STAR: Spatio-Temporal Taxonomy-Aware Tag Recommendation for Citizen Complaints[*]

Jingyue Gao, Yuanduo He[†]
{gaojingyue1997,ydhe}@pku.edu.cn
Peking University

Yasha Wang[‡]
wangyasha@pku.edu.cn
Peking University

Xiting Wang
xitwan@microsoft.com
Microsoft Research Asia

Jiangtao Wang
jiangtao.wang@lancaster.ac.uk
Lancaster University

Guangju Peng, Xu Chu
{pgj.pku12,chu_xu}@pku.edu.cn
Peking University

ABSTRACT

In modern cities, complaining has become an important way for citizens to report emerging urban issues to governments for quick response. For ease of retrieval and handling, government officials usually organize citizen complaints by manually assigning tags to them, which is inefficient and cannot always guarantee the quality of assigned tags. This work attempts to solve this problem by recommending tags for citizen complaints. Although there exist many studies on tag recommendation for textual content, few of them consider two characteristics of citizen complaints, i.e., the spatio-temporal correlations and the taxonomy of candidate tags. In this paper, we propose a novel **S**patio-Temporal **T**axonomy-**A**ware **R**ecommendation model (STAR), to recommend tags for citizen complaints by jointly incorporating spatio-temporal information of complaints and the taxonomy of candidate tags. Specifically, STAR first exploits two parallel channels to learn representations for textual and spatio-temporal information. To effectively leverage the taxonomy of tags, we design chained neural networks that gradually refine the representations and perform hierarchical recommendation under a novel taxonomy constraint. A fusion module is further proposed to adaptively integrate contributions of textual and spatio-temporal information in a tag-specific manner. We conduct extensive experiments on a real-world dataset and demonstrate that STAR significantly performs better than state-of-the-art methods. The effectiveness of key components in our model is also verified through ablation studies.

CCS CONCEPTS

• **Information systems → Data mining**; **Recommender systems**; • **Computing methodologies → Machine learning**;

KEYWORDS

tag recommendation; mining citizen complaints; spatio-temporal; taxonomy-aware

ACM Reference format:
Jingyue Gao, Yuanduo He, Yasha Wang, Xiting Wang, Jiangtao Wang, and Guangju Peng, Xu Chu. 2019. STAR: Spatio-Temporal Taxonomy-Aware Tag Recommendation for Citizen Complaints. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management, Beijing, China, November 3–7, 2019 (CIKM '19),* 10 pages.
https://doi.org/10.1145/3357384.3357894

1 INTRODUCTION

With the rapid progress of urbanization, citizens are increasingly concerned with emerging environmental and societal issues within cities, such as noise pollution [47] and traffic congestion [20], many of which are related to unsatisfying public services or lack of effective governance [46]. Thus governments need to be well informed of these problems so that they can quickly respond and conduct flexible governance. To this end, a number of platforms have been developed and deployed in many cities to collect and response to citizen complaints [4, 28, 47].

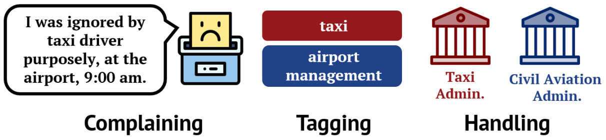


Figure 1: A typical citizen complaint platform.

The basic workflow of a typical citizen complaint platform in Figure 1 is as follows: 1) citizens submit complaints; 2) government officials assign complaints tags that help describe their content (one complaint can have multiple tags); 3) different departments retrieve relevant complaints based on tags and handle them properly. Tagging is of great importance in the workflow as it enables better organization and retrieval of content [1, 31, 39]. Analysis of complaints with suitable tags can also provide governments with valuable insights into problems within cities and facilitate better

[*] Jingyue Gao, Yuanduo He, Yasha Wang, Guangju Peng and Xu Chu are with Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing, China. Yuanduo He and Yasha Wang are also with National Engineering Research Center of Software Engineering, Peking University, Beijing, China.
[†] Jingyue Gao and Yuanduo He contributed equally to this work.
[‡] Yasha Wang is the corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357894

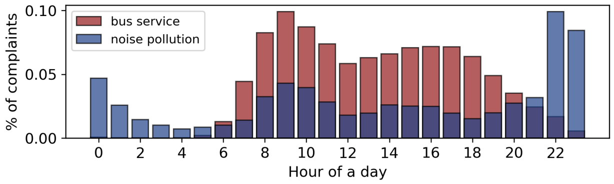


Figure 2: Temporal distributions of complaints tagged with *bus service* (red) and *noise pollution* (blue), respectively.

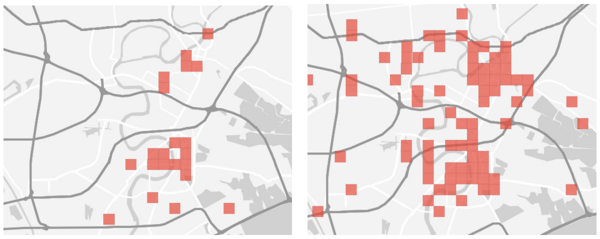


Figure 3: Spatial distributions of complaints tagged with *taxi* (left) and *garbage* (right), respectively.

policy-making [14, 47]. However, it is often painstaking to manually tag citizen complaints, especially when officials are uncertain about the suitable tags among hundreds of candidates. Therefore, automatic tag recommendation for complaints is essential as it not only improves the tagging efficiency but also guarantees the quality of assigned tags, which becomes an indispensable component of intelligent city management.

Prior efforts on tag recommendation for textual content are mainly based on topic models [19, 24, 27, 39]. Some recent studies try to adopt deep neural networks since they have shown promising results in many language processing tasks [16, 29]. For example, Gong and Zhang [9] propose a two-way CNN architecture that consists of local and global channels. Li et al. [21] incorporate topic modeling into LSTM through an attention mechanism and Tang et al. [31] apply a novel seq2seq model. These methods have achieved encouraging performance due to their ability to learn good representations for textual content. However, merely modeling the content of complaints is not enough. We identify the following two key characteristics of tag recommendation for citizen complaints, which offer new opportunities for further improvement.

C1. Spatio-Temporal Correlations. In addition to textual content, citizen complaints often contain timestamps and locations that indicate when and where the concerned issues happen. We observe that many tags of complaints are highly correlated with their spatio-temporal information. As shown in Figure 2, complaints[1] tagged with *noise pollution* happen most around midnight, while those tagged with *bus service* mainly occur during the running time of public buses. Similarly, in Figure 3, complaints with tag *taxi* usually happen at places next to roads, while those tagged with *garbage* scatter more randomly in the whole city. Spatio-Temporal correlations can be viewed as prior knowledge that suggests which tags are suitable for a complaint even without its textual content. Thus, we are inspired to harness spatio-temporal information to enhance tag recommendation for complaints.

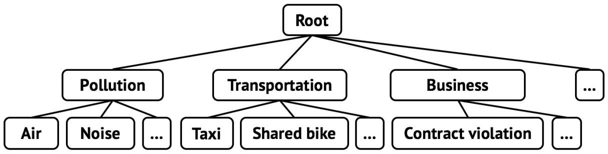


Figure 4: Taxonomy of candidate tags for citizen complaints.

C2. Taxonomy of Candidate Tags. Similar to E-commerce websites which provide item taxonomies [44], citizen compliant platforms usually set up a taxonomy of candidate tags for ease of management. As shown in Figure 4 , the taxonomy is essentially a tree-like structure where parents represent more abstract meanings than their children (e.g., *noise pollution* and *air pollution* belong to the general category of *pollution*). Since the taxonomy captures semantic tag correlations, we are motivated to leverage this structured knowledge to improve the tag recommendation performance.

