A Collaborative Learning Framework to Tag Refinement for Points of Interest

Jingbo Zhou[1,3], Shan Gou[1,4], Renjun Hu[1], Dongxiang Zhang[5]
Jin Xu[1], Airong Jiang[1,3], Ying Li[2], Hui Xiong[1,3,6,*]
[1]Business Intelligence Lab, Baidu Research [2]Baidu Inc., Beijing, China
[3]National Engineering Laboratory of Deep Learning Technology and Application, China
[4]University of Electronic Science and Technology of China [5]Zhejiang University [6]Rutgers University
{zhoujingbo,goushan,hurenjun01,xujin08,jiangairong,liying}@baidu.com
zhangdongxiang@zju.edu.cn,xionghui@gmail.com

ABSTRACT

Tags of a Point of Interest (POI) can facilitate location-based services from many aspects like location search and place recommendation. However, many POI tags are often incomplete or imprecise, which may lead to performance degradation of tag-dependent applications. In this paper, we study the POI tag refinement problem which aims to automatically fill in the missing tags as well as correct noisy tags for POIs. We propose a tri-adaptive collaborative learning framework to search for an optimal POI-tag score matrix. The framework integrates three components to collaboratively (i) model the similarity matching between POI and tag, (ii) recover the POI-tag pattern via matrix factorization and (iii) learn to infer the most possible tags by maximum likelihood estimation. We devise an adaptively joint training process to optimize the model and regularize each component simultaneously. And the final refinement results are the consensus of multiple views from different components. We also discuss how to utilize various data sources to construct features for tag refinement, including user profile data, query data on Baidu Maps and basic properties of POIs. Finally, we conduct extensive experiments to demonstrate the effectiveness of our framework. And we further present a case study of the deployment of our framework on Baidu Maps.

ACM Reference Format:
Jingbo Zhou[1,3], Shan Gou[1,4], Renjun Hu[1], Dongxiang Zhang[5] and Jin Xu[1], Airong Jiang[1,3], Ying Li[2], Hui Xiong[1,3,6,*]. 2019. A Collaborative Learning Framework to Tag Refinement for Points of Interest. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3292500.3330698

1 INTRODUCTION

Annotating semantic tags to a Point of Interest (POI) is an intriguing problem, which benefits a lot of location-based services [5, 23–25, 28, 31, 32, 36]. For example, in the online map services (like Google Maps and Baidu Maps), users are greatly facilitated by informative tags when searching and exploring new places. Tags also play an important role for POI recommendation since the tags can help to identify the service ability of POIs.

In real-life applications, such as online map services and location-based social networks, tags of many POIs are incomplete or imprecise, especially for those unpopular or newly-established POIs. Previous studies report that approximately 30% of places in Whrrl and Foursquare datasets lack any meaningful textual descriptions [34]. The phenomenon is mainly caused by two reasons: 1) for tag incompleteness, POI tags are mined from comments or annotated by users, however, most POIs do not have adequate users to generate tags; and 2) for tag imprecision, since tags are either mined from text by machine learning models or annotated by users, it is inevitable to bring errors to the tags. In addition, user-generated tags (from comments or by manual annotation) are often biased towards personal perspectives and context cues [11, 21, 26]. It is impractical for common users to annotate the tags of POI comprehensively. In a word, imprecision and incompleteness of tags of POIs probably lead to performance degradation of tag-dependent applications for POIs.

A possible way to make up the missing tags of POIs is to utilize the tag annotation techniques [12, 13, 15, 16, 29, 34] which have gained much attention from researchers in recent years. These techniques generally assume there is a perfect training dataset, and then cast the tag annotation as a classification problem. However, any missing or noisy tag could potentially lead to a biased estimation of the tag annotation model, resulting in suboptimal performances [30]. It is impractical to maintain a large amount of well annotated POIs, especially considering the concept drift phenomenon and the frequent generation of new POIs and tags.

The limitation of existing POI annotation methods motivates us to develop a new framework for tag refinement. We have two observations regarding this problem. First, the initially annotated tags of POIs, despite imperfect, still reveal the primarily relevant tag semantics of the POIs. We can present the relationships between POI and tag as a score matrix where each entry is a score that represents the relevance of a tag to a POI. The partially observed tags of POIs can also be represented as a binary matrix whose element (i, j) is 1 if and only if POI i is annotated with tag t_j. Our insight is that the optimal score matrix should not deviate from the binary matrix too much. Therefore we can adopt a machine learning approach to searching for an optimal POI-tag score matrix.

[*]Corresponding Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '19, August 4–8, 2019, Anchorage, AK, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6201-6/19/08...$15.00
https://doi.org/10.1145/3292500.3330698

Our second observation is that the crowd search behaviors on online map services in a short time session also provide cues for POI tag refinement. For example, when a user decides to have dinner, she/he may search several different restaurants on the maps in a short time interval. All the POIs searched by the user probably have high tag correlation. Even if some tags are missed or annotated incorrectly, such crowd user behaviors may imply the true tags of the POI. We can leverage such phenomenon to complete or correct tags of the POI.

In this paper, we propose a collaborative learning framework, named Tri-Adaptive Collaborative Learning framework (TACL for short), to tackle the tag refinement problem. TACL consists of three components including non-negative matrix factorization (NMF), pair-wise similarity matching and maximum likelihood estimation (MLE). The NMF component of TACL aims to search for an optimal POI-tag score matrix which is consistent with the observed POI-tag matrix. Then in the pair-wise similarity matching component, we devise a siamese structure neural network to model the consistency between feature similarity and POI-tag semantic similarity. Finally, we also adopt an MLE component to train a multi-label classification model to infer the possible tags of POIs. Moreover, TACL is a collaborative learning framework which trains the three components simultaneously on the same data with a unified optimization process. The advantage of collaborative learning is that the consensus of multiple views on the POI-tag matrix from different components provides supplementary information and regularization to each other, alleviating biased estimation caused by incomplete or noisy tags.

Besides, we propose a feature engineering method based on map query data on Baidu Maps. Especially, we build a POI session graph based on user search behaviors. Then we extract the tag refinement related features from the POI session graph. Other features from user profile and POI basic properities (like name, address and alias) are also included for tag refinement.

Our framework is a collaborative learning framework from two perspectives: 1) our framework includes an adaptive model which is collaboratively optimized for tag refinement; 2) the tag refinement is partially based on the features extracted from map query data which reflects the collaborative behavior of human beings on Baidu Maps. We summarize our contribution as follows:

- We propose a tri-adaptive collaborative learning framework to solve the POI tag refinement problem. As far as we know, we are the first to study this problem. The feature engineering method on Baidu Maps query data and user profile data is also discussed
- Extensive experiments validate the effectiveness of our framework which outperforms several competitors. We also report a case study of the deployment of our framework on Baidu Maps.

The rest of the paper is organized as follows. Next, we discuss related work in Section 2, followed by the preliminaries in Section 3. Then we present the details of the feature construction and TACL in Section 4 and Section 5, respectively. Finally we discuss experiments in Section 6 and present an online deployment case study in Section 7. We conclude the paper in Section 8.

2 RELATED WORK

Our work is closely related with tag annotation which is to automatically associate unlabelled or rarely labelled POIs with semantically related tags. The existing studies on POI tag annotation can be divided into two categories: feature-based methods and model-based methods. For the feature-based methods, much effort has been devoted to studying how to extract discriminative features for predicting the tags of places. In [15, 16], Krumm et al. propose a set of manually designed features extracted from publicly available location diaries and individuals' visits. There are also approaches [12, 13] to exploiting the features of user check-in activities and other user behavior data to train a generative probabilistic model to infer tags for POIs.

