity between questions and threads to retrieve corresponding threads.
- **XMLC** [11]: XMLC uses stacked BiGRU for text encoding along with cluster sensitive attention to find the correlations between users and threads.
- **CVAE** [23]: CVAE 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 performs well on datasets of academic articles and their citations.
- **CLIR** [22]: CLIR builds a thread neural network to capture thread characteristics and builds a user neural network to capture user interests. Then it matches the target thread's characteristics with candidate users' interests to make recommendations.
- **IATM+JNCTR** [10]: The authors use the topic model to capture the implicit interests embodied by users' textual descriptions in their profiles and how users interact with various symptoms.

Note that for a fair comparison, we choose above contrasting methods that use features from following aspects: (1) only **user-thread interactions**, such as ConvMF and AMF; (2) only **thread contents**, such as PVLM, XMLC; and (3) both **thread contents and user-thread interactions**, such as CVAE, CLIR, IATM+JNCTR.

5.3 Experimental Setup

To evaluate the performance of thread recommendation algorithms, we use the following metrics, which are commonly used in related work [10, 22]: Recall, NDCG (Normalized Discounted Cumulative Gain) and MRR (Mean Reciprocal Rank). To evaluate the top-k results returned by the recommendation system, we use Recall@k and NDCG@k with $k \in \{5, 10\}$.

We implement our model with Keras. We randomly split users into a training set (80%) and a test set (20%). Following the previous setting [10], users with fewer than five interactions always appear in the training set. We set the hidden dimension of our model to

128. The dimension of word embeddings is 100. We tested different learning rates of KETCH $lr = \{10^{-2}, 10^{-3}, 10^{-4}\}$ and depth $l = \{1, 2, 3, 4\}$. For baseline methods, we follow the network architectures as shown in the papers.

5.4 Thread Recommendation (EQ1)

To answer **EQ1**, we first compare KETCH with the representative recommendation algorithms introduced in Section 5.2 and then investigate the performance of KETCH when using different depths l.

5.4.1 Overall Comparison. Table 3 summarized the recommendation performance of all competing methods (reporting the average of 5 runs). From the table, we make the following observations:

- For user-thread interactions based methods, ConvMF and AMF, the performance is less than satisfactory. Both methods design a feature space to characterizes user interests to match threads and users. AMF performs slightly better than ConvMF as AMF represents the content of the thread as a bag of words. However, both methods can not handle long threads very well as they do not fully utilize the linguistic information of threads.
- For thread contents based methods, PVLM and XMLC perform better than those methods purely based on user-thread interactions, which indicates these methods can utilize the semantic and linguistic clues in the threads. XMLC can better capture the word and sentence embeddings by using the BiGRU structure.
- Moreover, methods using both user-thread interactions and thread contents, CVAE, CLIR, IATM+JNCTR and KETCH, perform comparable or better than those methods using either one of them. This indicates that user-thread interactions and thread contents can provide complementary information, which can be both beneficial to thread recommendation.
- Generally, among the methods using both user-thread interactions and thread contents, we can see that KETCH consistently outperforms other methods in terms of Recall@k, NDCG and MRR on three datasets.
- KETCH achieves a bigger improvement on Healthboards and MedHelp than that on Patient in terms of recall and MRR. We assume that it is because the average number of posts in a thread on Healthboards and MedHelp is larger than that of on Patient. In terms of NDCG, KETCH has significant improvement on Patient instead of Healthboards and MedHelp, because as for Patient, the average number of threads that a user replied to is larger. This allows KETCH to better capture the rich information of the threads and returns more related threads first.

5.4.2 Performance Comparison w.r.t. Depth in KG. We vary the depth l of KETCH to investigate how KG improves the performance on three datasets. The larger l allows the information to propagate to further nodes. We search l in the set of $\{1, 2, 3, 4\}$. As we did not get satisfying results on fourth- or higher-order interactions, we exclude those results. The results of Recall@10, NDCG@10 and MRR are listed in Figure 3. We can see that higher order of knowledge paths between entities can better improve the performance of KETCH. KETCH achieves the best results when $l = 3$.

5.5 Ablation Study (EQ2)

In order to answer **EQ2**, we explore each component of KETCH. We examine the components of the knowledge-enhanced text encoder and the user preference modeling by deriving several variants.