However, there exist several challenges in integrating above observations **C1** and **C2** for tag recommendation. First, it is improper to trivially combine textual content and spatio-temporal information together since they are of different modalities and describe complaints from different views. Thus they are supposed to be elaborately encoded and represented. Second, spatio-temporal information is of different importance to different tags. It may complement textual content well when recommending some tags but could be useless or even noise for other tags, making it difficult to fuse textual and spatio-temporal information effectively. Third, it is challenging to further consider the taxonomy of tags given above concerns. Some pioneering studies [3, 8] propose to learn node embeddings for a taxonomy, which fails to fully capture its structured knowledge. How to explicitly incorporate the tag taxonomy into our recommendation task still remains unexplored.

To tackle the aforementioned challenges, we propose a novel **S**patio-Temporal **T**axonomy-**A**ware **R**ecommendation model (STAR) to recommend tags for citizen complaints, which jointly incorporates spatio-temporal information and the taxonomy of candidate tags. Specifically, STAR includes two parallel channels, a textual channel to encode the textual content of a complaint, and a spatio-temporal channel to encode its spatio-temporal information. With textual and spatio-temporal representations respectively, we design chained neural networks that predict which tags should be recommended at each level of the taxonomy from general to specific. Predictions that violate parent-children relationships are further penalized. In this way, the obedience of the hierarchical structure is explicitly encouraged and the representations are gradually refined from coarse to fine. Thereafter, to effectively combine results based on textual and spatio-temporal information, we propose a fusion module that adaptively discriminates contributions of these two counterparts in a tag-specific manner. We conduct extensive experiments on a real-world dataset of citizen complaints to verify the rationality and effectiveness of our model.

To summarize, this work makes the following key contributions:

- We propose to integrate textual and spatio-temporal information to recommend tags for citizen complaints. In particular, we separately encode them through two parallel channels and adaptively fuse recommendations in a tag-specific way.

[1] Detailed description of the citizen complaints dataset is in Section 4.1.1

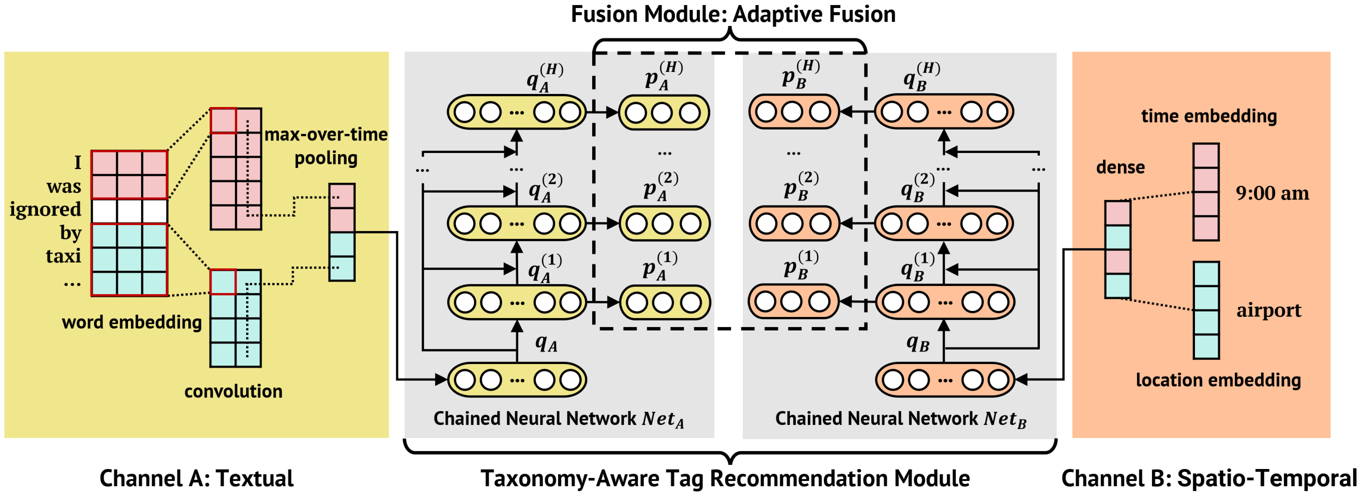


Figure 5: The framework of STAR.

- We propose to explicitly take into consideration the information of tag taxonomy for tag recommendation. Particularly, we incorporate the taxonomy of tags by employing chained neural networks for hierarchical recommendation with a novel taxonomy constraint.
- Experimental results on a real-world dataset show that STAR significantly outperforms state-of-the-art methods. Important components of our model are verified through ablation studies. We further demonstrate its robustness via thorough parameter sensitivity analysis. Case studies are also conducted to show that STAR can fuse textual and spatio-temporal information reasonably.

2 PROBLEM FORMULATION

We define the problem of tag recommendation for citizen complaints as follows.

Input. The input data of our model consists of a citizen complaint C and a given taxonomy Υ of tags.

Definition 1. *The **citizen complaint** C is a three-tuple, i.e., (C_w, C_{time}, C_{loc}), where C_w is a sentence of words describing the complaint content, C_{time} is the timestamp indicating when the concerned issue happens, and C_{loc} is the corresponding location.*

Definition 2. *The **tag taxonomy** Υ is a tree that expresses the hierarchy of tags via a parent-child relationship. As shown in Figure 4, a child node (e.g.,* air pollution*) is a sub-tag of its parent (e.g.,* pollution*). We denote the set of nodes in Υ as $\mathcal{T}$, where $\mathcal{T} = \{t_1, t_2, ..., t_L\}$ and L represents the total number of nodes in Υ. The set of all leaves in Υ is denoted as $\mathcal{T}_{leaf}$ and that of inner nodes is denoted as $\mathcal{T}_{inner}$.*

Definition 3. *The **hierarchical level** h_l of t_l is recursively defined based on its parent: $h_l = h_{parent(l)} + 1$, where $t_{parent(l)}$ is the parent of t_l. If t_l is the root of Υ, h_l is set to 0. Note that hierarchical levels of all leaves may not be the same. We use H to denote the largest hierarchical level of nodes.*

Output. Given a citizen complaint C and a tag taxonomy Υ, for each leaf tag $t_l \in \mathcal{T}_{\text{leaf}}$, our model outputs the probability $p_l \in [0, 1]$ that t_l is suitable for this complaint.

3 THE PROPOSED MODEL

In this section, we first introduce the model overview. Then, we describe the design of major components in STAR and illustrate how the components can be jointly optimized.

3.1 Model Overview

The overview of the STAR framework is illustrated in Figure 5. It mainly consists of the following components.

Textual and Spatio-Temporal Channels. Since textual content and spatio-temporal information are of different modalities and describe complaints from different views, we apply the textual channel A and the spatio-temporal channel B to process the complaint C respectively. In particular, channel A takes C_w as input and obtain the textual representation $\mathbf{q}_A$ that capture semantics of C. Given C_{time} and C_{loc}, channel B obtains the representation $\mathbf{q}_B$ that captures spatio-temporal features of C, where we also consider the spatial and temporal smoothness.

Taxonomy-Aware Tag Recommendation Module. In this module, we recommend tags based on $\mathbf{q}_A$ and $\mathbf{q}_B$. To explicitly incorporate the tag taxonomy Υ to guide the recommendation process, we design two symmetric **chained neural networks** (Net_A and Net_B) which operate on $\mathbf{q}_A$ and $\mathbf{q}_B$ respectively. Under the **taxonomy constraint**, the chained neural networks perform tag recommendation at different hierarchical levels of Υ and the initial representation $\mathbf{q}_A$ / $\mathbf{q}_B$ is gradually refined from coarse to fine, thus leading to better performance.

Fusion Module. The recommendation results (i.e, $p_{A,l}, p_{B,l}$ for each $t_l \in \mathcal{T}$) based on $\mathbf{q}_A$ and $\mathbf{q}_B$ are integrated in the fusion module. Since the relative importance of textual and spatio-temporal information varies when we consider different tags for recommendation, we perform **adaptive fusion** by determining the contributions of these two counterparts in a tag-specific manner.

