Textual Tag Recommendation with Multi-tag Topical Attention

Pengyu Xu[a], Mingxuan Xia[a], Lin Xiao[a], Huafeng Liu[a], Bing Liu[a], Liping Jing[a,*], Jian Yu[a]

[a]*Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, No.3 Shangyuancun, Haidian District, 100044, Beijing, P. R. China*

Abstract

Tagging can be regarded as the action of connecting relevant user-defined keywords to an item, indirectly improving the quality of the information retrieval services that rely on tags as data sources. Tag recommendation dramatically enhances the quality of tags by assisting users in tagging. Although there exist many studies on tag recommendation for textual content, few of them consider two characteristics in real applications, i.e., the long-tail distribution of tags and the topic-tag correlation. In this paper, we propose a Topic-Guided Tag Recommendation (TGTR) model to recommend tags by jointly incorporating dynamic neural topic. Specifically, TGTR first generates dynamic neural topic that would indicate the tags by a neural topic generator. Then, a sequence encoder is used to distill indicative features from the post. To effectively leverage the topic and alleviate the data imbalance, we design a multi-tag topical attention mechanism to get a tag-specific post representation for each tag with the help of dynamic neural topic. These three modules are seamlessly joined together via an end-to-end multi-task learning model, which is helpful for the three parts to enhance each other and balance the effects of topics and tags. Extensive experiments have been conducted on four real-world datasets and demonstrate that our model outperforms the state-of-the-art approaches by a large margin, especially on

*Corresponding author

Email addresses: pengyuxu@bjtu.edu.cn (Pengyu Xu), 19121638@bjtu.edu.cn (Mingxuan Xia), xiaolin@bjtu.edu.cn (Lin Xiao), huafeng@bjtu.edu.cn (Huafeng Liu), 18271161@bjtu.edu.cn (Bing Liu), lpjing@bjtu.edu.cn (Liping Jing), jianyu@bjtu.edu.cn (Jian Yu)

tail-tags. The code, data and hyper-parameter settings are publicly released for reproducibility. [1]

Keywords: Information Retrieval, Recommendation System, Tag Recommendation, Neural Topic Model, Multi-task Learning
PACS: 0000, 1111
2000 MSC: 0000, 1111

1. Introduction

Tagging usually refers to the action of associating a relevant keyword or phrase with an item (e.g., document, image, or video) [1]. With the explosive growth of the Internet and consequent success of social network websites, tagging as a new concept for information expression is being used in many fields [2, 3]. Tag recommendation refers to the process in which the platform recommends several related tags to users when users tag an item [4]. Tag recommendation not only improves user experience but also may enrich the quality of the generated tags, indirectly improving the quality of the information retrieval services that rely on tags as data sources [5, 6, 7].

Generally, existing tag recommendation methods can be divided into collaborative filtering methods and content-based methods. Methods based on collaborative filtering take user's tagging history as input and aim to recommend tags in a personalized manner. E.g., Rendele et al. [8] regard user, item and tag as three dimensions of tensor respectively, and each entity represents whether the triplet has interacted. However, collaborative filtering methods only consider a fixed set of items and cannot recommend tags for new items. Therefore, it can not meet the current application scenarios that need to deal with a large number of cold-start items. In contrast, the content-based methods aim to recommend suitable tags by directly modeling the content [9] without using user interaction history. There exist studies towards various types of content, e.g., documents [10, 11, 12, 13], images [14, 15], songs [14], and micro-videos [16, 17]. The focus of this paper is on the textual content.

Traditional content-based methods [1, 18, 19] treat the document as "Bag of Words" (BoW), and design different tag ranking strategies. Limited by BoW feature, traditional methods are unable to capture the sequential or spatial information of textual content [9]. Recently, researchers have adopted

[1] `https://github.com/stxupengyu/TGTR`

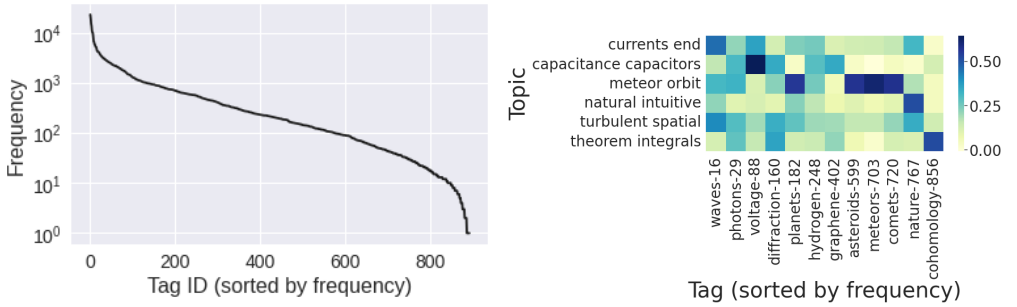
deep learning architecture to capture these crucial information. Gong and Zhang [11] adopt convolutional neural network (CNN) with an attention mechanism that consists of global and local channels. Considering the hierarchical structure of the document, Hassan et al. [13] use bidirectional gated recurrent unit (Bi-GRU) with a hierarchical attention mechanism for sentence encoder. Lei et al. [12] further introduce a capsule network to encode the intrinsic spatial relationship between the part and the whole. Recently, He et al. [20] leverage BERT-based pre-trained language models [21] for tag recommendation and achieve state-of-the-art (SOTA) performance.

These methods have achieved encouraging performance due to their ability to learn good representations of textual content. However, merely modeling the content is not enough for tag recommendation. We identify the following two key characteristics of tag recommendation for textual content in real applications, which offer new opportunities for further improvement.

Table 1: An example on the Q&A website Physics StackOverflow.

Post: Why did the June 2011 lunar eclipse last so long? It was kind of hard to miss the lunar eclipse this week, although I didn't see it in person. From what I understand it lasted about 100 minutes. I work that out as being about 9 minutes short of 1 degree of the Moon's orbit? How did it last so long? Surely 100 minutes is more than enough time for the Moon to move out of Earth shadow, or for Earth's shadow to "overtake" the Moon?
Tags: astronomy; moon; earth; eclipse
1^{st} **Topic:** moon; tide; tidal; shadow; moon's; ocean; lunar; night; eclipse; redshift

Long-tail Distribution. One important statistical characteristic is that tags usually follow a long-tail distribution. As shown in Figure 1 (a), a few tags are associated with a large number of instances, while a large fraction of tags are associated with a small number of instances. Only 15% of the tags have more than 1000 instances, while the 36% with below than 100 instances, which causes the challenge of the data imbalance on tail-tag recommendation. In this situation, it is not appropriate to learn a common post representation for all tags as in the previous approach [11, 13, 12, 20], since the head-tags



(a) Tag distribution in Physics

(b) Topic-tag correlation in Physics

Figure 1: The tag distribution and topic-tag correlation in Physics datasets. Here, tags are ordered by their frequency from the highest to the lowest in both subfigures.

can overwhelm the tail-tags. Therefore, we are motivated to leverage a tag-wise attention mechanism to get a tag-specific post representation for each tag that would alleviate the data imbalance.

Topic-tag Correlation. Each topic is a natural, coarse-grained tag, which indicates the tags [22]. We observe that many tags are highly correlated with their topic information. To illustrate the correlation, we take a post and its tags from Physics dataset in Table 1 as an example. After the topic obtained through neural topic model (NTM) [23], we show top-10 keywords in its most relevant topic. The tags "moon" and "eclipse" coincide with the topic's keywords, and the remaining tags "astronomy" and "earth" are also extended concepts of the topic. Figure 1 (b) furthermore illustrate the pairwise topic-tag correlation on Physics dataset. For each topic, the top-2 keywords are displayed. An interesting point is that there are strong correlations between topics and head-tags, as well as between topics and tail-tags. E.g., The head-tag "voltage" has strong correlation with the topic "capacitance capacitors", while the tail-tag "nature" also has strong correlation with the topic "natural intuitive". Thus, we are inspired to harness the valuable topic information to enhance tag recommendation. Instead of using topic directly, our approach lies in the adaptive usage of topic via the multi-tag topical attention mechanism, which makes each tag focused on a specific area of the post, thereby assisting tag recommendation.