For model-based methods, different approaches to POI annotation are investigated. The tag annotation for POIs is first studied in [34] which introduces a collective classification approach to feature extraction. The authors merge hundreds of tags into 21 categories to simplify the task. In [38], authors study how to select the most relevant features for POI tag classification. Yang et al. [33] propose an updatable sketching technique to learn compact sketches from user activity streams, and then they use a KNN classifier for inferring the labels of POIs. Wang et al. [29] propose a graph embedding method to learn POI embeddings from a POI-temporal bipartite graph, and then use the POI embedding vectors as input for a multi-class SVM classifier.

Whereas, in this paper, we study the tag refinement problem. Instead of assuming there is a perfect training dataset with unlabelled test set, POI tag refinement assumes the dataset is made up of partially annotated and even incorrectly annotated POIs. It is supposed to fill in missing tags and correct noisy tags for each POI. To the best of our knowledge, we are the first to study the tag refinement problem for POIs.

The research topic of this paper is also related to the image tag completion and refinement problem. Though many algorithms have been proposed for automatic image annotation [1, 6, 10], image tag refinement is treated as an independent problem, which has become an attractive subject of many researches [8, 19–22, 30, 35, 37]. However, the key idea of image tag completion is to utilize the complex image visual features to infer the semantic labels, which is quite different from the POI tag refinement scenarios. Hence, these image tag completion/refinement methods cannot be directly applied in our POI tag refinement problem.

3 PRELIMINARIES

We present preliminaries in this section. We use capital letters (e.g. P and T) to denote matrices, and use lower case letter with arrow (e.g. $\vec{p}_i$ and $\vec{t}_j$) to denote vectors. In particular, we use p_i to denote a POI, and use $\vec{p}_i \in \mathbb{R}^{k_p}$ to denote feature vector of the POI p_i. Similarly, we use t_j to denote a tag, and use $\vec{t}_j \in \mathbb{R}^{k_t}$ to denote the feature vector of the tag t_j. $\mathrm{P} = [\vec{p}_1, \vec{p}_2, ..., \vec{p}_n]^T$ is a POI feature matrix where n is the number of POIs. All unique tags annotated on the POIs are gathered in tag feature matrix $\mathrm{T} = [\vec{t}_1, \vec{t}_2, ..., \vec{t}_m]^T$, where m is the number of unique tags.

The observed tag annotation of all POIs can be presented in a binary *observed POI-tag matrix* $\hat{\mathrm{Y}} \in \mathbb{R}^{n \times m}$ with $\hat{y}_{i,j} = 1$ when POI p_i is annotated with tag t_j, and 0 otherwise. Usually the observed

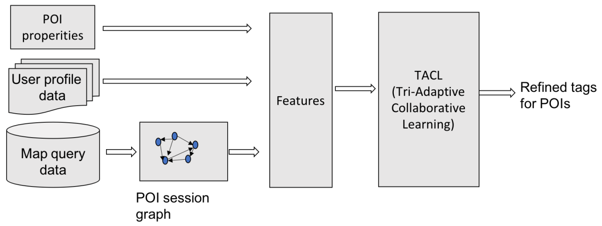


Figure 1: An overview of the POI tag refinement process

POI-tag matrix $\hat{\mathrm{Y}}$ is incomplete and imprecise. Our objective is to predict a *POI-tag score matrix* Y whose element $y_{i,j}$ indicates the confidence score of POI p_i having a tag t_j. Hereafter, we use $\vec{\hat{y}}_{i,*}$ to denote the observed POI-tag vector of POI p_i, and use $\vec{\hat{y}}_{*,j}$ to denote the observed POI-tag vector of tag t_j. It is the same for the cases of $\vec{y}_{i,*}$ and $\vec{y}_{*,j}$. We formally define the POI tag refinement problem as follows:

Definition 3.1 (POI tag refinement). *Given POI feature matrix* $\mathrm{P} = [\vec{p}_1, \vec{p}_2, ..., \vec{p}_n]^T$, *tag feature matrix* $\mathrm{T} = [\vec{t}_1, \vec{t}_2, ..., \vec{t}_m]^T$, *and an observed POI-tag matrix* $\hat{\mathrm{Y}} \in \mathbb{R}^{n \times m}$, *the POI tag refinement is to search an optimal POI-tag score matrix* $\mathrm{Y} \in \mathbb{R}^{n \times m}$ *where* y_{ij} *is the confidence score of assigning tag* t_j *to POI* p_i.

Figure 1 illustrates an overview of the POI tag refinement process. We first extract features of POIs and tags from map query data and user profile data. Then we conduct tag refinement by the TACL framework based on the extracted features. In the following two sections, we first introduce the feature extraction from Baidu's data, and then we present our TACL framework.

4 FEATURES FOR POI TAG REFINEMENT

In this section, we describe how to construct the features of POIs and tags. As shown in Figure 1, we construct features based on POI basic properties (like name and address), user profile data and map query data on Baidu Maps.

For basic properties of POI p_i, we cut the POI string information (which includes name, address and alias names) into words, and look up the word embedding trained on the Chinese corpus from Baidu Baike. Then we average all the vectors to form the basic property feature vector $\vec{p}_i^{\,w}$. In the rest of this section, we introduce how to extract features from user profile data and map query data.

4.1 Data sources

In this paper we use two data sources for tag refinement: map query data and user profile data. Map query data records users' search behaviors from Baidu Maps. We can simply formulate the map query data as a sequence of tuples $MD = \{(u_j, p_i, ts_a)\}$ each of which indicates that a user u_j has an interaction (search, click or view) with the POI p_i at timestamp ts_a.

The user profile data is obtained from a user profile platform that provides features for all Baidu's users, including age, gender, consumption level, job and education level. In Appendix A.3 and Table 5, we list the user profile features used to construct the POI profile. We denote the profile features of user u_j as $\vec{u}_j$.

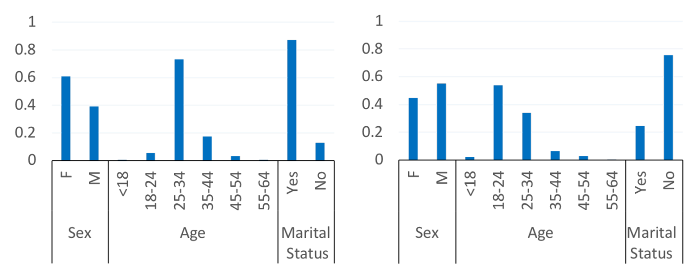
4.2 POI profile features

POI profile features are based on profile features of users who have searched the POI. This is inspired from a widely recognized assumption that the tags of a POI are closely related with its users. Different from existing POI tag annotation methods like [29, 34], we do not define "users" of a POI as people who have check-in to the POI, but people who have *searched* the POI on Baidu Maps. An advantage of this strategy is that the number of search actions is much larger than the one of check-ins.