3.2 Textual Channel

Given $C_w = [w_1, w_2, \cdots, w_n]$ with n words, we aim to obtain textual representation of C that captures its semantics. Here we choose the Kim CNN [16] architecture, which has been widely used for sentence representation learning [2, 33] due to its ability to simultaneously preserve word orders and enable efficient computation.

Specifically, C_w is first encoded into $\mathbf{W}_{1:n} = [e_1, e_2, \cdots, e_n] \in \mathbb{R}^{d_w \times n}$ through a word embedding layer, where d_w is the embedding size. Then a convolution operation with filter $\mathbf{f} \in \mathbb{R}^{d_w \times s}$ is applied to $\mathbf{W}_{1:n}$, where s ($s \leq n$) is the window size of the filter. A feature c_i^f is computed from the sub-matrix $\mathbf{W}_{i:i+s-1}$ as:

$$c_i^f = \sigma(\mathbf{f} * \mathbf{W}_{i:i+s-1} + b), \tag{1}$$

where $*$ is the convolution operation, σ is an activation function, and $b \in \mathbb{R}$ is the bias. By applying the filter to every possible position of $\mathbf{W}_{1:n}$ and performing max-over-time pooling, we select the most significant feature:

$$\hat{c}^f = \max(\{c_1^f, c_2^f, \cdots, c_{n-s+1}^f\}) \in \mathbb{R}. \tag{2}$$

Finally, all features $\hat{c}^f$ from multiple filters (with varying windows sizes) are concatenated together as the final representation:

$$\mathbf{q}_A = [\hat{c}^{f_1}, \hat{c}^{f_2}, \cdots, \hat{c}^{f_{d_A}}] \in \mathbb{R}^{d_A}, \tag{3}$$

where d_A is the total number of filters.

3.3 Spatio-Temporal Channel

In this channel, we aim to process C_{time} and C_{loc} to obtain the representation that captures spatio-temporal features of C.

Specifically, we equally partition a day into N_{time} slots and discretize C_{time} by the slot it falls into. Similarly, we partition the city into a grid map of N_{loc} cells for the discretization of C_{loc}. Therefore, C_{time} and C_{loc} are coded as two one-hot vectors. Then, we leverage two embedding matrices, $\mathbf{E}_{time}$ and $\mathbf{E}_{loc}$, to embed C_{time} and C_{loc} as $\mathbf{e}_{time} \in \mathbb{R}^{d_{time}}$, $\mathbf{e}_{loc} \in \mathbb{R}^{d_{loc}}$ respectively, where d_{time}, d_{loc} are embedding sizes and $\mathbf{E}_{time} \in \mathbb{R}^{N_{time} \times d_{time}}$, $\mathbf{E}_{time} \in \mathbb{R}^{N_{loc} \times d_{loc}}$ are parameters to learn.

We further concatenate e_{time}, e_{loc} together and pass it through a dense layer, i.e. fully connected network, to obtain the joint representation of spatio-temporal information by:

$$\mathbf{q}_B = \sigma(\mathbf{W}_B(\mathbf{e}_{time} \oplus \mathbf{e}_{loc}) + \mathbf{b}_B) \in \mathbb{R}^{d_B}, \tag{4}$$

where d_B is the size for spatio-temporal representation, $\oplus$ is the concatenation operator, $\mathbf{W}_B \in \mathbb{R}^{d_B \times (d_{time} + d_{loc})}$, $\mathbf{b}_B \in \mathbb{R}^{d_B}$ are parameters to learn.

To better instruct the learning of $\mathbf{E}_{time}$ and $\mathbf{E}_{loc}$, we additionally consider their temporal and spatial smoothness. Intuitively, two adjacent time slots (e.g., 1-2 a.m. and 2-3 a.m.) tend to show similar temporal characteristics and thus their corresponding embeddings are expected to be similar. The above observation also applies to locations. To incorporate such smoothness prior, we introduce the following loss term $\mathcal{L}_{\text{smo}}$ on $\mathbf{E}_{time}$ and $\mathbf{E}_{loc}$:

$$\mathcal{L}_{\text{smo}} = \|\mathbf{M}_{time}\mathbf{E}_{time}\|_F^2 + \|\mathbf{M}_{loc}\mathbf{E}_{loc}\|_F^2, \tag{5}$$

where $\|\cdot\|_F$ is the Frobenius norm. Inspired by [34], we define $\mathbf{M}_{time} \in \mathbb{R}^{N_{time} \times N_{time}}$ describing correlations between time slots and $\mathbf{M}_{loc} \in \mathbb{R}^{N_{loc} \times N_{loc}}$ describing correlations between cells by:

$$\mathbf{M}_{time} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 & -1 \\ -1 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{6}$$

$$\mathbf{M}_{loc} = \begin{bmatrix} 1 & -s_{1,2}/S_1 & \cdots & -s_{1,N_{loc}}/S_1 \\ -s_{2,1}/S_2 & 1 & \ddots & -s_{2,N_{loc}}/S_2 \\ \vdots & \vdots & \ddots & \vdots \\ -s_{N_{loc},1}/S_{N_{loc}} & -s_{N_{loc},2}/S_{N_{loc}} & \cdots & 1 \end{bmatrix}, \tag{7}$$

where $s_{i,j} = \frac{1}{d_{i,j}}$ and $d_{i,j}$ is the euclidean distance between the centers of cell i and j and $S_i = \sum_{j \neq i} s_{i,j}$ is for scaling.

3.4 Taxonomy-Aware Tag Recommendation

In this module, we perform tag recommendation based on $\mathbf{q}_A$ and $\mathbf{q}_B$. To leverage the tag taxonomy in guiding the recommendation process, we hierarchically make predictions for tags at different levels of Υ from general to specific, and impose the taxonomy constraint that encourages obedience of this hierarchical structure.

Since $\mathbf{q}_A$ and $\mathbf{q}_B$ describe the complaint from different views, two independent sets of recommendation results based on them can be obtained and further integrated (see Section 3.5), which is a common paradigm of multi-view learning [40]. As shown in Figure 5, two parallel chained neural networks Net_A and Net_B are employed to operate on them respectively. In the following, since Net_A and Net_B only differ in their inputs, we focus on illustrating the process for Net_A without loss of generality.

3.4.1 Hierarchical Recommendation.

3.4.1 Hierarchical Recommendation. We model the taxonomy of tags in Net_A by jointly predicting p_l for t_l at different levels. Our method is based on the chained neural network [36, 37] which is effective in generating multiple outputs in a hierarchical structure.

Since tags at different levels describe a complaint at different levels of abstractness, hierarchical recommendation over Υ requires representations of different granularities. In Net_A, the initial textual representation $\mathbf{q}_A$ sequentially flows through H chained layers, each layer corresponding to one level of Υ. In this way, $\mathbf{q}_A$ can be gradually refined from coarse-grained to fine-grained. Specifically, the representation $\mathbf{q}_A^{(1)}$ of the first level is calculated as follows:

$$\mathbf{q}_A^{(1)} = \sigma(\mathbf{W}_A^{(1)}\mathbf{q}_A + \mathbf{b}_A^{(1)}) \in \mathbb{R}^{d_1}, \tag{8}$$

where $\mathbf{W}_A^{(1)} \in \mathbb{R}^{d_1 \times d_A}$, $\mathbf{b}_A^{(1)} \in \mathbb{R}^{d_1}$ are parameters to learn. The representation $\mathbf{q}_A^{(h+1)}$ of level $h+1$ is based on that of level h:

$$\mathbf{q}_A^{(h+1)} = \sigma(\mathbf{W}_A^{(h+1)}(\mathbf{q}_A^{(h)} \oplus \mathbf{q}_A) + \mathbf{b}_A^{(h+1)}) \in \mathbb{R}^{d_{h+1}}, \tag{9}$$

where $\mathbf{W}_A^{(h+1)} \in \mathbb{R}^{d_{h+1} \times (d_A + d_h)}$, $\mathbf{b}_A^{(h+1)} \in \mathbb{R}^{d_{h+1}}$ are parameters to learn. Here we add a residual connection for $\mathbf{q}_A$ to avoid the loss of raw information during propagation through multiple layers.