Based on the above observations, we propose a Topic-Guided Tag Recommendation (TGTR) model to recommend tags by jointly incorporating dynamic neural topic. The main idea of TGTR is shown in Figure 2. Firstly,

A neural topic generator is applied to generate dynamic neural topic that would indicate the tags. Secondly, A sequence encoder is used to distill indicative features from the post. To effectively leverage the topic and alleviate the data imbalance, we design a multi-tag topical attention mechanism to get a tag-specific post representation for each tag with the help of dynamic neural topic. Then, a ranker is used to generate top-b predicted tags. Furthermore, our model is trained jointly on the above three modules, balancing the effect of topics and tags. We conduct extensive experiments on four real-world datasets to verify the rationality and effectiveness of our model. We summarize our main contributions as follows:

- A Topic-Guided Tag Recommendation (TGTR) model is proposed to recommend tags by jointly incorporating dynamic neural topic. To the best of our knowledge, our work is the first to integrate neural topic into tag recommendation in an end-to-end manner.
- In order to leverage the topic-tag correlation and alleviate the data imbalance, a multi-tag topical attention mechanism is specially designed, which can get a tag-specific post representation for each tag that would capture various intensive parts of the post through the guidance of dynamic neural topic.
- We conduct extensive experiments on four real-world datasets to demonstrate the superiority of our method over several state-of-the-art methods. Performance analysis on tail-tags is also conducted. In addition, we released our codes, datasets, and parameters to facilitate related researches.

The rest of the paper organized via four Sections. Section 2 discusses the related work. Section 3 describes the proposed TGTR model. The experimental setting and results are discussed in Section 4. A brief conclusion and future work are given in Section 5.

2. Related Work

Our work mainly relates to two fields: tag recommendation and topic model.

2.1. Tag Recommendation

The existing tag recommendation methods can be divided into two categories: collaborative filtering methods and content-based methods.

2.1.1. Collaborative Filtering Methods

Methods based on collaborative filtering take user's tagging history as input, and aim to recommend tags in a personalized manner. These techniques are primarily concerned with capturing the interaction between users, items, and tags. E.g., Rendele et al. [8] regard user, item and tag as three dimensions of tensor respectively, and each entity represents whether the triplet has interacted. After that, they proposed a personalized tag recommendation method based on pairwise interaction tensor decomposition [24]. Fang et al. [25] further introduce a nonlinear tensor factorization method based on Canonical Decomposition. Feng and Wang [26] model a social tagging system as a heterogeneous graph and recommended tags by learning the weights of nodes and edges in the graph. Shi et al. [27] exploit various types of relationships as features and design a topic-sensitive approach based on the factorization machines.

Deep learning approaches have recently been used to capture these interaction. Chen et al. [28] boost the classic pairwise interaction tensor factorization model by utilizing the graph neural networks. Sun et al. [29] propose a hierarchical attention model, which exploits two levels of attention to effectively aggregate different elements and different information. Zhao et al. [30] further model their interactive relationships in hyperbolic space to learn hierarchical data with lower distortion than Euclidean space. The combination of collaborative filtering method and content-based method is also studied [16, 17]. However, collaborative filtering methods only consider a fixed set of items and are unable to recommend tags for new items. Therefore, it can not meet the current application scenarios that need to deal with a large number of cold-start items (items without user information).

2.1.2. Content-Based Methods

Content-based methods treat all items as cold-start items and aim to recommend suitable tags by directly modeling the content [9] without using user's historical interaction. There exist studies towards various types of content, e.g., documents [10, 11, 12, 13], images [14, 15], songs [14], and micro-videos [16, 17]. The focus of this paper is on the textual content. Traditional content-based methods treat the document as "Bag of Words" (BoW), and design different tag ranking strategies. E.g., Song et al. [1] construct a bipartite graph of documents and tags. After graph clustering, the tags are sorted according to their importance. Xia et al. [10] propose a multi-label classifier based on Naive Bayes, and then uses the tag ranking

score predicted by the classifier. Wu et al. [18, 19] further consider the tag-content relevance phenomenon, using the probabilistic graphical models to simulate the tag generation process.

Limited by the BoW feature, traditional methods cannot capture the sequential or spatial information of textual content [9]. Recently, researchers have adopted deep learning architecture to capture these crucial information. Gong et al. [11] adopt a CNN with an attention mechanism that consists of global and local channels. Lei et al. [12] further introduce a capsule network to encode the intrinsic spatial relationship between the part and the whole. On the other hand, Li et al. [22] adopt a topical attention-based long short-term memory (LSTM) model that incorporates topic distributions generated by latent Dirichlet allocation (LDA) into sequential modeling. Considering the hierarchical structure of the document, [13] uses Bi-GRU with a hierarchical attention mechanism for sentence encoder. Tang et al. [9] propose a seq2seq method that jointly models sequential orders, tag correlation and content-tag overlapping. Recently, He et al. [20] leverage BERT-based pre-trained language models [21] for tag recommendation and achieve state-of-the-art performance.

2.2. Topic Model

A topic model is applied to a collection of documents and aims to discover a set of latent topics, each of which describes an interpretable semantic concept [31]. Well-known Bayesian probabilistic topic models (BPTMs), e.g., LDA [32] and Labeled LDA [33], have shown advantages in capturing effective semantic representations, and proven beneficial to varying downstream applications. However, like other Bayesian models, the learning of a BPTM is done by a Bayesian inference process (e.g. variational inference), which relies on the expertise involvement to customize model inference algorithms. Moreover, the inference processes for BPTMs can be hard to scale efficiently on extensive text collections. With the recent developments in deep generative models [34], Miao et al. [23] proposed neural topic model, to leverage deep neural networks (DNNs) to boost performance, efficiency, and usability of topic modeling. With appealing flexibility and scalability, NTM has been used in several NLP tasks, including text generation [35], document summarization [36], and so on. TLSTM [22] also adopts latent topic obtained by LDA model to enrich document representation for tag recommendation. Motivated by NTM and TLSTM, our model TGTR utilizes a neural topic generator (NTG) to generate dynamic neural topic that helps recommend

tags jointly in an end-to-end training stage. To the best of our knowledge, our work is the first to exploit the integration of the neural topic into tag recommendation in an end-to-end manner.

3. Proposed Method

Before we delve into the detailed model design, we first formulate the problems of tag recommendation in this article.

3.1. Problem Definition

Let calligraphic letter (e.g., $\mathcal{A}$) indicate set, capital letter (e.g. A) for scalar, lower-case bold letter (e.g., $\mathbf{a}$) for vector and capital bold letter (e.g., $\mathbf{A}$) for matrix. Tags are typically user-defined phrases or words that describe or categorize a specific content, such as an article, photo, or song [37]. Topic is often used to refer to a specific subject or theme that is discussed or analyzed within a corpus of textual data [32]. In general, tag can be regarded as a kind of fine-grained topic [22, 38]. We consider the tag recommendation problem for textual content. The input of the training stage includes N instances $\mathcal{P} = \{(\boldsymbol{X}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, each of which consists of a post $\boldsymbol{X}$ and several tags $\boldsymbol{y} = (y_1, ..., y_j, ..., y_l)$ related to the post. Here $y_j \in \{0, 1\}$, where $y_j = 1$ indicates that the j-th tag is associated with post $\boldsymbol{X}$, and l is the total number of candidate tags. Each post contains a sequence of words, $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_n)$, where $\boldsymbol{x}_i$ is a word (one-hot vector), and n is the post length. In topic modeling, we transform text sequence $\boldsymbol{X}$ into corresponding BoW feature $\boldsymbol{x}_{BoW} \in \mathbb{R}^{V_{BoW}}$, and V_{BoW} is the BoW vocabulary size. In the testing stage, we aim to recommend the most relevant tags for a new post which only contains a piece of textual content (i.e., without any known tags). For the ease of reading, the key symbols used in this paper are shown in Table 2.