- **w/o Preference**: In this model, we simply use the knowledge-enhanced text encoder without user preference. This variant is designed to validate the effectiveness of user preference modeling in our model.
- **w/o KG**: This variant excludes the medical knowledge graph and knowledge-aware attention from the original model, and keeps the text encoder. We develop this variant to evaluate the necessity of the knowledge graph.
- **w/o Post Node**: This variant only considers two types of nodes, the thread and user nodes, in the user graph. We also remove the user preference on posts from the model. This variant is to investigate the effectiveness of modeling user preference on posts, as most existing methods only use user-thread pairs.

We summarize the results in Table 4 and have the following findings:

- When we solely use the text encoder without considering the user graph, the performance of KETCH largely degrades, which suggests the necessity of modeling user preference.
- Removing the knowledge graph and related knowledge-aware attention degrades the model's performance, as the attention mechanism only considers the semantic clues instead of the relationships between medical conditions. The KG can provide useful side information, especially for forums like Patient where most users are not medical professionals.
- When we do not include post nodes in the user graph, we have no way of knowing how much interaction the user has with threads. Users submit more replies to the threads that they are more interested in. Thus, only considering the user-thread pair can not fully capture the user preference information.

Through the ablation study of KETCH, we can conclude that (1) knowledge-enhanced text encoder can contribute to thread recommendation in online health forums; (2) it is important to capture user preference on both thread- and post-level.

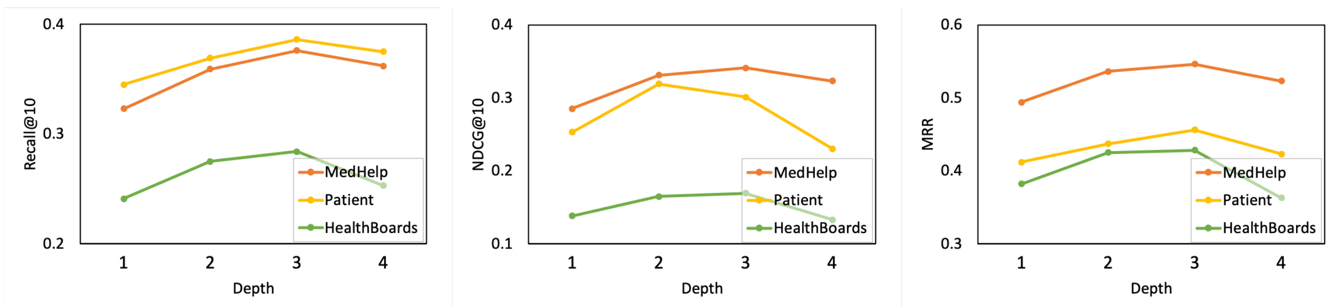
5.6 Case Study (EQ3)

In order to illustrate the potential of KETCH in providing trustworthy and high-quality information, we focus on two user studies: drug side-effect detection and user helpfulness prediction.

5.6.1 Drug Side-Effect Detection. The detection of drug side-effects is tightly linked to patient safety and pharmacovigilance. After the drug is on the market, it is very important to continuously improve and expand the drug list. Posts in online health communities often providing rich information about drug side-effects, such as "`Been on bisoprolol for 6 weeks now, 3.75mg does it affect breathing? I only seem to get the tiredness but sometimes it feels you have slight asthma`". In this section, we will explore whether the embeddings of drugs from KETCH can be used to identify the side-effects of drugs. We randomly delete some links between some drugs and their side-effects from KnowLife, and use this cut-down version of KG to train KETCH. Following

Table 3: Performance comparison on MedHelp, Patient and HealthBoards datasets.