Let L_h denote the number of tags at level h. With $\mathbf{q}_A^{(h)}$, we aim to calculate a result vector $\mathbf{p}_A^{(h)} \in \mathbb{R}^{L_h}$ which includes predicted probabilities $p_{A,l} \in [0,1]$ for all t_l at level h:

$$\mathbf{p}_A^{(h)} = [\cdots, p_{A,l}, \cdots] = \mathrm{sigmoid}(\mathrm{MLP}(\mathbf{q}_A^{(h)})), \quad (10)$$

where MLP(·) is a multi-layer perceptron that consists of one hidden layer for feature transformation and another layer for output.

3.4.2 Taxonomy Constraint. We further model the parent-child relationship between tags. Intuitively, *if tag t_l is suitable for complaint C, at least one child of t_l is also suitable and vice versa.* The taxonomy constrains that the predicted probability for an inner node should be close to the maximal value of its children. To penalize violation of this structural constraint, as illustrated in Figure 6 (a), we propose to perform max pooling over children of inner nodes and compute the following loss term:

$$\mathcal{L}_{\mathrm{tax}} = \sum_{t_l \in \mathcal{T}_{inner}} \| p_{A,l} - \max_{i \in children(l)} p_{A,i} \|^2, \quad (11)$$

where $p_{A,l}$ is the predicted probability for t_l by (10), and $children(l)$ is the index set of all children of t_l. By considering $\mathcal{L}_{\mathrm{tax}}$ in final optimization, we explicitly encourage obedience of the hierarchical structure in Υ. To help clarify this, the example in Figure 6 (d) is preferred to that in (b) and (c).

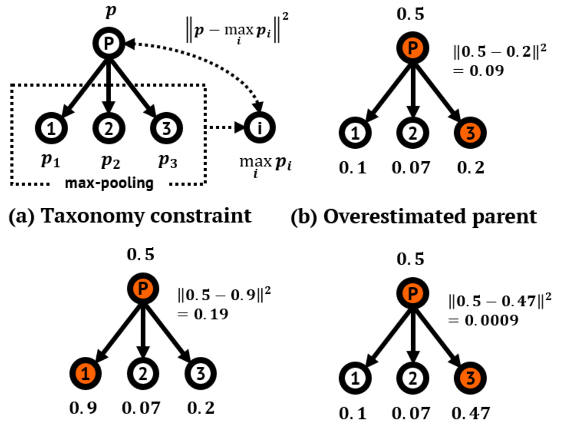
3.4.3 Summary. We apply binary cross entropy to calculate the prediction loss for $t_l \in \mathcal{T}$. Since only $t_l \in \mathcal{T}_{\mathrm{leaf}}$ will be recommended in practice, we particularly divide the loss of all tags into two groups:

$$\begin{aligned} \mathcal{L}_{\mathrm{inner}} &= - \sum_{t_l \in \mathcal{T}_{\mathrm{inner}}} (1 - y_l) \log(1 - p_{A,l}) + y_l \log p_{A,l}, \\ \mathcal{L}_{\mathrm{leaf}} &= - \sum_{t_l \in \mathcal{T}_{\mathrm{leaf}}} (1 - y_l) \log(1 - p_{A,l}) + y_l \log p_{A,l}, \end{aligned} \quad (12)$$

where $y_l \in \{0,1\}$ is the ground truth of whether t_l is assigned to C. y_l for $t_l \in \mathcal{T}_{\mathrm{inner}}$ can be recursively inferred: $y_l = \max_{i \in children(l)} y_i$. Then, the total loss for Net_A is defined as:

$$\mathcal{L}_A = \mathcal{L}_{\mathrm{leaf}} + \lambda_{\mathrm{inner}} \mathcal{L}_{\mathrm{inner}} + \lambda_{\mathrm{tax}} \mathcal{L}_{\mathrm{tax}}, \quad (13)$$

where $\lambda_{\mathrm{inner}}, \lambda_{\mathrm{tax}} > 0$ are hyper-parameters. The loss $\mathcal{L}_B$ for Net_B is defined in the same way.



(a) Taxonomy constraint **(b) Overestimated parent**

(c) Underestimated parent **(d) Properly estimated parent**

Figure 6: Illustration of our taxonomy constraint.

3.5 Fusion Module

For each $t_l \in \mathcal{T}$, the recommendation module outputs two predictions $p_{A,l}$ and $p_{B,l}$ based on $\mathbf{q}_A$ and $\mathbf{q}_B$ respectively. In this module, we aim to fuse them for the final prediction p_l.

Instead of using a constant fusion weight, we propose to adaptively combine $p_{A,l}$ and $p_{B,l}$ in a tag-specific manner, i.e, different fusion weights are used for different tags. The reason is that the relative importance of textual content and spatio-temporal information in the recommendation task varies with tags. For example, some tags are highly correlated with spatio-temporal information while others may be weakly correlated. Spatio-Temporal information is more important in the former case than the latter. To this end, we introduce a fusion vector $\mathbf{w} \in [0,1]^L$ to learn, where w_l represents the weight in combining $p_{A,l}$ and $p_{B,l}$:

$$p_l = w_l \, p_{A,l} + (1 - w_l) \, p_{B,l}. \quad (14)$$

Here the larger w_l is, the more textual information contributes to the final recommendation. Given p_l for each $t_l \in \mathcal{T}$, we compute the fusion loss $\mathcal{L}_{\mathrm{fusion}}$ similar to that for Net_A and Net_B in (13).

3.6 Joint Learning

Composed of channel A, channel B, recommendation module, and fusion module, our STAR model can be jointly optimized in an end-to-end manner. We minimize the following objective function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{fusion}} + \lambda_A \mathcal{L}_A + \lambda_B \mathcal{L}_B + \lambda_{\mathrm{smo}} \mathcal{L}_{\mathrm{smo}}, \quad (15)$$

where $\lambda_A, \lambda_B, \lambda_{\mathrm{smo}} > 0$ are weights for corresponding loss terms. We use the Adam optimizer [17] because it can automatically adjust the learning rate during the training phase.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate our proposed model[2]. We aim to answer the following research questions:

- **RQ1:** Compared with state-of-the-art tag recommendation models, how does STAR perform?
- **RQ2:** What is the influence of various components in the architecture of STAR?
- **RQ3:** How do different hyper-parameter settings (e.g., the size of training data, weights of loss terms and the size of word embedding) affect the performance of STAR?
- **RQ4:** Can the fusion module reasonably integrate results based on textual and spatio-temporal information?

4.1 Experimental Settings

4.1.1 Dataset Description. Our dataset comes from the records of a citizen complaint platform in Tianjin. With the resident population around 16 million, Tianjin is one of the four municipalities in China. Each record in our dataset contains the id, text description, timestamp, location and assigned tags of a complaint. We collect the complaints from June 1, 2016 to July 30, 2018. For privacy concerns, we have completely anonymized the dataset before data mining.

Following [31], we preprocess texts of complaints by removing all the punctuation marks from the text and splitting the remaining string into individual words. Common stop words are also discarded. For temporal information, we convert timestamp of each complaint

[2] The source code of STAR is available at https://github.com/jygao97/STAR.

Table 1: Basic statistics of the complaints dataset.

# Complaints	# Words	Avg. words per complaint
1,005,985	90,878	12.9
# Cells	**# Tags**	**# Layers of taxonomy**
6,205	163	4

into a 24-dim one-hot vector indicating which hour it belongs to. For spatial information, we partition the city into a grid map and map location of each complaint to the cell it belongs to. The side length of each cell is set to 1 kilometer following [12]. The basic statistics of the dataset are shown in Table 1.