3.2. Overall Framework

In this section, we describe our framework Topic-Guided Tag Recommendation (TGTR) model, as illustrated in Figure 2. We first provide an overview of our model and describe its main components. As we have said in the previous section, our model recommends tags by jointly incorporating dynamic neural topic. It consists of three modules: a neural topic generator, a sequence encoder and a topic-guided tag ranker. Firstly, we use a neural topic generator to get the dynamic neural topic $\boldsymbol{\theta}$ that would indicate the tags

Table 2: Notations used in the paper.

Symbols	Description
$\mathcal{P}$	Training set.
$\boldsymbol{X}$	The input post. $\boldsymbol{X} \in \mathbb{R}^{n \times V}$
$\boldsymbol{y}$	The tags related to the post. $\boldsymbol{y} \in \{0,1\}^l$
n	The length of post.
l	The total number of candidate tags.
$\boldsymbol{E}$	The embedding matrix. $\boldsymbol{E} \in \mathbb{R}^{V \times D_{embed}}$.
V	The vocabulary size.
$\boldsymbol{x}_{BoW}$	BoW feature of the post. $\boldsymbol{x}_{BoW} \in \mathbb{R}^{V_{BoW}}$.
V_{BoW}	BoW vocabulary size.
$\boldsymbol{\theta}$	Neural topic. $\boldsymbol{\theta} \in \mathbb{R}^K$.
K	Topic number.
D_{embed}	The size of low-dimensional embedding space.
D_{hidden}	The hidden size of Bi-GRU.
$\alpha_{k,j}$	Topic-guided attention score.
$\boldsymbol{r}_k$	The topic-guided post representation. $\boldsymbol{r}_k \in \mathbb{R}^{D_{hidden}}$.
δ	The hyper-parameter to balance joint training.

of the post based on BoW feature $\boldsymbol{x}_{BoW}$. Secondly, the sequence encoder is proposed to capture semantic and syntactic information in local consecutive word sequences $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$. Thirdly, in order to leverage the topic-tag correlation and alleviate the data imbalance, we design a topic-guided tag ranker which includes two parts. The multi-tag topical attention mechanism is designed to get a tag-specific post representation for each tag that would capture various intensive parts of the post through the guidance of dynamic neural topic. Then, the ranker is used to generate top-b predicted tags. We next discuss each step in detail.

3.3. Neural Topic Generator

In this section, we propose a neural topic generator to get the dynamic neural topic $\boldsymbol{\theta}$ that can represent the global information of the post based on BoW feature $\boldsymbol{x}_{BoW}$ and indicate the tags of the post. As shown in Figure 2, neural topic generator is based on variational autoencoder (VAE), that includes an inference model $q(\boldsymbol{\theta}|\boldsymbol{x}_{BoW})$ and a generative model $p(\boldsymbol{x}_{BoW}|\boldsymbol{\theta})$. VAE is a particularly natural choice for topic models, because it trains the

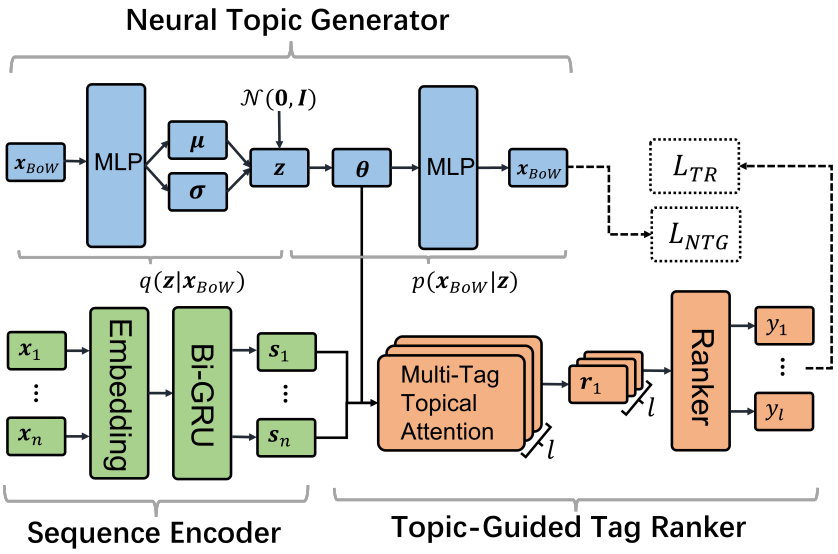


Figure 2: The architecture of TGTR.

inference model, a neural network that directly maps a document to an approximate posterior distribution [39]. Our neural topic generator combines the merits of both neural networks and traditional probabilistic topic models. They can be efficiently trained via backpropagation, scale to large datasets, and avoid overfitting [23]. Following the framework of neural variational inference [40], we construct an inference network conditioned on the observed post $\boldsymbol{x}_{BoW}$ to generate the variational parameters $\boldsymbol{\mu}(\boldsymbol{x}_{BoW})$ and $\boldsymbol{\sigma}(\boldsymbol{x}_{BoW})$ that are implemented by multilayer perceptrons (MLPs) $f_*(\cdot)$.

$$\boldsymbol{\mu}(\boldsymbol{x}_{BoW}) = f_\mu(\boldsymbol{x}_{BoW}) \tag{1}$$

$$\boldsymbol{\sigma}(\boldsymbol{x}_{BoW}) = f_\sigma(\boldsymbol{x}_{BoW}) \tag{2}$$

The generative model $q(\boldsymbol{x}_{BoW}|\boldsymbol{\theta})$ makes the following assumptions. There exist K topics, and each post $\boldsymbol{x}_{BoW}$ has a topic distribution $\boldsymbol{\theta}$ in the K-dimensional topic space. Inspired by neural topic model [23], our proposed generator introduces DNNs to parameterize the multinomial topic distribution. The generative process is:

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{3}$$

$$\boldsymbol{z} = \boldsymbol{\mu}(\boldsymbol{x}_{BoW}) + \boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}(\boldsymbol{x}_{BoW}) \tag{4}$$

$$\boldsymbol{\theta} = softmax(f_\theta(\boldsymbol{z})) \tag{5}$$

$$\boldsymbol{x}_{BoW} = softmax(f_\phi(\boldsymbol{\theta})) \tag{6}$$

Here we parameterize the multinomial document topic distributions ϕ by a MLP $f_\phi(\cdot)$ conditioned on a draw $\boldsymbol{z}$ from a Gaussian distribution. Equation (4) is the implementation of reparameterization trick to smooth the gradients. After that, we employ $\boldsymbol{\theta}$ as the dynamic neural topic to guide tag recommendation. The "dynamic" topic differs from the "static" topic [32, 22] learned by traditional LDA models. When the static topic is used for downstream tasks, it cannot be updated and optimized [22]. The dynamic topic obtained by NTG will be continuously optimized during the end-to-end training stage. Compared with the static topic, the dynamic neural topic is more suitable for tag recommendation tasks.

In the training stage, the objective function is defined based on the negative variational lower bound, which consists of two parts. The first part is the KL divergence that indicates discrepancy between prior distribution $p(\boldsymbol{z})$ and posterior distribution $q(\boldsymbol{z}|\boldsymbol{x}_{BoW})$ about the latent variable $\boldsymbol{z}$, and the second part reflects the reconstruction loss,

$$L_{NTG} = D_{KL}(q(\boldsymbol{z}|\boldsymbol{x}_{BoW})||p(\boldsymbol{z})) - E_{q(\boldsymbol{z}|\boldsymbol{x}_{BoW})}(p(\boldsymbol{x}_{BoW}|\boldsymbol{z})) \tag{7}$$

where $q(\boldsymbol{z}|\boldsymbol{x}_{BoW})$ and $p(\boldsymbol{x}_{BoW}|\boldsymbol{z})$ represent the process of inference model and generative model respectively.

3.4. Sequence Encoder

The sequence encoder is proposed to capture semantic and syntactic information in local consecutive word sequences $(\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_n) \in \mathbb{R}^{n \times V}$. Where $\boldsymbol{x}_i$ is a one-hot vector with length V, V is the vocabulary size.