The POI profile feature vector is the histogram statistics of user distribution. Given a POI p_i and a time interval $[ts_s, ts_e]$, we can retrieve a set of users $U^{p_i}_{[ts_s, ts_e]}$ from map query data MD that $U^{p_i}_{[ts_s, ts_e]} = \{u_j | (u_j, p_i, ts_a) \in MD \wedge ts_s \le ts_a \le ts_e\}$. Then the POI profile features of p_i is the aggregation of the user feature of $U^{p_i}_{[ts_s, ts_e]}$ which can be expressed as:

$$\vec{p_i}^{\,u} = \frac{1}{|U^{p_i}_{[ts_s, ts_e]}|} \sum \vec{u}_j; \; where \; u_j \in U^{p_i}_{[ts_s, ts_e]} \tag{1}$$

where $\vec{p}_i^{\,u}$ denotes the POI profile feature vector.



(a) Qiaohu Happy Island (b) Element Bar

Figure 2: Examples of POI profile features

The POI profile features can reflect characteristics of people interested in the POI. As we can see from Figure 2, for Qiaohu Happy Island (which is a kids garden), the ratio of married and unmarried users is 6.7 : 1, whereas the same ratio of Element Bar (which is a wine bar) is 0.32 : 1. Intuitively, such POI profile can play an important role in distinguishing POIs with different tags.

4.3 Features from POI session graph

We next introduce how to build a POI session graph from the map query data, and then extract features from the POI session graph. A *POI session graph*, denoted by $G^p = \{V^p, E^p\}$, is a directed graph with V^p being a set of POIs and E^p being a set of edges between the POIs, which encodes the user behavior correlation among POIs reflected in the map query data. In general, if there are many users interacting with $p_i \in V^p$ and $p_j \in V^p$ in a short time session, there exists an edge $e^p_{ij} =< p_i, p_j, w_{ij} >\in E^p$ between p_i and p_j with a weight $w_{ij} \in \mathbb{R}$. Here a time session is a short time interval that a user takes interactions (search, click or view) with POIs within a given time frame.

In this study we consider an edge weight as the number of users who interact with the two corresponding POIs in a session. Given a pair of POIs p_i and p_j, a time interval δ_h, we determine whether there is a link between p_i and p_j with the link boolean function:

$$\sigma_h(p_i, p_j|\delta_h, u_k) = \begin{cases} 1 & ((u_k, p_i, ts_a) \in MD) \wedge \\ & ((u_k, p_j, ts_b) \in MD) \wedge (0 < ts_b - ts_a \leq \delta_h) \\ 0 & \text{otherwise} \end{cases}$$

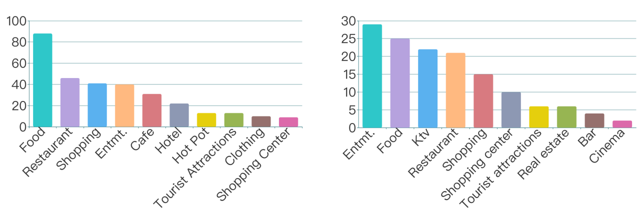
Note that the link has direction between p_i and p_j with the condition $ts_a < ts_b$. In our experiments, we set $\delta_h = 0.5h$ which balances the number of links and the semantic meaning of users in a short time session on Baidu Maps. Given a time interval $[ts_s, ts_e]$, we can calculate the number of links between p_i and p_j by the following function:

$$\kappa_d(p_i, p_j|ts_s, ts_e) = \sum_{\substack{<u', p_i, t'> \in MD \\ <u', p_j, t''> \in MD \\ ts_s \leq t', t'' \leq ts_e}} \sigma_h(p_i, p_j|\delta_h, u') \quad (2)$$

Then the edge weight of $e_{ij}^p =< p_i, p_j, w_{ij} >\in E^p$ within the time interval $[ts_s, ts_e]$ is simply $w_{ij} = \kappa_d(p_i, p_j|ts_s, ts_e)$.

(a) Starbucks (b) Windsor KTV

Figure 3: Examples of nodes and edges in POI session graph



(a) Starbucks (b) Windsor KTV

Figure 4: Examples of features from POI session graph

Finally we present how to extract tag-related features from the POI session graph. The key idea is that, for each POI p_i, we calculate the tag distribution of its neighbors in the POI session graph. Let $OutNer(p_i) = \{p_j| < p_i, p_j, w_{ij} >\in E^p\}$ denote the set of POIs pointed from p_i, and $InNer(p_i) = \{p_j| < p_j, p_i, w_{ji} >\in E^p\}$ denote the set of POIs pointing to p_i. We define the out-degree tag distribution features of p_i based on the POI session graph as:

$$\vec{p}_i^{ot} = \frac{1}{\sum_{p_j \in OutNer(p_i)} w_{ij}} \sum_{p_j \in OutNer(p_i)} w_{ij} \hat{y}_{j,*} \quad (3)$$

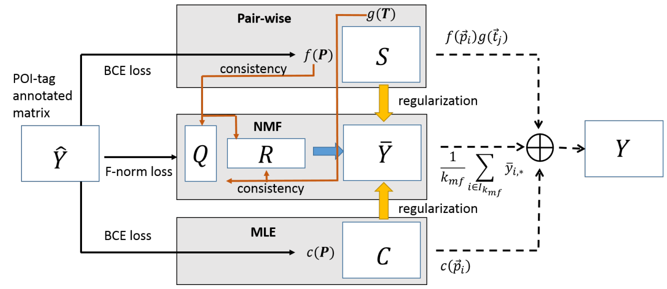


Figure 5: The tri-adaptive collaborative framework for POI tag refinement

Similarly, we can also get the in-degree tag distribution feature on POI session graph of p_i as:

$$\vec{p}_i^{it} = \frac{1}{\sum_{p_j \in InNer(p_i)} w_{ji}} \sum_{p_j \in InNer(p_i)} w_{ji} \hat{y}_{j,*}$$

We show an example of two nodes with their $OutNer(\cdot)$ neighbors in the POI session graph in Figure 3. As we can see from Figure 3, the POI "Starbucks" has very different neighbors from "Windsor KTV". We also illustrate the top-10 features of "Starbucks" and "Windsor KTV" based on their neighbors of $OutNer(\cdot)$ according to Equation 3 in Figure 4.

4.4 Features of POIs and Tags

Here we summarize the features of POIs so far. We have already construct POI features by word embedding $\vec{p}_i^w$, POI profile features $\vec{p}_i^u$, POI session graph features $\vec{p}_i^{ot}$ and $\vec{p}_i^{it}$. Therefore, the feature of p_i is $\vec{p}_i = [\vec{p}_i^w, \vec{p}_i^u, \vec{p}_i^{ot}, \vec{p}_i^{it}]$.

The tag features is aggregated from the POI features. For a tag t_j, whether a POI has the tag t_j can be indicated in a vector $\hat{y}_{*,j} = \hat{Y}[:, j]$ (which means $\vec{\hat{y}}_{*,j}[i] = 1$ if p_i has tag t_j, and $\vec{\hat{y}}_{*,j}[i] = 0$ otherwise), then the features of t_j can be calculated as:

$$\vec{t}_j = \frac{1}{|\hat{y}_{*,j}|} \sum_i \hat{y}_{*,j}[i] \vec{p}_i \quad (4)$$

5 TAG REFINEMENT FRAMEWORK

In this section we first present a framework overview, and then introduce the details of TACL. Finally, we briefly discuss the optimization technique and prediction method based on TACL.