Method	MedHelp					Patient					HealthBoards				
	Recall		NDCG		MRR	Recall		NDCG		MRR	Recall		NDCG		MRR
	@5	@10	@5	@10		@5	@10	@5	@10		@5	@10	@5	@10	
ConvMF	0.018	0.035	0.097	0.115	0.224	0.066	0.082	0.036	0.049	0.051	0.117	0.125	0.055	0.067	0.053
AMF	0.053	0.092	0.121	0.134	0.306	0.091	0.095	0.062	0.083	0.115	0.086	0.112	0.053	0.064	0.094
PVLM	0.119	0.130	0.125	0.126	0.355	0.168	0.185	0.142	0.158	0.214	0.155	0.164	0.097	0.121	0.286
XMLC	0.180	0.220	0.086	0.102	0.326	0.285	0.312	0.193	0.201	0.408	0.157	0.162	0.088	0.112	0.336
CVAE	0.156	0.175	0.042	0.114	0.329	0.152	0.177	0.125	0.135	0.320	0.134	0.224	0.139	0.157	0.254
CLIR	0.153	0.176	0.281	**0.334**	0.396	0.224	0.259	0.182	0.194	0.349	0.098	0.156	0.128	0.142	0.239
IATM+JNCTR	0.203	0.242	0.172	0.258	0.421	0.318	0.322	0.213	0.262	0.407	0.136	0.149	0.127	0.133	0.264
KETCH	**0.243**	**0.376**	**0.321**	0.331	**0.546**	**0.365**	**0.386**	**0.283**	**0.301**	**0.456**	**0.258**	**0.284**	**0.156**	**0.169**	**0.428**
Improvement (%)	19.70%	55.37%	14.23%	-0.90%	29.69%	14.78%	19.86%	32.86%	14.89%	11.76%	64.33%	73.17%	12.23%	7.64%	27.38%



(a) Recall@10 (b) NDCG@10 (c) MRR

Figure 3: Effects of the depth in KG.

Table 4: Ablation study of KETCH demonstrated the advantage of modeling user preference and KG.

Method	MedHelp					Patient					HealthBoards				
	Recall		NDCG		MRR	Recall		NDCG		MRR	Recall		NDCG		MRR
	@5	@10	@5	@10		@5	@10	@5	@10		@5	@10	@5	@10	
w/o Preference	0.115	0.128	0.258	0.296	0.313	0.264	0.273	0.143	0.164	0.362	0.125	0.151	0.131	0.156	0.259
w/o KG	**0.276**	0.347	0.313	0.326	0.503	0.327	0.351	0.206	0.242	0.465	0.158	0.172	0.144	0.162	0.394
w/o Post Node	0.225	0.359	**0.325**	0.329	0.528	0.359	**0.392**	**0.285**	0.296	**0.473**	0.238	0.261	0.138	0.153	0.415
KETCH	0.243	**0.376**	0.321	**0.331**	**0.546**	**0.365**	0.386	0.283	**0.301**	0.456	**0.258**	**0.284**	**0.156**	**0.169**	**0.428**
Improvement (%)	-11.96%	4.74%	-1.23%	0.61%	3.41%	1.67%	-1.53%	-0.70%	1.69%	11.76%	8.40%	8.81%	8.33%	4.32%	3.13%

Tang et al. [29], we take the embeddings from the Knowledge Propagation Net in Section. 4.1 and feed them into an MLP (Multi-Layer Perceptron) to get the probability that a link between a drug and a side-effect exists. We also test the results with TransR and GCN [20] as embedding methods.

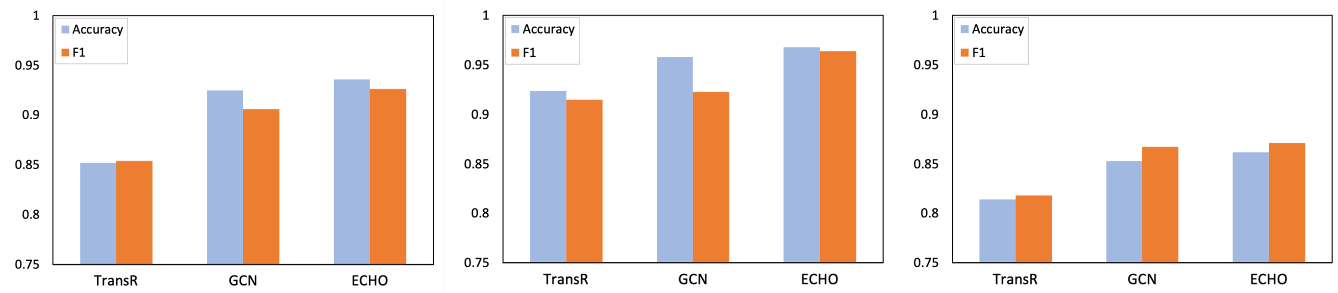
We test the above mentioned three embedding methods on MedHelp and Patient datasets. To evaluate the performance of drug side-effect detection, we use the following metrics, which are commonly used to evaluate classification performance: Accuracy and F1 Score. The results are shown in Figure 4. From the results, we can see that KETCH outperforms the two baseline models. It performs well in extracting information from the forum data.