3.4.1. Embedding Layer

In the embedding layer, the model mainly focuses on the extraction of word-level information. Specifically, through an embedding matrix $\boldsymbol{E}$, each word is projected into a low-dimensional real-value vector $\boldsymbol{e}_i$, i.e., word embedding [41].

$$\boldsymbol{e}_i = \boldsymbol{E}^{\mathrm{T}} \boldsymbol{x}_i \tag{8}$$

where embedding matrix $\boldsymbol{E} \in \mathbb{R}^{V \times D_{embed}}$, and D_{embed} is the size of low-dimensional embedding space. Due to the highly professional and informal

of the real-world corpus, instead of using pre-trained word embedding [13], we use the embedding matrix of random initialization [42], which is trainable in the training stage.

3.4.2. Bi-GRU

In order to capture the forward and backward sides contextual information of post, we adopt bidirectional gated recurrent unit to learn from word embedding sequences $(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n) \in \mathbb{R}^{n \times D_{embed}}$. At each time step, the hidden state can be updated by the input $\boldsymbol{e}_i$ of the current step and the hidden state of the previous step.

$$\overrightarrow{\boldsymbol{s}_i} = GRU(\overrightarrow{\boldsymbol{s}_{i-1}}, \boldsymbol{e}_i) \tag{9}$$

$$\overleftarrow{\boldsymbol{s}_i} = GRU(\overleftarrow{\boldsymbol{s}_{i-1}}, \boldsymbol{e}_i) \tag{10}$$

$$\boldsymbol{s}_i = (\overrightarrow{\boldsymbol{s}_i}, \overleftarrow{\boldsymbol{s}_i}) \tag{11}$$

where $\overrightarrow{\boldsymbol{s}_i}, \overleftarrow{\boldsymbol{s}_i} \in \mathbb{R}^{D_{hidden}/2}$ indicate the forward and backward word context representations respectively, and $D_{hidden}/2$ is the hidden size of unidirectional GRU. Then, the whole post can be represented by Bi-GRU as $(\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n) \in \mathbb{R}^{n \times D_{hidden}}$, that will be used for subsequent tasks. The sequence encoder can be replaced with any modern language models such as BERT [21], XLNet [43], BART [44] and etc.

3.5. Topic-Guided Tag Ranker

In this section, in order to leverage the topic and alleviate the data imbalance, we design a topic-guided tag ranker which includes two parts. The multi-tag topical attention mechanism is designed to get a tag-specific post representation for each tag that would capture various intensive parts of the post through the guidance of dynamic neural topic. Finally, the ranker is used to generate top-b predicted tags.

3.5.1. Multi-Tag Topical Attention

The distribution of tag is heavily skewed towards a few frequent tags with a long-tail consisting of less frequent tags, which causes the challenge of the data imbalance on tail-tag recommendation. In this situation, it is not appropriate to learn a common post representation for all tags as in the previous approach [11, 13, 12], since the head-tags can overwhelm the tail-tags. Therefore, we are motivated to get a tag-specific post representation for each tag with the help of dynamic neural topic $\boldsymbol{\theta}$. Since the most relevant

context of each tag may be different, inspired by multi-head attention [45], we exploit a multi-tag topical attention mechanism to capture various intensive parts of the post through the guidance of the topic. As shown in Figure 2, we can get the tag-specific post representation $\boldsymbol{r}_k \in \mathbb{R}^{D_{hidden}}$ of k-th tag through this kind of attention mechanism.

When calculating the topical attention score for k-th tag, the attention weights on $\boldsymbol{\alpha}_j \in \mathbb{R}^{D_{hidden}}$ is defined as:

$$g_{k,j} = \boldsymbol{v}_k^{\mathrm{T}} tanh(\boldsymbol{W}_{1,k}\boldsymbol{s}_j + \boldsymbol{W}_{2,k}\boldsymbol{\theta}) \tag{12}$$

$$\alpha_{k,j} = \frac{exp(g_{k,j})}{\sum_{j'=1}^{n} \exp(g_{k,j'})} \tag{13}$$

where $\boldsymbol{W}_{1,k} \in \mathbb{R}^{D_{hidden} \times D_{hidden}}$, $\boldsymbol{W}_{2,k} \in \mathbb{R}^{D_{hidden} \times K}$, and $\boldsymbol{v}_k \in \mathbb{R}^{D_{hidden} \times 1}$ are so-called attention parameters. The normalized coefficient $\alpha_{k,j}$ guided by the topic determines how much attention should be given to the j-th word of the post for k-th tag. We hence obtain the topic-guided post representation for k-th tag $\boldsymbol{c}_k \in \mathbb{R}^{D_{hidden}}$ with:

$$\boldsymbol{c}_k = \sum_{j=1}^{n} \alpha_{k,j}\boldsymbol{s}_j \tag{14}$$

In addition, each of the $\boldsymbol{c}_k$ is employed a fully connected feed-forward network, which is used to enrich the information extraction ability for attention mechanism. This consists of two linear transformations with a ReLU activation in between.

$$\boldsymbol{r}_k = max(0, \boldsymbol{c}_k^{\mathrm{T}}\boldsymbol{W}_{3,k} + \boldsymbol{b}_{1,k})\boldsymbol{W}_{4,k} + \boldsymbol{b}_{2,k} \tag{15}$$

Finally, $(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_l) \in \mathbb{R}^{l \times D_{hidden}}$ is the topic-guided post representation for each tag.

3.5.2. Ranker

The goal of the ranker is to model the relevance between the post and the tags. Formally, given a tag k and an instance $\boldsymbol{X}$, after the tag-specific post representation $\boldsymbol{r}_k$ is obtained, a simple one-versus-all (OVA) approach realizes the scoring function f_s as:

$$f_s(\boldsymbol{X}, k) = sigmoid(\boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{r}_k) \tag{16}$$

We further define the top-b prediction operator as:

$$f_b(\boldsymbol{X}) = Top\text{-}b([f(\boldsymbol{X}, 1), ..., f(\boldsymbol{X}, l)]) \tag{17}$$

where $f_b(\boldsymbol{X})$ is an index set containing the top-b predicted tags. Then, we use corresponding binary cross entropy as the loss function for tag ranker.

$$L_{TR} = -\sum_{i=1}^{N}\sum_{k=1}^{l}(y_{i,k}log(f(\boldsymbol{X}_i, k))) + (1 - y_{i,k})log(1 - f(\boldsymbol{X}_i, k)) \tag{18}$$

where, N is the number of training data and l is the number of tags. $y_{i,k} \in \{0, 1\}$ indicates whether the i-th post actually contains the k-th tag.

3.6. Jointly Learning Topics and Tags

Our framework first exploits the integration of neural topic generation into tag recommendation via an end-to-end multi-task learning manner. Then, we define the training objective of the entire framework with the linear combination of L_{NTG} and L_{TR}:

$$L = \delta \cdot L_{NTG} + L_{TR} \tag{19}$$

where the hyper-parameter δ balances the effects of neural topic generation and tag ranker. Our two modules can be jointly trained with their parameters updated simultaneously.

4. Experiment

In this section, we conduct extensive experiments on four real-world datasets to demonstrate the superiority of our method over several state-of-the-art methods. First, we will introduce the datasets and experimental settings and then present the baseline algorithms selected for comparisons. Furthermore, we discuss performance evaluation and analysis of model effectiveness, respectively. Finally, hyperparameters analyses and case studies are presented.

4.1. Datasets and Experimental Settings

4.1.1. Datasets

In our experiments, we use four real-world datasets of different sizes: Physics-StackExchange (Physics), Academia-StackExchange (Academia), Law-StackExchange (Law) and AskUbuntu (AU). StackExchange and AskUbuntu

Table 3: Summary of experimental datasets. No.posts: number of posts. No.words: vocabulary size of posts. No.tags: vocabulary size of tags. Avg.words: the average number of tokens per post. Avg.tags: the average number of tags per post.