4.1.2 Baselines. To demonstrate the effectiveness of our proposed STAR, we select seven competitive methods for comparison:

- **LDA** [19] recommends tags for textual content based on Latent Dirichlet Allocation (LDA).
- **Maxide** [41] is a traditional multi-label learning method for tag recommendation, which makes predictions based on speedup matrix completion with side information.
- **Kim CNN** [16] is first proposed for sentence classification. Here it can be used to learn representations for complaints with CNN and then recommend tags.
- **LSTM** is a widely-used variant of RNN. Similar to [21], we perform an average pooling operation on the hidden vectors at each position of LSTM as the representation of complaints.
- **TLSTM** [21] is a novel attention-based LSTM model which incorporates topic modeling into the LSTM architecture through an attention-mechanism.
- **ABC** [9] adopts an attention-based CNN architecture for tag recommendation, which models the textual content with a local attention channel and a global channel.
- **iTag** [31] is an integrated model which jointly considers sequential text modeling, tag correlation and content-tag overlapping in a coherent encoder-decoder framework.

4.1.3 Evaluation Metrics. To evaluate the performance of models, we adopt the widely-used **F1** (the threshold is set to 0.5) and Precision-Recall **AUC** that jointly consider precision and accuracy. A higher F1 or AUC indicates a better performance.

4.1.4 Implementation Details. Our STAR model is implemented in Pytorch[3]. We randomly split the dataset into training (70%), validation (15%) and test (15%) sets. We tune hyper-parameters on the validation set and report the performance on the test set. For all methods, we use the Adam optimizer [17] with an initial learning rate of 0.001. The co-efficient of L_2 norm regularization is fixed to 0.001 and the batch size is fixed to 256. The word embedding size d_w is fixed to 128. The above settings are for fair consideration. For other hyper-parameters of baselines, we directly reuse them if reported by their authors. Otherwise, we tune them on the validation set. After parameter tuning, we set the number of topics in LDA and TLSTM to 60. The filter sizes in Kim CNN and STAR are set to $\{1, 2, 3, 4, 5\}$, each size with 80 filters. λ_A, λ_B, λ_{inner}, λ_{tax} and λ_{smo} of STAR are set to 1.0, 1.0, 1.5, 0.1, and 0.01 respectively. We use ReLU [23] for activation functions in STAR as it performs best on the validation set. Each experiment is repeated twenty times and we report the average and standard deviation as the result.

Table 2: Comparison among different models. Best results are highlighted in bold. p-value is the probability of no significant difference with STAR on both F1 and AUC by t-test.

Models	F1 (%)	Impv.	AUC (%)	Impv.	p-value
STAR	**81.6 ± 0.1**	-	**89.3 ± 0.1**	-	-
LDA	37.4 ± 0.1	+118.2%	62.4 ± 0.1	+43.1%	$< 10^{-3}$
Maxide	57.5 ± 0.1	+41.9%	70.6 ± 0.0	+26.5%	$< 10^{-3}$
Kim CNN	77.6 ± 0.3	+5.2%	86.4 ± 0.1	+3.4%	$< 10^{-3}$
LSTM	77.0 ± 0.5	+6.0%	85.6 ± 0.3	+4.3%	$< 10^{-3}$
TLSTM	79.8 ± 0.3	+2.3%	87.9 ± 0.2	+1.6%	$< 10^{-3}$
ABC	79.0 ± 0.2	+3.3%	87.5 ± 0.1	+2.1%	$< 10^{-3}$
iTag	67.8 ± 1.0	+20.4%	-	-	$< 10^{-3}$

4.2 Performance Comparison (RQ1)

We report the performance of all models in Table 2. The improvements of STAR compared with baselines are also listed. Note that iTag does not output probability for each candidate tag, which prevents us from calculating its AUC. After analyzing the results, we have the following observations.

First, methods based on deep learning (Kim CNN, LSTM, TLSTM, ABC, iTag, and Ours) significantly outperform those traditional methods (LDA and Maxide). The improvement on average is 62.6% on F1 and 31.3% on AUC. This is because deep neural networks can better capture the semantics of complaints than bag-of-words or topic modeling used by traditional methods. It also demonstrates the necessity of adopting neural networks in our textual channel which effectively encodes textual content of complaints.

Second, TLSTM and ABC perform better than Kim CNN and LSTM by 2.7% on F1 and 2.0% on AUC. We attribute it to the fact that the attention mechanism applied by ABC and TLSTM allows them to focus on words that are more important in recommending suitable tags. We notice that iTag performs worse than LSTM. It makes sense since iTag overemphasizes the sequential order of tags which does not matter in this task. It also focuses on modeling content-tag overlapping phenomenon [31] which is rare in citizen complaints. These improper assumptions of iTag hurt its performance.

Third, our proposed STAR achieves the best performance among all models, outperforming the second best one by 2.3% on F1 and 1.6% on AUC. Its improvements over baselines are all statistically significant. Sharing the same textual channel with Kim CNN, STAR outperforms it by 5.2% on F1 and 3.4% on AUC. We attribute the superiority of STAR to its two properties. First, STAR recommends tags in a taxonomy-aware way so that coarse-grained representation of complaints can be gradually refined in different levels from general to specific. Second, STAR exploits correlations between tags and spatio-temporal information of complaints, which makes it more comprehensive and robust than models totally based on textual content and yields better performance consequently.

4.3 Ablation Study (RQ2)

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

- **STAR without taxonomy (V1)** removes the modeling of tag taxonomy and directly considers candidate tags T_{leaf}.

[3] https://pytorch.org/

Table 3: Comparison among variants of STAR. Best results are highlighted in bold. p-value is the probability of no significant difference with STAR on both F1 and AUC by t-test.

Variants	F1 (%)	AUC (%)	p-value
STAR	**81.6 ± 0.1**	**89.3 ± 0.1**	-
STAR without taxonomy	77.6 ± 0.2	86.5 ± 0.1	$< 10^{-3}$
STAR with channel A only	81.1 ± 0.1	89.2 ± 0.0	$< 10^{-3}$
STAR with channel B only	14.2 ± 0.4	25.7 ± 0.2	$< 10^{-3}$
STAR without fusion	81.1 ± 0.1	89.2 ± 0.1	$< 5 \times 10^{-2}$
STAR with constant fusion	81.0 ± 0.1	89.0 ± 0.1	$< 10^{-3}$

- **STAR with channel A only (V2)** removes the spatio-temporal channel and is solely based on the textual content.
- **STAR with channel B only (V3)** removes the textual channel and is totally based on spatio-temporal information.
- **STAR without fusion (V4)** feeds the concatenation of textual and spatio-temporal representations into the recommendation module. Thus it does not require a fusion module.
- **STAR with constant fusion (V5)** fuses results based on two channels with a fixed weight w. Here w is empirically set to 0.9 so that more attention can be paid to textual content.

Comparison results are listed in Table 3, from which we make the following conclusions.

Effectiveness of being taxonomy-aware. Compared with V1, STAR achieves 5.2% higher F1 and 3.2% higher AUC. It demonstrates the effectiveness of incorporating the tag taxonomy. Such hierarchical knowledge allows our model to capture semantic correlations between tags and refine the representation of complaints in different levels, thus enabling better tag recommendation.

Effectiveness of spatio-temporal information. The improvement of STAR over V2 shows that spatio-temporal information can act as a useful complement to textual content. However, the poor performance of V3 suggests that merely modeling spatio-temporal information of complaints is far from enough. It is reasonable since complaints may happen at the same location and same time slot but describe totally different issues. Overall, comparison with V2 and V3 demonstrates the necessity of jointly modeling textual content and spatio-temporal information via two parallel channels in STAR.