5.6.2 User Helpfulness Prediction. In the second user study, we evaluate how well our model can predict # thanks that a post received in a community. We extract a subset of user posts in the dataset with various # of thanks. Our goal is to predict the # thanks received by post through its text embedding. To compare with other baselines such as CVAE and CLIR, we get the post embeddings from the pre-trained models and them into a Logistic Regression classifier. We run the LR classifier using scikit-learn [27] with default parameter settings.

Table 5: Performance comparison (RMSE) of predicting user helpfulness on MedHelp and Patient.

Method	MedHelp	Patient
CVAE	1.295	1.335
CLIR	0.918	1.256
KETCH	**0.835**	**1.117**

As HealthBoards does not have such information, we exclude HealthBoards from the user study. We use the # of thanks and whether the post is from a medical professional (1 and 0 are for from and not from) on Patient and MedHelp respectively. We use



(a) MedHelp (b) Patient (c) HealthBoards

Figure 4: Performance comparison of detecting drug side-effects on MedHelp, Patient and HealthBoards.

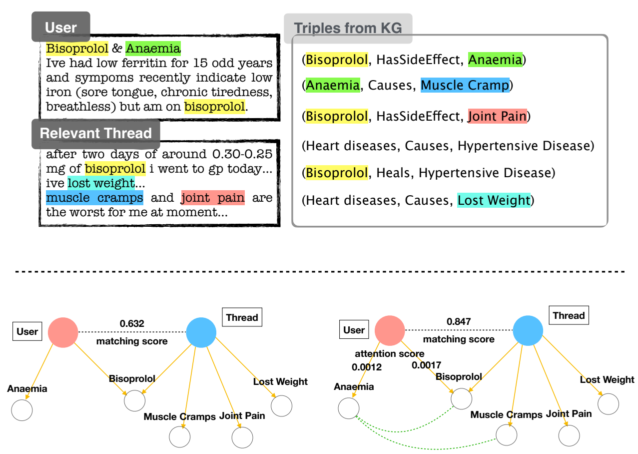


Figure 5: A real-world example visualizing how the triples from KG can contribute to the performance of the thread recommendation.

the metric RMSE to evaluate the results in Table 5. We can see that KETCH outperforms the baselines in predicting user helpfulness.

5.6.3 Visualization. In order to illustrate the importance of knowledge graph for helping health thread recommendation, we use an example to show the triples captured by KETCH in Figure 5. The user posted in a thread if anaemia is related to Bisoprolol. They also replied to another heart disease thread where another user reported the use of Bisoprolol and several symptoms. From the KG, we can infer that both users (1) suffer from related symptoms: anaemia and muscle cramp; (2) might suffer from the side-effects of Bisoprolol (anaemia and joint pain); (3) are using Bisoprolol to treat heart disease. Thus, the KG can give us more additional information than the word co-occurrence to determine the relevant threads.

To visualize the effects of the KG, we randomly select a user and one of its positive relevant thread samples in the test set. As shown in Figure 5, the triples from KG can build connections between the user and the thread. The left part shows a prediction without the help of KG. "Bisoprolol" is the only overlapping word, resulting in a relatively low matching score between the two. The right part shows a higher matching score between the two after introducing triples from KG. KG can build multiple connections between the two (see the green dashed lines as examples). Thus, by providing additional information, the thread can be recommended to the user with higher matching score.

In addition, "Bisoprolol" has higher attention weights to the texts. The related triples $(Bisoprolol, HasSideEffect, JointPain)$ and $(Bisoprolol, HasSideEffect, Anaemia)$ can explain why the user and the thread are related. We can see that KETCH can not only perform thread recommendation but also yields the explanations of the recommendation results.

6 CONCLUSION

In this paper, we propose KETCH, a knowledge graph enhanced thread recommendation in online healthcare forums. KETCH leverages additional information from a medical knowledge graph to guide the text embedding with a knowledge-aware attention mechanism. The network also adopts the message passing idea to capture user preference on both thread- and post-level in order to better represent user intents. We conduct extensive experiments on three real-world medical forum datasets to demonstrate the strong performance of our method over several state-of-the-art methods. We also use two case studies to show the potential of KETCH in completing knowledge graph and promoting trustworthy information to users.