5.1 Framework overview

Figure 5 illustrates three components of TACL for POI tag refinement: non-negative matrix factorization (NMF), pari-wise similarity matching and maximum likelihood estimation (MLE) for multi-label classification. These three components are adaptively trained to optimize the framework. The central component of TACL is the NMF part which can reconstract a POI-tag score matrix by minimizing deviation from the initial observed binary POI-tag matrix. Our insight is that the binary observed POI-tag matrix $\hat{Y}$, despite imperfections, still reveals the primary semantics and functions of each POI. Therefore we use NMF to recover the POI-tag score matrix Y to simultaneously fill missing tags and de-emphasize noisy tags with a limited number of observations in matrix $\hat{\mathrm{Y}}$.

The upper component is the pair-wise matching part which defines the matching similarity between POIs and tags. The bottom component is the MLE part using a multi-label classification model to predict the tags of POIs. Both the upper and bottom components reflect the semantic connection between the POI features and tag features, i.e. we try to identify the candidate tags for each POI based on the information indicating in the POI features and tag features. Note that the pair-wise matching has three regularization terms: the deviation between the similarity matching matrix and NMF recovered POI-tag score matrix $\bar{\mathrm{Y}}$, the similarity consistency between $f(\cdot)$ and NMF matrix QR, and the similarity consistency between $g(\cdot)$ and NMF matrix QR. The bottom component also has a regularization of the deviation between the predicted matrix and NMF recovered POI-tag score matrix $\bar{\mathrm{Y}}$. These regularizations are added to guarantee that the objective of each component is to search the optimal POI-tag score matrix.

TACL can be considered as a multi-view collaborative learning method [27]. The optimal POI-tag score matrix is learned from low rank matrix factorization, pair-wise similarity matching, and multi-label maximum likelihood estimation. All the models are trained simultaneously on the same data, while exploiting commonalities and differences across views in the data with regularization to each other. Then the consensus of the POI-tag score matrix from multiple views of three components provides supplementary information to alleviate biased estimation of the model caused by incomplete or noisy tags. Since the three components are adaptively training in a unified process, we name our framework as "tri-adaptive" collaborative learning framework.

5.2 TACL framework

In this section, we present the detail of our framework. The first component of TACL is low rank matrix factorization. We can assume that the annotated tags of each POI are drawn independently from a multinomial distribution. Our goal is to recover the multinomial distribution from a limited number of observed tags in $\hat{Y}$. It is not easy for this task since the number of parameters to be estimated is significantly larger than the number of annotated tags. Like most topic model approaches, an effective technique to tackle this problem is to assume the tags are sampled from a mixture of a small number of multinomial distributions in latent space, which implies that POI-tag score matrix is low rank [4, 8]. In this paper, we adopt NMF to conduct the low rank matrix reconstruction. The advantage of NMF is that the resulting low-rank factors of NMF lead to physically natural interpretations[18].

Given the matrix $\hat{\mathrm{Y}}$, the goal of NMF is to find two matrices $\mathrm{Q} \in \mathbb{R}^{n\times k}$ and $\mathrm{R} \in \mathbb{R}^{k\times m}$ having only nonnegative entries such that $\hat{\mathrm{Y}} \approx \mathrm{QR}$. Hereafter, we also denote the recovered POI-tag matrix by NMF as $\bar{\mathrm{Y}} = \mathrm{QR}$. The matrices Q and R can be found by solving an optimization problem defined with the Frobenius norm, Kullback-Leibler (KL) divergence or other divergences. Here we use the Frobenius norm as the optimization loss function [17]:

$$\mathcal{L}_{mf} = \|\hat{\mathrm{Y}} - \mathrm{QR}\|_F^2 \ with\ \mathrm{Q} \geq 0, \mathrm{R} \geq 0 \tag{5}$$

For the pair-wise similarity matching component (upper component in Figure 5), we transform the POI and tag features into another feature spaces, and then use the dot product to measure the matching probability between POIs and tags, which are:

$$S = f(\mathrm{P})g(\mathrm{T})^T \tag{6}$$

$$f(\mathrm{P}) = [f(\vec{p}_1), ..., f(\vec{p}_n)]^T \tag{7}$$

$$g(\mathrm{T}) = [g(\vec{t}_1), ..., g(\vec{t}_m)]^T \tag{8}$$

where the combination of $f(\cdot)$ and $g(\cdot)$ is a siamese network with two subnetworks processing the POI and tag in parallel [3]. Here we use Multilayer Perceptron (MLP) to model $f(\cdot)$ and $g(\cdot)$.

The insight of the pair-wise similarity matching method is to optimize $f(\cdot)$ and $g(\cdot)$ that $f(\vec{p}_i)g(\vec{t}_j)$ has a high value if tag t_j belongs to p_i. Different from the traditional siamese network, we have two objectives to optimize $f(\vec{p}_i)g(\vec{t}_j)$. The first objective is to minimize the loss between the pairwise similarity and the observed POI-tag matrix $\hat{\mathrm{Y}}$. In this case we use the binary cross entropy (BCE) to define the loss function:

$$\mathcal{L}_{ps1} = -\sum_{\substack{1\leq i\leq n\\ 1\leq j\leq m}} \Big(\hat{y}_{ij}\log(f(p_i)g(t_j)) + (1-\hat{y}_{ij})\log(1-f(p_i)g(t_j))\Big)$$

The second objective is to regularize the difference between the pairwise similarity matrix and low-rank recovered matrix $\bar{\mathrm{Y}} = \mathrm{QR}$, and the loss function is:

$$\mathcal{L}_{ps2} = \|f(\mathrm{P})g(\mathrm{T}) - \mathrm{QR}\| \tag{9}$$

The reasons to minimize the error between $\bar{\mathrm{Y}}$ and $f(\mathrm{P})g(\mathrm{T})$ are: 1) since $\bar{\mathrm{Y}}$ is partially observed and possibly noisy, solely optimizing $\mathcal{L}_{ps1}$ cannot achieve the purpose for tag refinement; and 2) we can learn to optimize the pair-wise function and matrix factorization collaboratively in a multi-view training process.

Note that $f(\cdot)$ also indicates that the similarity between p_i and p_j. The POI similarity in tag space can be calculated by $\mathrm{S_{pp}} = \bar{\mathrm{Y}}\bar{\mathrm{Y}}^T = \mathrm{QR}(\mathrm{QR})^T \in \mathbb{R}^{n\times n}$. In order to ensure the consistency between the POI similarity in tag space and the MLP network $f(\cdot)$, we add the following regularization term in the loss function:

$$\mathcal{L}_{pp} = \|f(\mathrm{P})f(\mathrm{P})^T - \mathrm{QR}(\mathrm{QR})^T\|_2 \tag{10}$$

By adding the regularization term, we also build the connection between QR and $f(\cdot)$. We will utilize $f(\cdot)$ to make prediction based on QR which will be introduced in Section 5.3.