Datasets	No.posts	No.words	No.tags	Avg.words	Avg.tags
AU	358,655	1,555,886	3,117	192.83	2.77
Physics	180,166	400,863	893	192.2	3.17
Academia	35,071	76,074	478	193.36	2.7
Law	18,788	68,401	799	179.74	2.52

are online Q&A forums, that are mainly for question asking (with a title and a description) and seeking answers from others. Particularly, for AskUbuntu and StackExchange datasets, we concatenate the question title together with its description as the source input. XML files of all datasets are officially published and publicly available[2]. We also released our processed datasets to facilitate related researches[3]. The statistics of the four datasets are shown in Table 3. The number of posts in these datasets ranges from 18K to 358K, while the number of tags per post is around 3. For Law dataset, it only contains 18,788 posts, but the vocabulary size of the tags is 799, so it is more difficult to recommend tags on the Law dataset.

4.1.2. Parameter Settings

For each dataset, we employed natural language toolkit (NLTK) for tokenization[4], and the most frequently 50,000 words are kept as the vocabulary. For BoW input $\boldsymbol{x}_{BoW}$ of NTG, stopwords and meaningless words (e.g., single-character ones) were removed. For all the compared methods, we set the word embedding size to 100, and the hidden size of the GRU/LSTM is 256 (128 for Bi-GRU/LSTM). We adopt the Adam optimizer with the batch size set to 256, and early stopping to stop the training process when the validation loss is no longer decreasing. The key hyper-parameters, i.e., jointly learning coefficient δ and topic number K are determined by experiments for each dataset in 4.2.4. The detailed parameter settings are released in our code.

[2] `https://archive.org/details/stackexchange`

[3] `https://drive.google.com/drive/folders/1gUj6zjfn7UzLf9_hUtfYWu0RnyZfiX-f?usp=sharing`

[4] `https://www.nltk.org/`

4.1.3. Comparisons

To demonstrate the effectiveness of TGTR on the four datasets, we selected eight most representative baseline models in the groups of related work discussed in the Session 2.1.2. Collaborative filtering based methods are excluded, since it cannot handle the cold-start items that our task focuses on.

- TagCom [10]: It analyzes tag recommendation problem by combining multi-label component, similarity-based ranking component and tag-term based ranking component based on TF-IDF features.
- ABC [11]: It adopts CNN and a word-level attention mechanism to extract global and local information for tag recommendation.
- TLSTM [22]: A two-stage method, that uses latent topic obtained by LDA model to enrich document representation by a topical attention.
- HAN [13]: It utilizes hierarchical word and sentence level attention networks for aggregating important words and sentences for tag recommendation.
- PBAM [46]: It builds a position-based attention model to automatically tag documents with keywords[5].
- ITAG [9]: It tackles the tag recommendation problem by modeling sequential text modeling, tag correlation, and content-tag overlapping into an encoder-decoder framework[6].
- ACN [12]: It incorporates an attention mechanism into the capsule network to capture important and distinguishable features for tag recommendation.
- PTM4Tag [20]: It leverages BERT-based pre-trained language models for tag recommendation with a triplet architecture, which models the components (i.e., title, description and code) of a post with independent language models.

[5]Since the datasets we study have no hierarchical information of words, we regard all words as the same level in the part of positional attention.

[6]https://github.com/SoftWiser-group/iTag

Since BERT-based model is pre-trained by a huge number of corpora, it is unfair to compare it with other competitors. For a fair comparison with PTM4Tag, we also extend TGTR in a BERT version (BTGTR), which means we replace the sequence encoder of TGTR with the encoder of pre-trained BERT [21].

4.1.4. Implementation Details

The experiments reported in this paper are performed on a Linux system with a NVIDIA RTX A4000 GPU. Furthermore, the model was implemented in Pytorch [47] using the huggingface re-implementation of BERT [48]. The whole model is trained via Adam [49]. Early-stopping strategy is adopted based on the validation loss. Before joint training, we pretrain NTG for 100 epochs to find better initial neural topic. The parameters of all baselines are either adopted from their original papers or determined by experiments. More details of implementation are released in our code.[7]

4.1.5. Evaluation Metrics

In terms of evaluation metrics, we use widely used metrics [9, 11]: Precision@k (P@k), Recall@k (R@k) and F1-score@k (F1@k) (k=1,3,5) to measure the performance. For all nine metrics, the higher the better. For the sake of brevity, we omit the percentage sign of the evaluation metrics.

$$P@k = \frac{1}{M}\sum_{i=1}^{M}\frac{hit(k)_i}{k} \tag{20}$$

$$R@k = \frac{1}{M}\sum_{i=1}^{M}\frac{hit(k)_i}{tag_i} \tag{21}$$

$$F1@k = \frac{2 \cdot P@k \cdot R@k}{P@k + R@k} \tag{22}$$

where M is the size of the test data, k is the number of the tags recommended by algorithm, $hit(k)_i$ is the right number recommended to instance i, tag_i is the actual number of the tags assigned to instance i.

4.2. Experimental Results

This section investigates the performance of the proposed TGTR on four real-world datasets (AU, Math, Physics, Academia) from five facets. We aim

[7] `https://github.com/stxupengyu/TGTR`

Table 4: Comparing TGTR with baselines on AU dataset. The best results are highlighted in bold and the second-best results are underlined.

Methods	P@1	R@1	F1@1	P@3	R@3	F1@3	P@5	R@5	F1@5
Non Pre-trained Models									
TagCom	48.6	19.4	27.7	32.3	36.5	34.3	24.4	45.1	31.7
ABC	65.8	27.6	38.9	46.3	54.0	49.9	34.8	65.7	45.5
TLSTM	67.5	28.5	40.1	47.8	55.8	51.5	35.9	67.7	46.9
HAN	66.4	28.0	39.4	46.7	54.5	50.3	35.2	66.3	46.0
PBAM	69.5	29.5	41.4	39.6	55.4	46.2	34.9	66.0	45.7
ITAG	37.8	13.9	20.3	39.6	45.0	42.1	34.7	65.2	45.3
ACN	61.6	25.6	36.2	44.5	51.7	47.8	33.8	63.7	44.2
TGTR	**71.9**	**30.5**	**42.8**	**51.0**	**59.2**	**54.8**	**38.2**	**71.4**	**49.8**
Pre-trained Models									
PTM4Tag	71.8	30.6	42.9	51.0	59.5	54.9	38.0	71.5	49.7
BTGTR	**75.1**	**32.4**	**45.3**	**53.8**	**62.9**	**58.0**	**40.3**	**75.7**	**52.6**

Table 5: Comparing TGTR with baselines on Physics dataset. The best results are highlighted in bold and the second-best results are underlined.

Methods	P@1	R@1	F1@1	P@3	R@3	F1@3	P@5	R@5	F1@5
Non Pre-trained Models									
TagCom	57.3	20.7	30.4	40.0	40.9	40.4	30.6	51.0	38.3
ABC	68.9	25.4	37.1	50.0	51.5	50.7	38.2	63.7	47.8
TLSTM	66.8	24.6	36.0	48.4	49.8	49.1	36.9	61.5	46.1
HAN	66.2	24.4	35.7	47.7	49.1	48.4	36.3	60.7	45.4
PBAM	69.4	25.7	37.5	49.3	50.7	50.0	37.1	61.9	46.4
ITAG	49.2	17.2	25.5	40.7	41.3	41.0	35.1	58.1	43.8
ACN	63.1	23.1	33.8	45.5	46.8	46.1	34.8	58.2	43.6
TGTR	**73.0**	**27.0**	**39.4**	**53.8**	**55.1**	**54.4**	**41.1**	**67.9**	**51.2**
Pre-trained Models									
PTM4Tag	74.5	27.7	40.4	53.7	55.1	54.4	40.7	67.6	50.8
BTGTR	**77.8**	**29.3**	**42.6**	**57.0**	**59.1**	**58.0**	**43.3**	**72.2**	**54.1**

Table 6: Comparing TGTR with baselines on Academia dataset. The best results are highlighted in bold and the second-best results are underlined.