This work aims at healthcare domain and uses a medical knowledge graph to enrich the connections. In subsequent work, we plan to extend KETCH to other thread recommendation problems, such as MOOCs recommendation. Besides, several interesting future directions need to be investigated. First, we can consider the hierarchical structure of posts under each thread for better embedding. On most forums, users can either reply to the first message (query) or to a specific post in the thread. The latter creates a lot of sub-threads under each thread. We sort the posts according to their timestamps, without considering sub-threads in this paper. Second, we can further incorporate user credibility, such as certified health workers, to recommend trustworthy information to users. On forums like MedHelp, a lot of verified medical professionals interact with users and answer questions. Their answers have higher credibility and quality.

ACKNOWLEDGMENTS

This work was in part supported by NSF awards #1940076, #1909702, and #1934782.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[3] Juan Chen, Xiaorong Hou, and Wenlong Zhao. 2016. Research on the model of consumer health information seeking behavior via social media. *International Journal of Communications, Network and System Sciences* 9, 8 (2016), 326–337.

[4] Jason HD Cho, Parikshit Sondhi, Chengxiang Zhai, and Bruce R Schatz. 2014. Resolving healthcare forum posts via similar thread retrieval. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. 33–42.

[5] Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. 2020. DETERRENT: Knowledge Guided Graph Attention Network for Detecting Healthcare Misinformation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 492–502.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 3844–3852.

[7] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'15)*. MIT Press, Cambridge, MA, USA, 2224–2232.

[8] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.

[9] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction Using Graph Convolutional Networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6533–6542.

[10] 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*. 1589–1598.

[11] 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.

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 1025–1035.

[13] 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.

[14] Ling Jiang and Christopher C Yang. 2016. Personalized Recommendation in online health communities with heterogeneous network mining. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, 281–284.

[15] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P Xing. 2019. Rethinking knowledge graph propagation for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11487–11496.

[16] 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.

[17] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* 30, 8 (2016), 595–608.

[18] 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*. 233–240.

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*. https://openreview.net/pdf?id=SJU4ayYgl

[21] Andrew S Lan, Jonathan C Spencer, Ziqi Chen, Christopher G Brinton, and Mung Chiang. 2018. Personalized thread recommendation for MOOC discussion forums. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 725–740.

[22] 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 & Knowledge Management*. 765–774.

[23] 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.

[24] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[25] 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.

[26] Subhabrata Mukherjee, Gerhard Weikum, and Cristian Danescu-Niculescu-Mizil. 2014. People on drugs: credibility of user statements in health communities. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 65–74.

[27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[28] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 2017. 3d graph neural networks for rgbd semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 5199–5208.

[29] Jizhi Tang, Yansong Feng, and Dongyan Zhao. 2019. Learning to Update Knowledge Graphs by Reading News. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2632–2641.

[30] 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.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[32] Jing Wang, Chuntao Man, Yifei Zhao, and Feiyue Wang. 2016. An answer recommendation algorithm for medical community question answering systems. In *2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE, 139–144.

[33] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.

[34] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6857–6866.

[35] 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.

[36] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.

[37] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. 2091–2102.

[38] Zhenlei Yan and Jie Zhou. 2012. A new approach to answerer recommendation in community question answering services. In *European Conference on Information Retrieval*. Springer, 121–132.

[39] Diyi Yang, Mario Piergallini, Iris Howley, and Carolyn Rose. 2014. Forum thread recommendation for massive open online courses. In *Educational Data Mining 2014*. Citeseer.

[40] Lei Yang and Xavier Amatriain. 2016. Recommending the World's Knowledge: Application of Recommender Systems at Quora. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 389–389.

[41] Thomas Zhang, Jason HD Cho, and Chengxiang Zhai. 2014. Understanding user intents in online health forums. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. 220–229.

[42] Stan Zhao, Meng Jiang, Quan Yuan, Bing Qin, Ting Liu, and ChengXiang Zhai. 2017. ContextCare: Incorporating Contextual Information Networks to Representation Learning on Medical Forum Data.. In *IJCAI*. 3497–3503.

[43] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 13 (2018), i457–i466.