Similarly, the tag similarity in POI space can be calculated by $\mathrm{S_{tt}} = \bar{\mathrm{Y}}^T\bar{\mathrm{Y}} = (\mathrm{QR})^T\mathrm{QR} \in \mathbb{R}^{m\times m}$, and we can also add the regularization term for the MLP network $g(\cdot)$, which ensures the consistency between the tag similarity in POI space and the learned representation of tags:

$$\mathcal{L}_{tt} = \|g(\mathrm{T})g(\mathrm{T})^T - (\mathrm{QR})^T\mathrm{QR}\|_2 \tag{11}$$

To sum up, the loss function to be optimized for pair-wised similarity matching is:

$$\mathcal{L}_{ps} = \lambda_{ps1}\mathcal{L}_{ps1} + \lambda_{ps2}\mathcal{L}_{ps2} + \lambda_{pp}\mathcal{L}_{pp} + \lambda_{tt}\mathcal{L}_{tt} \tag{12}$$

The bottom component of TACL is maximum likelihood estimation (MLE) part which essentially is a multi-label classification model to enhance the tag refinement quality. Here we use an MLP model to predict the tags of a POI, and suppose the prediction function is $c(\mathrm{P})$. We also adopt to optimize $c(\cdot)$ according to two objectives, as the same with pair-wise similarity matching model. The first objective is to minimize the loss between $c(\cdot)$ and $\hat{\mathrm{Y}}$, and we also use the BCE loss:

$$\mathcal{L}_{c1} = -\sum_{1\le i\le n}\Big(\hat{y}_{i,*}\log(c(p_i)) + (1-\hat{y}_{i,*})\log(1-c(p_i))\Big) \quad (13)$$

The second objective is to minimize the difference between the $c(\cdot)$ and $\bar{\mathrm{Y}} = QR$, and the loss function is:

$$\mathcal{L}_{c2} = \|c(\mathrm{P}) - \bar{\mathrm{Y}}\|_2 \quad (14)$$

Then the loss function of $c(\cdot)$ to be optimized is:

$$\mathcal{L} = \lambda_{c1}\mathcal{L}_{c1} + \lambda_{c2}\mathcal{L}_{c2} \quad (15)$$

Finally, we can summarize the above three components together of TACL into the following optimization problem:

$$\min \mathcal{L} = \lambda_{mf}\mathcal{L}_{mf} + \lambda_{ps}\mathcal{L}_{ps} + \lambda_c\mathcal{L}_c + \lambda_\theta\|\Theta\|_2 \quad (16)$$

where $\|\Theta\|_2$ denotes L_2 regularization on all trainable parameters.

5.3 Optimization and prediction

The formulation in Equation 16 is a quadratic optimization problem since $\mathcal{L}_{mf}$ has nonnegative constraints. Except for QR, other parameters can be optimized by gradient descent. In our model, we adopt the alternating optimization strategy to optimize the parameters. In particular, for each epoch, we optimize the variable of QR first with others fixed by standard quadratic programming. The NMF is optimized by a coordinate-wise algorithm[9], where each unknown variable can be solved sequentially and explicitly as simple quadratic optimization problems. Then we optimize other variables with fixing QR. For the gradient descent optimization, we use the adaptive momentum (ADAM) optimizer [14]. We repeat this procedure until meeting a predefined stop condition.

The final refinement result is an ensemble [7] of the three components. Given a candidate POI p_0, our objective is to generate a tag confidence vector $y_{0,*}$ where each entry $y_{0,j}$ denotes a score that the p_0 has tag t_j. The result of the pairwise component can be obtained by $y^{ps}_{0,*} = f(\vec{p}_0)g(T)^T$, and the one of the MLE model is $y^c_{0,*} = c(\vec{p}_0)$. For the NMF model, we use a collaborative filtering-like method to do such prediction. Given p_0, we first retrieve top k_{mf} POIs from the training dataset (which have the low rank matrix fatorization $\hat{\mathrm{Y}} = \mathrm{QR}$) according to the similarity defined by function $f(\cdot)$, and denote their index in $\hat{\mathrm{Y}}$ as $I_{k_{mf}} = \{i_1, i_2, ..., i_{k_{mf}}\}$. The tag confidence vector by NMF model as: $y^{mf}_{0,*} = \frac{1}{k_{mf}}\sum_{i\in I_{k_{mf}}}\bar{\mathrm{Y}}_{i,*}$. The final prediction result is the ensemble of three components:

$$y_{0,*} = (1-\alpha-\beta)y^{mf}_{0,*} + \alpha y^{ps}_{0,*} + \beta y^c_{0,*} \quad (17)$$

6 EXPERIMENTS

6.1 Datasets and settings

We evaluate the performance of our framework on both Bejing and Chengdu datasets. All the map query data MD and user profile data are collected from Baidu Maps from August 1 2018 to Octorber 31 2018. For Beijing dataset, we use POIs located in six main urban areas of Beijing – Dongcheng, Xicheng, Haidian, Chaoyang, Shijingshan and Fengtai. For Chengdu dataset, we use POIs located in five main urban areas of Chengdu – Qingyang, Jinniu, Wuhou, Chenghua and Jinjiang. The Beijing dataset contains 306K POIs and Chengdu data contains 234K POIs. We summarize the statistics of the data in Table 1.

Table 1: Statistics of map query data and POIs

Dataset	# of map queries	# of POIs	Avg. # of tag
Beijing	50.6M	306K	2.43
Chengdu	21.0M	234K	2.15

We randomly separate the dataset into three folds. One fold consisting of 80% of POIs is used as training data, one fold consisting of 10% of POIs is used as validation data and another fold consisting of 10% of POIs is used as testing data. All experiments are conducted on a GPU-CPU platform with GTX 1080.The program and baselines are implemented in Python 2.7.

We use metrics Average Precision@N (AP@N), Average Recall@N (AR@N), Coverage@N (C@N), Mean Average Precision (MAP@N) and Total Mean Average Precision (MAP@Total[1]) to evaluate our framework. Introduction about the metrics is in Appendix A.1. We compare TACL with the following state-of-the-art methods:

- **TransE** [2] is a method for the prediction of missing relationships in knowledge graphs. We add a "has" relation between POI and tag if a POI has a tag, then use TransE to predict the possible tags of POIs.
- **PPE** (Predictive Place Embedding) [29] is a state-of-the-art POI tag annotation method though graph embedding.
- **TMC** (Tag Completion Algorithm) [30] is a tag completion method for images by searching an optimal tag matrix.
- **NMF** (Non-negative Matrix Factorization) has been widely used in many fields for matrix recovery. Here we use NMF to recover the POI-tag score matrix based on observed binary POI-tag matrix.
- **MLP** (Multilayer Perceptron) is a feedforward neural network that can do multi-label classification. Here we BCE loss to train the MLP, and use the features of POI as the input of MLP.

We conduct performance evaluation of the tag refinement on Beijing dataset and Chengdu dataset with three settings: 1) evaluating on original POI data (Section 6.2); 2) evaluating on POI data with randomly adding noisy tags to 50% of POIs (Section 6.3); and 3) evaluating on POI data with randomly removing a half of tags of 50% POIs (Section 6.4).

6.2 Performance evaluation on original data

Table 2 shows the evaluation results on original data with different metrics. As shown in Table 2, TACL outperforms all baselines. First, TACL's performance on AP, AR, C and MAP can substantially outperform baselines with the same N. Second, with increasing N, all models' performance becomes worse with regard to AP

[1]Since Precision, Recall and Coverage of total result is always 1 for all models, we only report MAP@Total here.