Methods	P@1	R@1	F1@1	P@3	R@3	F1@3	P@5	R@5	F1@5
Non Pre-trained Models									
TagCom	44.1	18.4	26.0	32.0	38.7	35.0	24.6	48.6	32.7
ABC	61.9	26.7	37.3	<u>41.8</u>	<u>50.6</u>	<u>45.8</u>	<u>31.2</u>	<u>61.3</u>	<u>41.4</u>
TLSTM	54.1	23.2	32.5	35.5	43.2	39.0	26.6	52.9	35.4
HAN	55.0	23.6	33.0	35.9	43.5	39.3	27.0	53.5	35.9
PBAM	<u>63.0</u>	<u>27.2</u>	<u>38.0</u>	41.4	50.1	45.3	30.4	60.0	40.4
ITAG	39.5	15.4	22.2	34.5	37.5	35.9	24.0	47.3	31.8
ACN	49.4	20.9	29.4	32.4	39.0	35.4	24.4	48.1	32.4
TGTR	**66.8**	**29.0**	**40.4**	**46.3**	**55.8**	**50.6**	**34.9**	**68.1**	**46.1**
Pre-trained Models									
PTM4Tag	67.6	29.4	41.0	46.2	55.9	50.6	33.6	66.3	44.6
BTGTR	**72.8**	**31.9**	**44.4**	**50.9**	**61.3**	**55.6**	**37.6**	**73.2**	**49.7**

Table 7: Comparing TGTR with baselines on Law dataset. The best results are highlighted in bold and the second-best results are underlined.

Methods	P@1	R@1	F1@1	P@3	R@3	F1@3	P@5	R@5	F1@5
Non Pre-trained Models									
TagCom	39.9	17.4	24.2	24.9	32.1	28.0	18.4	39.1	25.0
ABC	<u>52.9</u>	<u>24.6</u>	<u>33.6</u>	<u>34.1</u>	<u>44.8</u>	<u>38.7</u>	<u>24.8</u>	<u>53.2</u>	<u>33.8</u>
TLSTM	38.1	17.2	23.7	23.7	29.1	25.3	17.0	36.3	23.2
HAN	34.5	15.7	21.6	20.9	27.4	23.7	16.2	34.6	22.1
PBAM	25.8	9.9	14.3	14.2	17.8	15.8	11.3	24.0	15.4
ITAG	28.3	11.4	16.3	14.7	17.7	16.1	9.9	19.6	13.2
ACN	37.3	17.0	23.4	21.1	27.4	23.8	15.8	33.7	21.5
TGTR	**54.8**	**25.4**	**34.7**	**37.0**	**47.8**	**41.7**	**27.8**	**58.1**	**37.6**
Pre-trained Models									
PTM4Tag	61.7	28.3	38.8	40.9	52.9	46.2	29.6	62.1	40.1
BTGTR	**62.1**	**29.3**	**39.8**	**41.7**	**54.7**	**47.4**	**30.7**	**65.4**	**41.8**

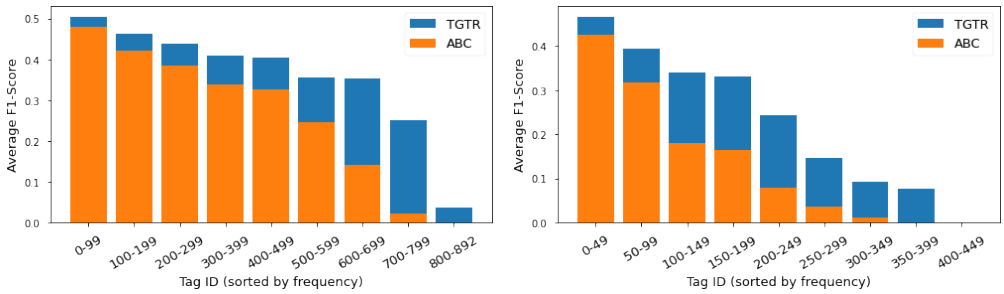
to answer the following research questions:

- RQ1: Compared with state-of-the-art tag recommendation models, how does TGTR perform?
- RQ2: Does TGTR help tail-tag recommendation?
- RQ3: What is the influence of various components in the architecture of TGTR?
- RQ4: How do different hyper-parameter settings (jointly learning coefficient δ and topic number K) affect the performance of TGTR?
- RQ5: Can TGTR explicitly show its result with practical examples?

4.2.1. Performance Comparison (RQ1)

From Table 4 to Table 7, we show the main comparison results on four datasets. For each algorithm, we conducted 10-fold cross validation and reported the average result. After analyzing the results, we have the following observations.

- TGTR significantly outperforms the existing non pre-trained methods. We observe that the proposed TGTR generally outperforms the compared methods on the four datasets. For example, on the F1@5 metric, the relative improvements of TGTR are 6.2%, 7.1%, 11.4% and 11.2% compared with its best competitors on the four datasets, respectively. TagCom performs relatively poorly as it does not consider the sequential structure of the post. For some DNN-based methods (ABC, ACN, HAN, PBAM), although they designed different strategies for text representation, they do not explicitly extract tag-specific semantic components from posts. Moreover, valuable topic information is ignored, which limits the performance of these methods. For the encoder-decoder method ITAG, we got similar results to [50]. It overemphasizes the content-tag overlapping phenomenon which is rare in real-world datasets.
- Dynamic neural topic is helpful for tag recommendation. TLSTM achieves effective results especially on large-scale dataset AU by introducing static latent topic. However, since topic generation and tag recommendation are separated in TLSTM, the topic is not particularly suitable for the downstream task. TGTR avoids this problem by



(a) Physics (b) Academia

Figure 3: Average F1-score on tags.

learning neural topic dynamically, and thus getting the dynamic neural topic that is more appropriate. Therefore, TGTR achieves the best results, significantly outperforming all (non pre-trained) comparisons by a large margin.

- Multi-tag topical attention is effective. BERT-based TGTR outperforms the SOTA pre-trained model, i.e., PTM4Tag, by a large margin. To be specific, on the F1@5 metric, the relative improvements of BTGTR are 5.8%, 6.5%, 11.4% and 4.2% compared with PTM4Tag, on the four datasets, respectively. An interesting point is that, TGTR even outperforms PTM4Tag on three datasets. This is not surprising since PTM4Tag merely models the content, and ignores the data imbalance on tags, while our approach can alleviate this problem.

4.2.2. Performance Analysis on Tail-tags (RQ2)

To further verify the proposed TGTR, we compare it with its best (non pre-trained) competitor ABC on the two datasets on tags. Figure 3 (a) shows their average F1-score on the tag groups in Physics. We can see that the F1-scores of TGTR on tail-tag groups are much higher than those from the ABC, especially on the extreme tail-tag groups (600-892). Our method is able to alleviate the long-tailed problem by introducing multi-tag topical attention. TGTR can significantly improve the performance on tail tags. Other methods, such as ABC, do not take data sparsity (on tail tags) into account, and thus perform worse on tail tags. The same phenomenon can also be observed in Figure 3 (b).

4.2.3. Ablation Study (RQ3)

We analyze the impacts of key components in TGTR via ablation studies. The default model is compared with the following variants.

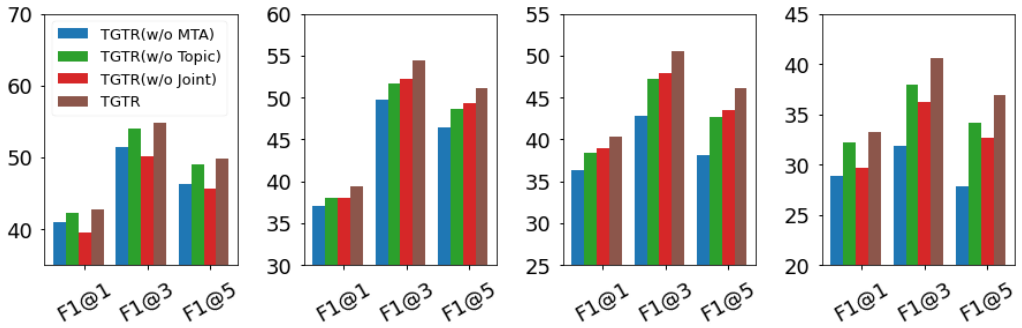
- TGTR(w/o Joint) removes the jointly training mode, i.e., fixed $\delta = 0$ in Equation. 19.
- TGTR(w/o Topic) removes the topic information in multi-tag topical attention mechanism, i.e., taking off $\boldsymbol{\theta}$ in Equation. 12.
- TGTR(w/o MTA) removes multi-tag topical attention mechanism and is solely based on sequence encoder.