Effectiveness of adaptive fusion. The superiority of STAR over V4 shows that the effects of textual content and spatio-temporal information on recommendation should be modeled independently and then fused together. Simply concatenating and feeding them into the recommendation module will worsen the performance. We also observe that STAR outperforms V5 which uses a fixed weight in the fusion module. It is because the importance of spatio-temporal correlations for tags may vary from one to another and constant weighting fails to handle this problem properly. It further verifies our design of dynamic fusion that adaptively discriminates contributions of textual content and spatio-temporal information.

4.4 Parameter Sensitivity Analysis (RQ3)

We start by exploring how the performance of STAR changes with the varying amounts of training data. We then analyze the impact of word embedding size d_w. Moreover, we study how the weights of different loss terms affect performance. When conducting parameter sensitivity analysis, we set other hyper-parameters to values described in Section 4.1.4.

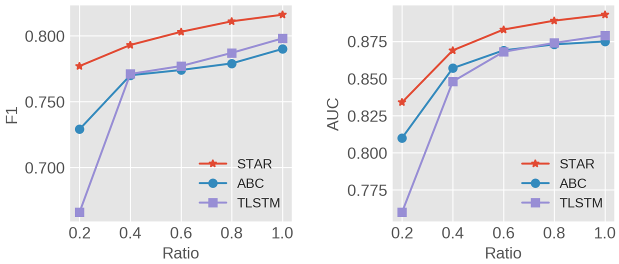


Figure 7: Performance of STAR w.r.t different ratios of training data (compared with two competitive baselines).

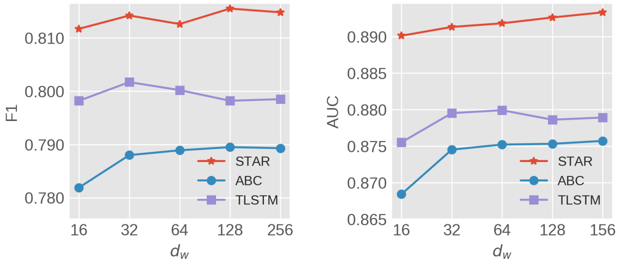


Figure 8: Performance of STAR w.r.t different word embedding sizes (compared with two competitive baselines).

4.4.1 *Varying the amount of training data.*

We take 20%, 40% , 60% and 80% of the complete training data as four new training datasets. Two competitive baselines (TLSTM and ABC) are used for comparison. The results are shown in Figure 7. We can see that STAR consistently achieves the best performance with varying amount of training data, outperforming TLSTM by 16.7% on F1 when the ratio is set as 20%. It demonstrates the robustness of STAR against data insufficiency. We owe it to the incorporation of the tag taxonomy which contains semantic correlations of tags. Such prior knowledge is especially useful when the low incidence of some tags in the training data provides little learning opportunity to complex models. We also notice that the performance of TLSTM drops significantly when encountering data insufficiency, which may be caused by the smooth nature of LSTM [11].

4.4.2 *Varying the word embedding size d_w.*

In our model, an embedding layer is applied to encoding textual content of citizen complaints. To see the effect of embedding size on the performance, we vary it in the set $\{16, 32, 64, 128, 256\}$. ABC and TLSTM are also chosen for comparison. As shown in Figure 8, STAR constantly outperforms baselines with varying d_w, which further demonstrates the robustness of our approach. We also observe that increasing d_w of models does not necessarily improve the performance of tag recommendation because too many parameters in the embedding layer may lead to overfitting.

4.4.3 *Varying the weight λ_{inner} of $\mathcal{L}_{inner}$.*

Although we only recommend leaf tags for complaints in practice, our taxonomy-aware model additionally focuses on making predictions for inner tags by considering $\mathcal{L}_{\text{inner}}$ in (13). To study the impact of $\mathcal{L}_{\text{inner}}$, we vary its weight in $\{0, 0.5, 1.0, 1.5, 2.0\}$. Figure 9 shows that STAR achieves the best performance when λ_{inner} is 1.5. Smaller or larger λ_{inner} tends to hurt the performance. It is reasonable since we have to balance the emphasis put on $\mathcal{L}_{\text{leaf}}$ and $\mathcal{L}_{\text{inner}}$. Too small or large

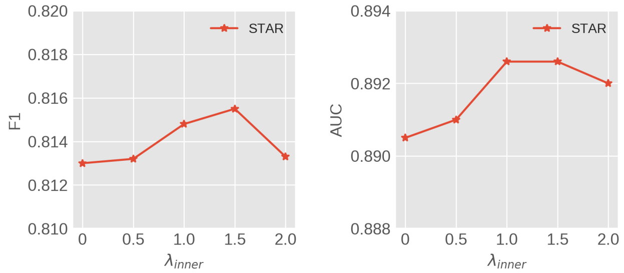


Figure 9: Performance of STAR w.r.t weights of $\mathcal{L}_{\text{inner}}$.

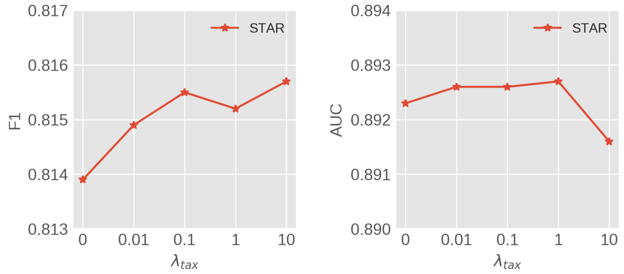


Figure 10: Performance of STAR w.r.t weights of $\mathcal{L}_{\text{tax}}$.

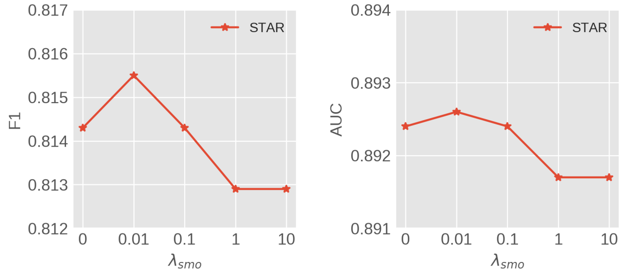


Figure 11: Performance of STAR w.r.t weights of $\mathcal{L}_{\text{smo}}$.

λ_{inner} may make one of them dominate the learning process and yield unsatisfying model performance.

4.4.4 Varying the weight λ_{tax} of $\mathcal{L}_{tax}$. We penalize the misalignment between predictions on parents and children by further considering $\mathcal{L}_{\text{tax}}$ in (13). We vary its weight λ_{tax} in $\{0, 0.01, 0.1, 1, 10\}$ and show results in Figure 10. It can be seen that the best F1 and AUC are achieved when $\lambda_{\text{tax}} = 0.1$ and 1.0 respectively. The performance degrades when $\lambda_{\text{tax}} = 0$. This demonstrates the importance of involving alignment between parents and children. Such design allows STAR to predict in a way that is coherent with hierarchical knowledge in the taxonomy of tags.

4.4.5 Varying the weight λ_{smo} of $\mathcal{L}_{smo}$. Timestamps and locations of complaints are encoded via two embedding matrices in the spatio-temporal channel. We further encourage the smoothness of embedding matrices by imposing $\mathcal{L}_{\text{smo}}$ on model learning. To study its influence, we vary its weight λ_{smo} in $\{0, 0.01, 0.1, 1, 10\}$. As shown in Figure 11, the best performance of STAR is achieved when $\lambda_{\text{smo}} = 0.01$. Both F1 and AUC degenerate when $\lambda_{\text{smo}} = 0$, i.e., not considering $\mathcal{L}_{\text{smo}}$ at all. Thus we conclude that it is necessary to consider the smoothness prior, which helps STAR learn better embedding matrices for spatio-temporal information. We also note that too large λ_{smo} worsens the performance. It makes sense since too much emphasis on $\mathcal{L}_{\text{smo}}$ enforces the embeddings of time slots and cells to be identical and thus the entire spatio-temporal channel becomes meaningless.