Table 2: Performance(%) comparison between TACL and baselines

Dataset		Beijing						Chengdu					
Model		TransE	PPE	TMC	NMF	MLP	TACL	TransE	PPE	TMC	NMF	MLP	TACL
AP@N	1	11.32	23.49	62.50	80.67	83.25	**87.83**	12.54	31.30	63.06	80.81	84.19	**88.42**
	3	8.42	27.33	40.87	49.25	60.06	**64.03**	8.41	22.97	36.82	45.42	55.50	**58.67**
	5	7.29	26.73	29.13	33.19	40.60	**42.82**	7.12	17.91	25.97	30.27	37.16	**38.79**
AR@N	1	11.05	4.36	28.34	37.72	38.76	**41.10**	12.37	19.33	37.09	46.08	47.73	**49.98**
	3	21.25	16.42	52.52	63.18	76.94	**81.60**	21.53	33.33	57.42	67.70	81.08	**84.92**
	5	29.26	25.56	61.62	70.16	84.96	**89.10**	29.52	42.12	65.08	73.83	88.06	**91.16**
C@N	1	11.32	23.49	62.50	80.67	83.25	**87.83**	12.54	31.30	63.06	80.81	84.19	**88.42**
	3	25.23	19.71	77.63	87.68	91.45	**93.35**	25.20	55.43	81.89	87.58	93.20	**94.68**
	5	34.56	52.57	83.85	91.25	94.42	**95.93**	33.96	65.85	86.43	90.66	95.74	**96.75**
MAP@N	1	11.32	23.49	62.50	80.67	83.25	**87.83**	12.54	31.30	63.06	80.81	84.19	**88.42**
	3	16.67	31.21	68.18	82.89	86.14	**89.40**	17.50	44.24	70.62	83.18	87.58	**90.56**
	5	18.96	38.29	67.52	81.61	85.20	**88.62**	19.63	46.80	70.36	82.09	86.85	**89.91**
	Total	20.77	31.21	54.65	67.58	80.84	**85.72**	22.05	35.39	57.24	70.98	83.39	**87.41**

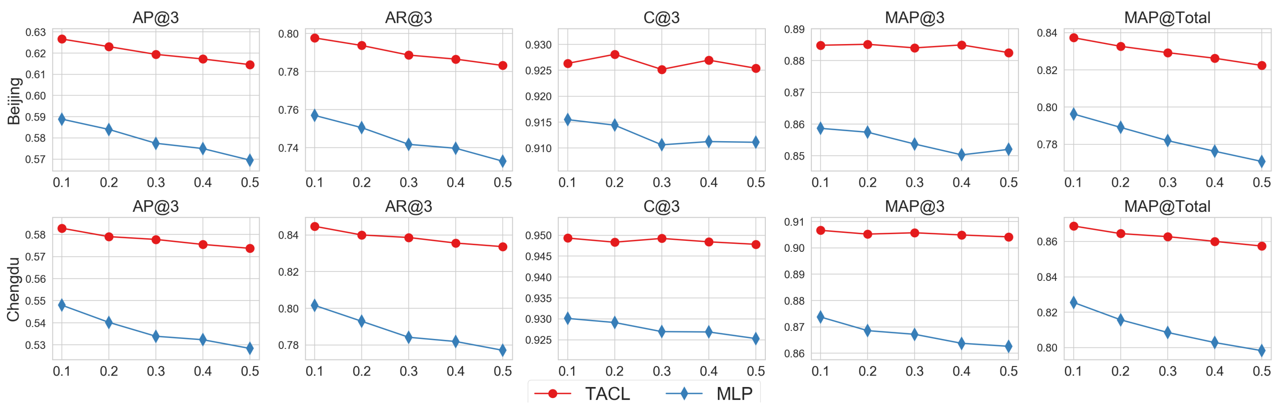


Figure 6: TACL and MLP' s results at different ratio of POIs with noisy tags

and becomes better with regard to AR. It is because the larger N results in the larger denominator when computing the precision of each POI. Meanwhile, the size of intersection of recovered tags and ground truth tags also becomes larger, leading to the numerator being larger when computing recall. Last but not least, TACL and baselines show the same ranks on different metrics on Beijing and Chengdu datasets, and our TACL beats all the baselines consistently. We also find that all models achieve the best performance on MAP when $N = 3$. To make the experiment be more convincing, we add additional experiments of MAP@Total to show the results in extreme situation in the last row of Table 2.

6.3 Performance evaluation with randomly adding noisy tags

In this section, we evaluate TACL on POI data with randomly adding noisy tags. We randomly select α percent of POIs as target POIs, and then add noisy tags to the target POIs. The number of added noisy tags for a POI equals the number of tags of this POI.

We present the experiment result among all baselines with $\alpha = 50\%$ in Table 3. As we can see from Table 3, TACL outperforms all baselines. This demonstrates TACL's better robustness with regard to noisy tags. From Table 3, we can see that MLP is the second strongest competitor. We further compare TACL with MLP with different percent of POIs with noisy tags shown in Figure 6. It is worth noting that, with more POIs with noisy tags, for AP, AR and MAP, MLP's curves drop faster than TACL. Theses results on both Beijing and Chengdu datasets prove that TACL is more robust than MLP with regard to noisy tags.

6.4 Performance evaluation with randomly removing tags

We then examine the effectiveness of our model on Beijing and Chengdu datasets after randomly removing a half of tags of α percent of POIs. Table 4 shows the results under different metrics when $N = 3$ and $\alpha = 50\%$. We observe that the performance of all models becomes worse than the ones with complete tags. However, the proposed TACL is significantly better than all competitors. We remove a half tags of different proportions of POIs to evaluate the effect of data incompletion in Figure 7. We also compare TACL with MLP (which is the second strongest baseline). Figure 7 shows that the performance of both TACL and MLP declines with more

Table 3: Performance(%) evaluation with adding noisy tags to 50% of POIs

Dataset		Beijing						Chengdu					
Model		TransE	PPE	TMC	NMF	MLP	TACL	TransE	PPE	TMC	NMF	MLP	TACL
N=3	AP@N	8.12	18.63	40.55	47.56	56.95	**61.46**	10.91	22.94	38.14	44.48	52.84	**57.37**
	AR@N	21.62	20.51	52.09	61.18	73.29	**78.32**	28.28	33.39	58.76	66.62	77.71	**83.36**
	C@N	24.32	41.26	77.78	87.12	91.11	**92.54**	32.71	55.71	82.95	87.41	92.53	**94.78**
	MAP@N	16.87	30.02	68.75	82.31	85.20	**88.25**	20.89	44.01	71.99	82.72	86.26	**90.42**
MAP@Total		21.12	23.87	54.49	65.30	77.08	**82.24**	24.92	35.20	58.59	69.87	79.83	**85.74**

Table 4: Performance(%) evaluation with randomly removing a half of tags of 50% of POIs

Dataset		Beijing						Chengdu					
Model		TransE	PPE	TMC	NMF	MLP	TACL	TransE	PPE	TMC	NMF	MLP	TACL
N=3	AP@N	1.35	17.45	34.83	48.40	57.97	**63.24**	4.60	21.99	34.50	44.11	53.07	**58.08**
	AR@N	3.84	21.58	45.73	62.54	74.47	**80.66**	11.75	32.57	54.52	66.26	78.08	**84.28**
	C@N	4.06	51.80	78.50	89.07	92.00	**93.70**	13.81	55.68	80.26	87.15	92.89	**94.90**
	MAP@N	1.52	35.84	67.50	83.19	85.44	**88.89**	8.42	38.55	69.68	82.02	85.39	**89.66**
MAP@Total		5.78	23.27	46.87	66.35	78.35	**84.34**	15.23	30.52	53.78	69.09	79.61	**86.12**