Figure 4 shows the results evaluated on four datasets in terms of F1@1, F1@3 and F1@5. There are several interesting observations:

- Effectiveness of multi-tag topical attention. Compared with TGTR(w/o MTA), TGTR achieves 7.6%, 10.1%, 20.7% and 32.7% higher F1@5 on four datasets respectively. It demonstrates the effectiveness of incorporating multi-tag topical attention. Such an attention mechanism allows our model to capture various intensive parts of the post for each tag through the guidance of dynamic neural topic, thus enabling better tag recommendation.
- Jointly training mode is crucial. By comparing TGTR(w/o Joint) and TGTR, TGTR(w/o Joint) that only with pre-trained neural topic extract by NTG performs worse, and TGTR which training neural topic dynamically with NTG performs relatively much better (about 3.9%-12.8% improvement on F1@5). This means that the dynamic neural topic is much better than the static neural topic for tag recommendation. Furthermore, we observe that, TGTR(w/o Joint) is worse than TGTR(w/o Topic) on two datasets. It is reasonable since pre-trained static topic is not necessarily suitable for tag recommendation.

4.2.4. Parameter Sensitivity (RQ4)

There are two major hyper-parameters we proposed in TGTR, including topic number K and jointly learning coefficient δ that balances the effects of neural topic generation and tag ranker. The results are shown in Figure 5 and Figure 6, where we report the F1@5 results on the four datasets for brevity.

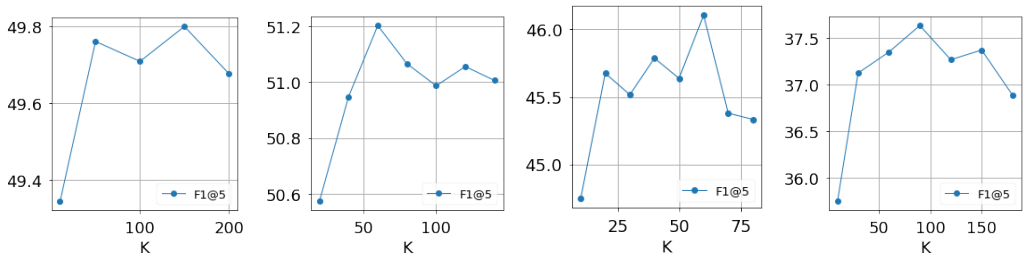


(a) AU (b) Physics (c) Academia (d) Law

Figure 4: Ablation analysis on four datasets.

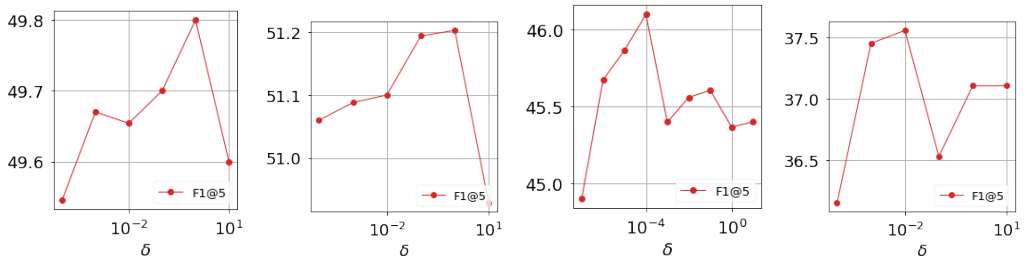
- We can observe from Figure 5 that it is crucial to choose an appropriate topic number K for each dataset. The too large or too small value of K is not conducive to TGTR learning. Generally speaking, the ideal topic number is positively correlated with the number of tags. It is consistent with our experimental results. When $K = 150$, TGTR achieves the best experimental results on AU dataset. The best values of K are $\{60, 60, 90\}$ for the last three datasets respectively. It is not surprising that the best topic number of AU is much larger than Physics, since the tag number of AU is almost three times that of Physics.

- As shown in Figure 6, an appropriate δ is also important for TGTR model, since it balances the effects of neural topic generation and tag ranker. As δ increases, the impact of NTG will be greater. When δ becomes smaller, the jointly training mode will gradually disappear. The best values of δ are $\{1, 1, 0.0001, 0.01\}$ for four datasets respectively. Since we pretrain neural topic generator for 100 epochs to find better initial neural topic before joint training, $\delta = 1$ means that in joint learning, the model pays more attention to the optimization of the topic generator. This indicates the model needs to further optimize the topic in the joint learning stage, so that the topic generated by neural topic generator will be more suitable for the current task. And when $\delta = 0.0001$, it indicates that in the joint learning stage, the model slowly optimizes the neural topic generator. It means that the gap between the initial topic and the current task is small.



(a) AU (b) Physics (c) Academia (d) Law

Figure 5: Performance of TGTR w.r.t different number of topics K.



(a) AU (b) Physics (c) Academia (d) Law

Figure 6: Performance of TGTR w.r.t different jointly learning coefficient δ.

4.2.5. Case Study (RQ5)

To further illustrate the effectiveness of the proposed model, we present a case study on the results of different algorithms. Table 8 presents the results of TGTR and ABC. Eventually our model correctly predicts 3 tags as "eclipse", "moon", "astronomy", while ABC only gives two correct prediction "eclipse" and "moon". We can observe the following phenomena:

- NTG generates meaningful topics. All the words in 1^{st} topic are related to "eclipse", and all the words in 2^{nd} topic are related to "astronomy".
- Topic information is helpful for tag recommendation. Due to the guidance of the topic, TGTR can infer tags "moon" and "eclipse" from 1^{st} topic, then infer tags "astronomy" and "sun" from 2^{nd} topic. Without such topic guidance, ABC didn't learn the co-occurrence between "earth", "moon" and "astronomy", thus it ignored the tag "astronomy".

Table 8: Example of tag recommendation results (selected from the Physics dataset). Tags in bold are target tags predicted by the different models. In the post, Words in red represent 5 most important words from the multi-tag topical attention mechanism of tag "eclipse". Words in blue represent 5 most important words of tag "astronomy".

Post: Why did the June 2011 lunar eclipse last so long? It was kind of hard to miss the lunar eclipse this week, although I didn't see it in person. From what I understand it lasted about 100 minutes. I work that out as being about 9 minutes short of 1 degree of the Moon's orbit? How did it last so long? Surely 100 minutes is more than enough time for the Moon to move out of Earth shadow, or for Earth's shadow to "overtake" the Moon?
Tags: astronomy; **moon**; **earth**; **eclipse**
TATR Result: eclipse; **moon**; **astronomy**; sun; visible-light
ABC Result: eclipse; **moon**; sun; distance; visible-light;
1^{st} **Topic:** moon; tide; tidal; shadow; moon's; ocean; lunar; night; eclipse; redshift
2^{nd} **Topic:** earth; stars; sun; astrophysics; satellites; orbital; earth's; solar-system moon; astronomy

- It is necessary to get post representation for each tag. In the post, the words in red represent the five most important words from the multi-tag topical attention mechanism of tag "eclipse". The words in blue mean the five most important words of the tag "astronomy". For each tag, its post representation focuses on different areas through the corresponding multi-tag topical attention mechanism. E.g., the tag "eclipse" focuses on words with related semantics such as "shadow" and "lunar". Tag "astronomy" focuses on the words with related semantics such as "orbit" and "Earth". It further demonstrates the effectiveness of multi-tag topical attention mechanism, which can learn a tag-specific post representation for each tag that would capture various intensive parts of the post through the guidance of the topic.

5. Conclusions and Future Work

In this paper, we propose a novel topic-guided tag recommendation model TGTR to recommend tags for textual content, which jointly incorporates dynamic neural topic. Firstly, we use a neural topic generator to get the dynamic neural topic that would indicate the tags of the post based on BoW feature. Secondly, the sequence encoder is proposed to capture semantic and syntactic information in local consecutive word sequences. Thirdly, in order to leverage the topic-tag correlation and alleviate the data imbalance, we design a multi-tag topical attention mechanism to get a tag-specific post representation for each tag that would capture various intensive parts of the post through the guidance of dynamic neural topic. Extensive experiments demonstrate that TGTR outperforms the state-of-the-art approaches by a large margin, especially on tail-tags.