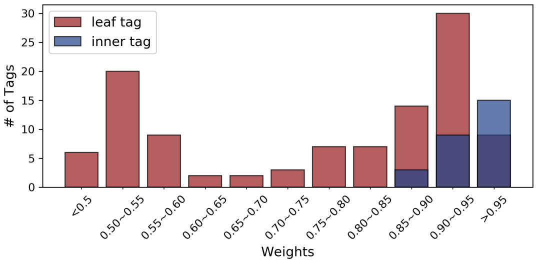


Figure 12: Histogram of learned fusion weights for both inner tags (blue) and leaf tags (red).

4.5 Study on the Fusion Module (RQ4)

In (14) of Section 3.5, we combine predictions based on textual and spatio-temporal information in a tag-specific manner, where $w_l \in [0, 1]$ is the fusion weight learned for tag t_l. The larger w_l is, the more textual information contributes to the final recommendation of t_l and vice versa. In this section, we first give an overview of the distribution of w_l for all tags. Then we conduct case studies to illustrate that STAR is able to reasonably determine w_l.

4.5.1 Overview of all weights. The distribution of w_l for $t_l \in \mathcal{T}$ is illustrated as a histogram in Figure 12. We observe that w_l mainly distributes between 0.50 and 1.0, showing that textual information consistently plays an important role in tag recommendation. It is reasonable since C_w of an complaint often describes the concerned issue in details and is indispensable for tag recommendation, which corresponds with the bad performance of V3 in Section 4.3. We also notice that all w_l of inner tags are larger than 0.80, which makes sense since spatio-temporal correlations for inner tags (coarse-grained) are often weaker than leaf tags (fine-grained). Thus spatio-temporal information contributes less to prediction for inner tags.

Another interesting observation is that there are two dense regions of w_l for leaf tags, the one around 0.55 and the one around 0.90, corresponding to tags that are highly correlated and weakly correlated with spatio-temporal information. In the following, we further study examples of tags with large or small w_l.

4.5.2 Case studies of weights. In Table 4, we list several tags of interest, i.e. with either large (∼ 0.90) or small (∼ 0.50) fusion weights w_l. Basically, tags that are highly correlated with certain locations usually have small w_l. For example, complaints about *vessel inspection*, *port management* mostly take place in the area of harbor and thus spatial information is of great significance in recommending these tags. On the contrary, when complaints may happen in a rather large area, e.g., *air pollution*, *posting service*, etc, recommendation relies relatively more on the textual information.

Among the listed tags, we notice two intriguing examples. First, *noise pollution*, which is closely correlated with temporal information (often happens at midnight), has a fusion weight of 0.9455, indicating that recommendation of this tag largely depends on textual information. After checking raw complaints tagged with *noise pollution*, we explain that corresponding C_w often directly

Table 4: Examples of tags of large or small fusion weights.

Tags of large weights	w_l	Tags of small weights	w_l
city sanitation	0.9682	port management	0.4996
real estate	0.9526	parking lot	0.4999
air pollution	0.9517	water carriage	0.5030
parking	0.9487	vessel inspection	0.5047
noise pollution	0.9455	overspeed	0.5055
consumption card	0.9202	water conservancy	0.5115
posting service	0.9025	railway transportation	0.5332
landscaping	0.8974	irrigation	0.5797

writes like *"...a large noise outside ..."*, making merely textual information effective enough for precise tag recommendation and dominate the task. The second is a comparison made between the two "parking" related tags. For *"parking lot"* which refers to a particular type of location, the tag recommendation is largely affected by spatio-temporal information with $w_l \sim 0.50$. On the other hand, for *"parking"* which describes an event which may happen at many places (e.g., parking lots or roadside anywhere), the recommendation mainly depends on the complaint text with $w_l \sim 0.95$ as it is weakly correlated with spatio-temporal information.

The above study verifies that tags are not equally correlated with spatio-temporal information and thus it is necessary to adaptively determine fusion weights in a tag-specific manner. It also shows that STAR learns reasonable weights in the fusion module.

5 RELATED WORK

In this section, we review the studies related to our work.

5.1 Mining Citizen Complaints

In recent years, a number of citizen complaint platforms have been deployed to facilitate communication between citizens and governments [4, 14, 28]. Since such complaints reveal underlying problems within cities, there is now a significant research interest in mining citizen complaints to obtain valuable insights. For example, Zheng et al. [47] propose to diagnose the noise pollution of a city based on the complaints data. Zheng et al. [48] also attempt to detect collective anomalies with the aid of citizen complaints. Besides, Zhao et al. [46] propose to predict crimes by integrating citizen complaints with other data sources such as human mobility and point of interests. However, all of these studies require citizen complaints to be correctly organized into fine-grained categories or given suitable tags describing their content. In fact, it is painstaking to manually tag citizen complaints and the tagging quality cannot be guaranteed under many circumstances. Thus we aim to automatically recommend suitable tags for citizen complaints, which can benefit above downstream applications and is of great importance especially under the context of intelligent city management.

5.2 Tag Recommendation

Existing work on tag recommendation can be divided into collaborative filtering methods and content-based methods [31, 39].

Collaborative filtering methods. Methods in this class aim to employ users' tagging histories (i.e., user-item-tag tuples) and recommend tags to users in a personalized manner. Typically, Symeonidis et al. [30] construct user-item-tag tensors and model personalized tag recommendation as a tensor factorization problem. Rendle et al. [26] further incorporate pairwise rankings into tensor factorization. Fang et al. [5] exploit Gaussian radial basis function to increase the capacity of Canonical Decomposition. It is also modeled as a link prediction problem in a heterogeneous graph by [6, 45]. However, collaborative filtering methods only consider a fixed set of items and fail to recommend tags for new content. Thus, we focus on content-based methods when recommending tags for citizen complaints which cannot be determined beforehand.

Content-based methods. Content-based methods try to recommend suitable tags by directly modeling the content. There exist studies towards various types of content, e.g., videos [32], audios [7], images [25, 43], and text [24]. We mainly pay attention to those for textual content since citizen complaints are often in this format. Ramage et al. [24] and Krestel et al. [19] propose to recommend tags based on topic models. Wu et al. [38, 39] further consider the tag-content relevance phenomenon. Many recent studies apply neural networks to learn representations for the textual content, which have achieved encouraging results. Gong and Zhang [9] adopt a novel attention-based CNN architecture that consists of global and local channels. Li et al. [21] propose topical attention-based LSTM that incorporates topic distributions into sequential modeling of the content. Tang et al. [31] propose a seq2seq method that jointly models tag correlation and content-tag overlapping.

To improve the recommendation performance, some studies attempt to combine the content information with other types of data. For example, Gong et al. [10] additionally model types of tags as a hidden variable into their DPMM (Dirichlet Process Mixture Models). Besides, Ma et al. [22] and Zhang et al. [42] propose to introduce temporal information for improvement. However, the taxonomy of tags utilized in our approach is ignored by existing studies on tag recommendation, though it exists widely in real-world systems and is easy to obtain [13, 15]. Based on the spatio-temporal characteristic of citizen complaints, we further incorporate spatio-temporal information into our proposed model.

5.3 Taxonomy-Aware Recommendation

Since the taxonomy data is available in many scenarios (e.g., music taxonomy [18], product taxonomy [13], and aspect taxonomy [8]), it has been successfully exploited in several recommendation tasks. For example, Koenigstein et al. [18] propose to utilize item taxonomy for music recommendation. Kanagal et al. [15] combine the taxonomy data with latent factor models to improve sequential recommendation. Huang et al. [13] further propose taxonomy-aware multi-hop reasoning networks for sequential recommendation. Wang et al. [35] and Gao et al. [8] propose to generate appropriate explanations for recommendation results with the aid of the taxonomy data. Different from these approaches, we explicitly incorporate the taxonomy data by performing hierarchical recommendation under a novel taxonomy constraint. In this way, the recommendation process is guided by the structured knowledge and representations of complaints are gradually refined, thus improving the performance of tag recommendation.