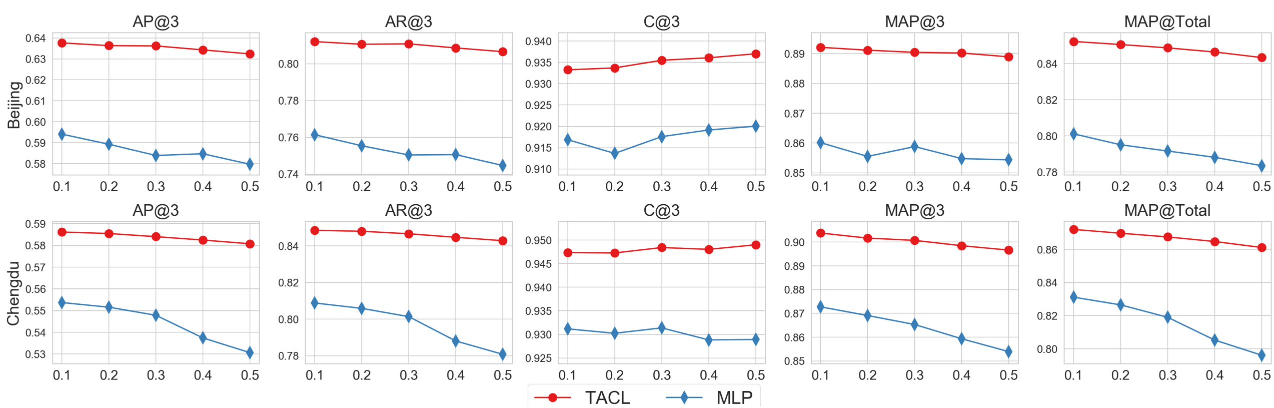


Figure 7: TACL and MLP' s results at different ratio of POIs with incomplete tags

incomplete POIs. However, for each metric, we find that the downtrend of TACL is less than the MLP model with ranging the incomplete POI ratio from 10% to 50%, indicating that TACL is more effective for completing tags of POIs.

7 ONLINE DEPLOYMENT – A CASE STUDY

Our framework has already been applied on Baidu Maps to improve the quality of POI tags. Here we describe one deployed case for tag refinement, "parent-kids" tag completion, to show the usefulness of TACL. A POI with "parent-kids" tag means it is suitable for parent and kids to visit together. Such tag is valuable to convey more information to parents. However, the "parent-kids" tag is rare, and many candidate POIs for "parents-kids" tag are not annotated. Using our TACL framework, we increase the number of POIs with "parent-kids" tag by 55.6%. A Product Manager of Baidu Maps manually checked two hundreds new discovered POIs with the "parent-kids" tag, and concluded that the accuracy of the discovered result is 99.5%. Finally, these new discovered POIs with "parent-kids" tags are deployed online on Baidu Maps on January 7, 2019. In Figure 8, we illustrate two POIs with the new labelled tag of "parent-kids" (in Beijing andChengdu respectively) which are exhibited on Baidu Maps since January 7, 2019. After the deployment, the total click volume of the POIs having "parent-kids" tag is increased by 38.0%.

8 CONCLUSION

In this paper, we study the tag refinement problem for Points of Interest. We propose a collaborative learning framework, called TACL, to tackle the tag refinement problem for POIs. To the best of our knowledge, we are the first to study this problem. The proposed framework contains three components which are non-negative matrix factorization (NMF), pair-wise similarity matching and maximum likelihood estimation (MLE) for multi-label classification. The three components are jointly trained on the same dataset and provide regularization to each other, aiming to search an optimal POI-tag score matrix. Then the consensus of multiple views on the POI-tag matrix from different components can avoid biased estimation of the model caused by incomplete or noisy tags. In addition,

(a) Beileou Theme Park (Beijing) (b) Panda Valley (Chengdu)

Figure 8: Tag refinement example on Baidu Maps

we also propose feature engineering method based on map query data and user profile data. We conducted extensive experiments to demonstrate the effectiveness of our proposed framework on noisy and incomplete data, and present a discussion about the deployed case of TACL's output results on Baidu Maps.

ACKNOWLEDGMENTS

We thank all anonymous reviewers for insightful comments. This research is supported in part by grants from the National Natural Science Foundation of China (No.71531001).