In the future, we would like to explore how to leverage the contextualized representations of the post to further improve the topic generation process. In addition, to take advantage of domain knowledge, domain-specific language models would be investigated as the sequence encoder.

6. Acknowledgments

This work was partly supported by the Beijing Natural Science Foundation under Grant (Z180006, L211016); the National Natural Science Foundation of China under Grant (62176020); the National Key Research and Development Program (2020AAA0106800); CAAI-Huawei MindSpore Open Fund; and Chinese Academy of Sciences (OEIP-O-202004).

References

[1] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, C. L. Giles, Real-time automatic tag recommendation, in: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pp. 515–522.

[2] B. Chen, Y. Ding, X. Xin, Y. Li, Y. Wang, D. Wang, Airec: Attentive intersection model for tag-aware recommendation, Neurocomputing 421 (2021) 105–114.

[3] M. Yang, S. Cao, B. Hu, X. Chen, H. Cui, Z. Zhang, J. Zhou, X. Li, Intellitag: An intelligent cloud customer service system based on tag recommendation, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 2559–2570.

[4] B. Sigurbjörnsson, R. van Zwol, Flickr tag recommendation based on collective knowledge, in: Proceedings of the 17th international conference on World Wide Web, 2008, pp. 327–336.

[5] F. M. Belém, J. M. Almeida, M. A. Gonçalves, A survey on tag recommendation methods, in: Journal of the Association for Information Science and Technology archive, Vol. 68, 2017, pp. 830–844.

[6] R. Huang, N. Wang, C. Han, F. Yu, L. Cui, Tnam: A tag-aware neural attention model for top-n recommendation, Neurocomputing 385 (2020) 1–12.

[7] S. Ahmadian, M. Ahmadian, M. Jalili, A deep learning based trust-and tag-aware recommender system, Neurocomputing 488 (2022) 557–571.

[8] S. Rendle, L. Balby Marinho, A. Nanopoulos, L. Schmidt-Thieme, Learning optimal ranking with tensor factorization for tag recommendation, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 727–736.

[9] S. Tang, Y. Yao, S. Zhang, F. Xu, T. Gu, H. Tong, X. Yan, J. Lu, An integral tag recommendation model for textual content, in: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Annual Conference on Innovative Applications of Artificial Intelligence, IAAI 2019 and

the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Vol. 33, 2019, pp. 5109–5116.

[10] X. Xia, D. Lo, X. Wang, B. Zhou, Tag recommendation in software information sites, in: 2013 10th Working Conference on Mining Software Repositories (MSR), 2013, pp. 287–296.

[11] Y. Gong, Q. Zhang, Hashtag recommendation using attention-based convolutional neural network, in: IJCAI'16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2782–2788.

[12] K. Lei, Q. Fu, M. Yang, Y. Liang, Tag recommendation by text classification with attention-based capsule network, Neurocomputing 391 (2020) 65–73.

[13] H. A. M. Hassan, G. Sansonetti, F. Gasparetti, A. Micarelli, Semantic-based tag recommendation in scientific bookmarking systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 465–469.

[14] Y. S. Rawat, M. S. Kankanhalli, Contagnet: Exploiting user context for image tag recommendation, in: Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 1102–1106.

[15] Y. Wang, S. Wang, J. Tang, G. Qi, H. Liu, B. Li, Clare: A joint approach to label classification and tag recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, 2017.

[16] M. Li, T. Gan, M. Liu, Z. Cheng, J. Yin, L. Nie, Long-tail hashtag recommendation for micro-videos with graph convolutional network, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 509–518.

[17] Y. Wei, Z. Cheng, X. Yu, Z. Zhao, L. Zhu, L. Nie, Personalized hashtag recommendation for micro-videos, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 1446–1454.

[18] Y. Wu, Y. Yao, F. Xu, H. Tong, J. Lu, Tag2word: Using tags to generate words for content based tag recommendation, in: Proceedings of the

25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 2287–2292.

[19] Y. Wu, S. Xi, Y. Yao, F. Xu, H. Tong, J. Lu, Guiding supervised topic modeling for content based tag recommendation, Neurocomputing 314 (2018) 479–489.

[20] J. He, B. Xu, Z. Yang, D. Han, C. Yang, D. Lo, Ptm4tag: Sharpening tag recommendation of stack overflow posts with pre-trained models, arXiv preprint arXiv:2203.10965 (2022).

[21] J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[22] Y. Li, T. Liu, J. Jiang, L. Zhang, Hashtag recommendation with topical attention-based lstm, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3019–3029.

[23] Y. Miao, E. Grefenstette, P. Blunsom, Discovering discrete latent topics with neural variational inference, in: the 34th International Conference on Machine Learning (ICML), 2017, pp. 2410–2419.

[24] S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, in: Proceedings of the third ACM international conference on Web search and data mining, 2010, pp. 81–90.

[25] X. Fang, R. Pan, G. Cao, X. He, W. Dai, Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 29, 2015.

[26] W. Feng, J. Wang, Incorporating heterogeneous information for personalized tag recommendation in social tagging systems, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 1276–1284.

[27] M. Shi, J. Liu, D. Zhou, Y. Tang, A topic-sensitive method for mashup tag recommendation utilizing multi-relational service data, IEEE Transactions on Services Computing 14 (02) (2021) 342–355.

[28] X. Chen, Y. Yu, F. Jiang, L. Zhang, R. Gao, H. Gao, Graph neural networks boosted personalized tag recommendation algorithm, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[29] J. Sun, M. Zhu, Y. Jiang, Y. Liu, L. Wu, Hierarchical attention model for personalized tag recommendation, Journal of the Association for Information Science and Technology 72 (2) (2021) 173–189.

[30] W. Zhao, A. Zhang, L. Shang, Y. Yu, L. Zhang, C. Wang, J. Chen, H. Yin, Hyperbolic personalized tag recommendation, in: International Conference on Database Systems for Advanced Applications, Springer, 2022, pp. 216–231.

[31] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, W. Buntine, Topic modelling meets deep neural networks: A survey, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, 2021, pp. 4713–4720.

[32] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[33] D. Ramage, D. Hall, R. Nallapati, C. D. Manning, 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, 2009, pp. 248–256.

[34] D. P. Kingma, M. Welling, Auto-encoding variational bayes, in: ICLR 2014 : International Conference on Learning Representations (ICLR) 2014, 2014.

[35] Y. Wang, J. Li, H. P. Chan, I. King, M. R. Lyu, S. Shi, Topic-aware neural keyphrase generation for social media language, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2516–2526.

[36] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du, W. Buntine, Topic modelling meets deep neural networks: A survey, arXiv preprint arXiv:2103.00498 (2021).

[37] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, Data mining with big data, IEEE transactions on knowledge and data engineering 26 (1) (2013) 97–107.

[38] S. Li, R. Pan, Y. Zhang, Q. Yang, Correlated tag learning in topic model., in: UAI, 2016.

[39] A. Srivastava, C. Sutton, Autoencoding variational inference for topic models, in: 5th International Conference on Learning Representations, 2017.

[40] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: International conference on machine learning, PMLR, 2016, pp. 1727–1736.

[41] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems 26, Vol. 26, 2013, pp. 3111–3119.

[42] H. Huang, Q. Zhang, Y. Gong, X. Huang, Hashtag recommendation using end-to-end memory networks with hierarchical attention, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 943–952.

[43] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).

[44] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[46] B. Sun, Y. Zhu, Y. Xiao, R. Xiao, Y. Wei, Automatic question tagging with deep neural networks, IEEE Transactions on Learning Technologies 12 (1) (2019) 29–43.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[48] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).

[49] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: ICLR 2015 : International Conference on Learning Representations 2015, 2015.

[50] J. Gao, Y. He, Y. Wang, X. Wang, J. Wang, G. Peng, X. Chu, Star: Spatio-temporal taxonomy-aware tag recommendation for citizen complaints, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1903–1912.