6 CONCLUSION

In this paper, we propose a novel spatio-temporal taxonomy-aware recommendation model STAR to recommend tags for citizen complaints, which jointly incorporates spatio-temporal information and the taxonomy of candidate tags. Specifically, STAR employs two parallel channels to learn textual and spatio-temporal representations of complaints, and then feed them into chained neural networks that perform hierarchical recommendation under the taxonomy constraint. An adaptive fusion module is further proposed to integrate results in a tag-specific manner. Extensive experiments on a real-world dataset demonstrate that STAR outperforms state-of-the-art methods significantly. The effectiveness of key components in STAR is also verified through ablation studies.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61772045).

REFERENCES

[1] Fabiano M Belém, Jussara M Almeida, and Marcos A Gonçalves. 2017. A survey on tag recommendation methods. *Journal of the Association for Information Science and Technology* 68, 4 (2017), 830–844.

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *WWW*. 1583–1592.

[3] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. GRAM: graph-based attention model for healthcare representation learning. In *SIGKDD*. ACM, 787–795.

[4] Kevin C Desouza and Akshay Bhagwatwar. 2012. Citizen apps to solve complex urban problems. *Journal of Urban Technology* 19, 3 (2012), 107–136.

[5] Xiaomin Fang, Rong Pan, Guoxiang Cao, Xiuqiang He, and Wenyuan Dai. 2015. Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. In *AAAI*.

[6] Wei Feng and Jianyong Wang. 2012. Incorporating Heterogeneous Information for Personalized Tag Recommendation in Social Tagging Systems. In *SIGKDD*. ACM, 1276–1284.

[7] Frederic Font, Joan Serrí, and Xavier Serra. 2014. Class-based Tag Recommendation and User-based Evaluation in Online Audio Clip Sharing. *Knowledge-Based Systems* 67 (2014), 131–142.

[8] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable Recommendation Through Attentive Multi-View Learning. In *AAAI*.

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

[10] Yeyun Gong, Qi Zhang, and Xuanjing Huang. 2015. Hashtag recommendation using dirichlet process mixture models incorporating types of hashtags. In *EMNLP*. 401–410.

[11] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards A Deep and Unified Understanding of Deep Neural Models in NLP. In *ICML*.

[12] Hsun-Ping Hsieh, Shou-De Lin, and Yu Zheng. 2015. Inferring Air Quality for Station Location Recommendation Based on Urban Big Data. SIGKDD.

[13] Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-Aware Multi-Hop Reasoning Networks for Sequential Recommendation. In *WSDM*. ACM, 573–581.

[14] Rocío B Hubert, Ana G Maguitman, Carlos I Chesñevar, and Marcos A Malamud. 2017. CitymisVis: a Tool for the Visual Analysis and Exploration of Citizen Requests and Complaints. In *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance*. ACM, 22–25.

[15] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. 2012. Supercharging Recommender Systems Using Taxonomies for Learning User Purchase Behavior. *PVLDB* 5, 10 (2012), 956–967.

[16] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

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

[18] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Recsys*. 165–172.

[19] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent dirichlet allocation for tag recommendation. In *Recsys*. 61–68.

[20] Agus Trisnajaya Kwee, Meng-Fen Chiang, Philips Kokoh Prasetyo, and Ee-Peng Lim. 2018. Traffic-cascade: Mining and visualizing lifecycles of traffic congestion events using public bus trajectories. In *CIKM*. ACM, 1955–1958.

[21] Yang Li, Ting Liu, Jing Jiang, and Liang Zhang. 2016. Hashtag recommendation with topical attention-based LSTM. COLING.

[22] Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2014. Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter. In *CIKM*. ACM, 999–1008.

[23] Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. (2010), 807–814.

[24] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*. 248–256.

[25] Yogesh Singh Rawat and Mohan S Kankanhalli. 2016. ConTagNet: Exploiting User Context for Image Tag Recommendation. (2016), 1102–1106.

[26] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2009. Learning optimal ranking with tensor factorization for tag recommendation. In *SIGKDD*. 727–736.

[27] Paul Seitlinger, Dominik Kowald, Christoph Trattner, and Tobias Ley. 2013. Recommending tags with a model of human categorization. In *CIKM*. ACM, 2381–2386.

[28] Po-Ling Sun, Cheng-Yuan Ku, and Dong-Her Shih. 2015. An implementation framework for E-Government 2.0. *Telematics and Informatics* 32, 3 (2015), 504–520.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.

[30] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. 2008. Tag recommendations based on tensor dimensionality reduction. In *Recsys*. 43–50.

[31] Shijie Tang, Yuan Yao, Suwei Zhang, Feng Xu, Tianxiao Gu, Hanghang Tong Tong, Yan Xiaohui, and Jian Lu. 2019. An Integral Tag Recommendation Model for Textual Content. In *AAAI*.

[32] G. Toderici, H. Aradhye, M. PasÃğa, L. Sbaiz, and J. Yagnik. 2010. Finding meaning on YouTube: Tag recommendation and category discovery. In *CVPR*. 3447–3454.

[33] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.

[34] Leye Wang, Daqing Zhang, Animesh Pathak, Chao Chen, Haoyi Xiong, Dingqi Yang, and Yasha Wang. 2015. CCS-TA: Quality-guaranteed Online Task Allocation in Compressive Crowdsensing. In *Ubicomp*. ACM.

[35] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. TEM: Tree-enhanced Embedding Model for Explainable Recommendation. In *WWW*. 1543–1552.

[36] Jônatas Wehrmann, Rodrigo C Barros, Silvia N das Dôres, and Ricardo Cerri. 2017. Hierarchical multi-label classification with chained neural networks. In *Proceedings of the Symposium on Applied Computing*. ACM, 790–795.

[37] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *ICML*. 5225–5234.

[38] Yong Wu, Shengqu Xi, Yuan Yao, Feng Xu, Hanghang Tong, and Jian Lu. 2018. Guiding supervised topic modeling for content based tag recommendation. *Neurocomputing* 314 (2018), 479–489.

[39] Yong Wu, Yuan Yao, Feng Xu, Hanghang Tong, and Jian Lu. 2016. Tag2word: Using tags to generate words for content based tag recommendation. In *CIKM*. ACM, 2287–2292.

[40] Chang Xu, Dacheng Tao, and Chao Xu. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* (2013).

[41] Miao Xu, Rong Jin, and Zhi-Hua Zhou. 2013. Speedup Matrix Completion with Side Information: Application to Multi-label Learning. In *NIPS*. 2301–2309.

[42] Qi Zhang, Yeyun Gong, Xuyang Sun, and Xuanjing Huang. 2014. Time-aware personalized hashtag recommendation on social media. In *COLING*. 203–212.

[43] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. 2017. Hashtag Recommendation for Multimodal Microblog Using Co-attention Network. In *IJCAI*. 3420–3426.

[44] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. 2014. Taxonomy discovery for personalized recommendation. In *WSDM*. ACM, 243–252.

[45] Wei Zhao, Ziyu Guan, and Zheng Liu. 2015. Ranking on heterogeneous manifolds for tag recommendation in social tagging services. *Neurocomputing* 148 (2015), 521–534.

[46] Xiangyu Zhao and Jiliang Tang. 2017. Modeling Temporal-Spatial Correlations for Crime Prediction. In *CIKM*. ACM, 497–506.

[47] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014. Diagnosing New York city's noises with ubiquitous data. In *Ubicomp*. ACM, 715–725.

[48] Yu Zheng, Huichu Zhang, and Yong Yu. 2015. Detecting Collective Anomalies from Multiple Spatio-temporal Datasets Across Different Domains. In *SIGSPATIAL*. ACM.