REFERENCES

[1] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M Blei, and Michael I Jordan. 2003. Matching words and pictures. *Journal of machine learning research* 3, Feb (2003), 1107–1135.
[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*. 2787–2795.
[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *NIPS*. 737–744.
[4] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9, 6 (2009), 717.
[5] Gang Chen, Sai Wu, Jingbo Zhou, and Anthony KH Tung. 2014. Automatic itinerary planning for traveling services. *IEEE transactions on knowledge and data engineering* 26, 3 (2014), 514–527.
[6] Qimin Cheng, Qian Zhang, Peng Fu, Conghuan Tu, and Sen Li. 2018. A survey and analysis on automatic image annotation. *Pattern Recognition* 79 (2018), 242–259.
[7] Liang Duan, Charu Aggarwal, Shuai Ma, Renjun Hu, and Jinpeng Huai. 2016. Scaling up Link Prediction with Ensembles. In *WSDM*. 367–376.
[8] Zheyun Feng, Songhe Feng, Rong Jin, and Anil K Jain. 2014. Image tag completion by noisy matrix recovery. In *ECCV*. Springer, 424–438.
[9] Vojtěch Franc, Václav Hlaváč, and Mirko Navara. 2005. Sequential coordinate-wise algorithm for the non-negative least squares problem. In *CAIP*. Springer, 407–414.
[10] K-S Goh, Edward Y Chang, and Beitao Li. 2005. Using one-class and two-class SVMs for multiclass image annotation. *IEEE Transactions on Knowledge and Data Engineering* 17, 10 (2005), 1333–1346.
[11] Scott A Golder and Bernardo A Huberman. 2006. Usage patterns of collaborative tagging systems. *Journal of information science* 32, 2 (2006), 198–208.
[12] Tieke He, Hongzhi Yin, Zhenyu Chen, Xiaofang Zhou, Shazia Sadiq, and Bin Luo. 2016. A spatial-temporal topic model for the semantic annotation of POIs in LBSNs. *TIST* 8, 1 (2016), 12.
[13] Vinod Hegde, Josiane Xavier Parreira, and Manfred Hauswirth. 2013. Semantic tagging of places based on user interest profiles from online social networks. In *ECIR*. Springer, 218–229.
[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[15] John Krumm and Dany Rouhana. 2013. Placer: semantic place labels from diary data. In *UbiComp*. ACM, 163–172.
[16] John Krumm, Dany Rouhana, and Ming-Wei Chang. 2015. Placer++: Semantic place labels beyond the visit. In *PerCom*. IEEE, 11–19.
[17] Da Kuang, Jaegul Choo, and Haesun Park. 2015. Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*. Springer, 215–243.
[18] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788.
[19] Xue Li, Bin Shen, Bao-Di Liu, and Yu-Jin Zhang. 2016. A locality sensitive low-rank model for image tag completion. *IEEE Transactions on Multimedia* 18, 3 (2016), 474–483.
[20] Zijia Lin, Guiguang Ding, Mingqing Hu, Jianmin Wang, and Xiaojun Ye. 2013. Image tag completion via image-specific and tag-specific linear sparse reconstructions. In *CVPR*. 1618–1625.
[21] Dong Liu, Xian-Sheng Hua, Meng Wang, and Hong-Jiang Zhang. 2010. Image retagging. In *MM*. ACM, 491–500.
[22] Dong Liu, Shuicheng Yan, Xian-Sheng Hua, and Hong-Jiang Zhang. 2011. Image retagging using collaborative tag propagation. *IEEE Transactions on Multimedia* 13, 4 (2011), 702–712.
[23] Hao Liu, Ting Li, Renjun Hu, Yanjie Fu, Jingjing Gu, and Hui Xiong. 2019. Joint Representation Learning for Multi-Modal Transportation Recommendation. In *AAAI*. 2787–2795.
[24] Hao Liu, Yongxin Tong, Panpan Zhang, Xinjiang Lu, Jianguo Duan, and Hui Xiong. 2019. Hydra: A Personalized and Context-Aware Multi-Modal Transportation Recommendation System. In *KDD*.
[25] Yanchi Liu, Chuanren Liu, Xinjiang Lu, Mingfei Teng, Hengshu Zhu, and Hui Xiong. 2017. Point-of-interest demand modeling with human mobility patterns. In *KDD*. 947–955.
[26] Krystyna K Matusiak. 2006. Towards user-centered indexing in digital image collections. *OCLC Systems & Services: International digital library perspectives* 22, 4 (2006), 283–298.
[27] Guocong Song and Wei Chai. 2018. Collaborative Learning for Deep Neural Networks. In *NeurIPS*.
[28] Ying Sun, Hengshu Zhu, Fuzhen Zhuang, Jingjing Gu, and Qing He. 2018. Exploring the urban region-of-interest through the analysis of online map search queries. In *KDD*. 2269–2278.
[29] Yan Wang, Zongxu Qin, Jun Pang, Yang Zhang, and Jin Xin. 2017. Semantic Annotation for Places in LBSN through Graph Embedding. In *CIKM*. ACM, 2343–2346.
[30] Lei Wu, Rong Jin, and Anil K Jain. 2013. Tag Completion for Image Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 3 (2013), 716–727.
[31] Yuan Xia, Jingbo Zhou, Jingjia Cao, Yanyan Li, Fei Gao, Kun Liu, Haishan Wu, and Hui Xiong. 2018. Intent-Aware Audience Targeting for Ride-Hailing Service. In *ECML/PKDD*. 136–151.
[32] Mengwen Xu, Tianyi Wang, Zhengwei Wu, Jingbo Zhou, Jian Li, and Haishan Wu. 2016. Demand driven store site selection via multiple spatial-temporal data. In *SIGSPATIAL*. ACM, 40.
[33] Dingqi Yang, Bin Li, and Philippe Cudré-Mauroux. 2016. POIsketch: semantic place labeling over user activity streams. In *IJCAI*. AAAI Press, 2697–2703.
[34] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. 2011. On the semantic annotation of places in location-based social networks. In *KDD*. ACM, 520–528.
[35] Jiaming Zhang, Shuhui Wang, and Qingming Huang. 2017. Location-based parallel tag completion for geo-tagged social image retrieval. *ACM Transactions on Intelligent Systems and Technology* 8, 3 (2017), 38.
[36] Jingbo Zhou, Anthony KH Tung, Wei Wu, and Wee Siong Ng. 2013. A "semi-lazy" approach to probabilistic path prediction in dynamic environments. In *KDD*. 748–756.
[37] Guangyu Zhu, Shuicheng Yan, and Yi Ma. 2010. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *MM*. ACM, 461–470.
[38] Yin Zhu, Erheng Zhong, Zhongqi Lu, and Qiang Yang. 2013. Feature engineering for semantic place prediction. *Pervasive and mobile computing* 9, 6 (2013), 772–783.

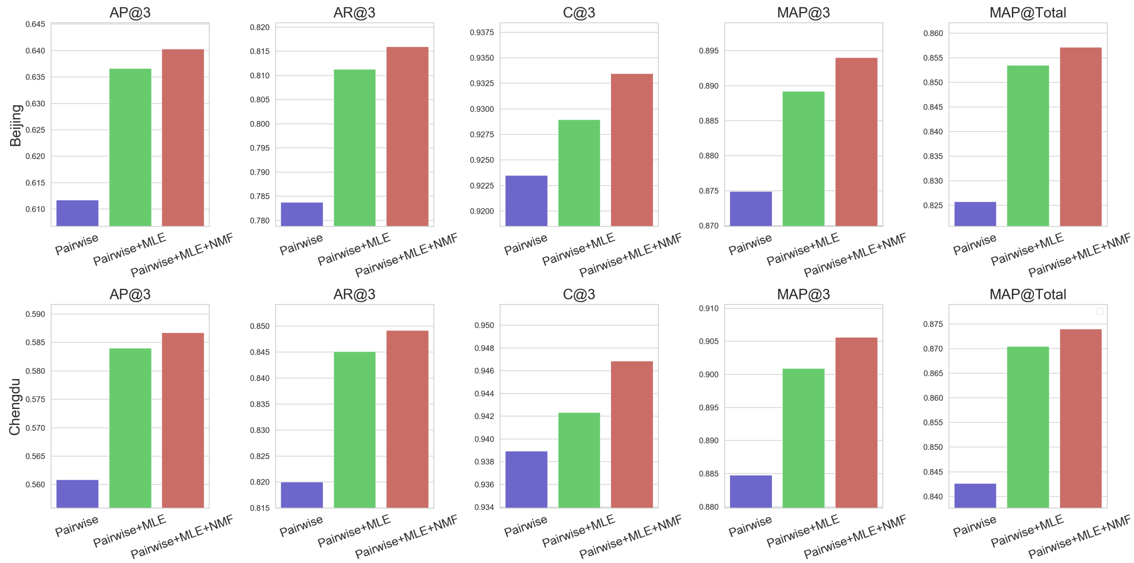


Figure 9: Effectiveness of collaborative learning

Table 5: User profile feature

Category	Attribute	Category	Attribute
Sex	Female	Age	< 18
	Male		18-24
Marital Status	Yes		25-34
	No		...
Interest	Book	Career	Cook
	...		...
Stage	Student	Industry	IT
	...		...
Consumption	Low	Income	≤ 2499
	Medium		25003999
	High		40007999
Education	High School		≥ 8000
	College	Hobby	Fishing
	Bachelor		Hiking
Car	Have car		Cycling
	No car		...

A APPENDIX

A.1 Evaluation Metrics

The experiment resutls of tag refinement are evaluated with the following metrics:

- **Average Precision@N (AP@N)** measures the average percentage of the top N predicted tags that are correct.
- **Average Recall@N (AR@N)** measures the percentage of correct tags that are predicted out of all ground truth tags.
- **Coverage@N (C@N)** measures the percentage of POI with at last one correctly predicted tag.
- **Mean Average Precision (MAP@N)** measures the mean of the average precision scores for each POI of the top N predicted tags. It considers the rank of the predicted tags.
- **Total Mean Average Precision (MAP@Total)** measures mean of the average precision for each POI of all tags.

A.2 Effectiveness of collaborative learning

As we discussed in Section 5, TACL consists of three component: NMF, pairwise and MLE. We conduct an experiment to demonstrate the effectiveness of the collaborative joint learning of the three models. As shown in Figure 9, with adding the components into TACL, the performance of TACL is increasing for all metrics. The results demonstrate that our ensemble framework is effective to obtain better results.

A.3 Table for user profile feature

Table 5 shows the features of user profile data. All the features are aggregated as features of POIs. Hence, the POI profile features have the same number of dimensions as the user profile feature.