# Correlation-Aware Multi-Label Active Learning for Web Service Tag Recommendation

Weishi Shi, Xumin Liu, and Qi Yu
College of Computing and Information Science
Rochester Institute of Technology
Email:ws7586@g.rit.edu, xumin.liu@rit.edu, qi.yu@rit.edu

***Abstract*—Tag recommendation has gained significant popularity for annotating various web-based resources including web services. Compared with other approaches, tag recommendation based on supervised learning models usually lead to good accuracy. However, a high-quality training data set is needed, which demands manual tagging efforts from domain experts. While we could leverage the tags of existing web services assigned by their developers, the quality of these tags may not be good enough to build accurate classifiers for tag recommendation. In this paper, a novel multi-label active learning approach is proposed for web service tag recommendation. The proposed approach is able to identify a small number of most informative web services to be tagged by domain experts. We further minimize the domain expert efforts by learning and leveraging the correlations among tags to improve the active learning process. We conduct a comprehensive experimental study on a real-world data set and results demonstrate the effectiveness of our approach.**

***Index Terms*—Tag recommendation, Active learning, Multi-label classification.**

## I. Introduction

Service discovery has been a key and long lasting research topic in the area of service computing ever since the web service technologies were introduced [1]. Traditional service discovery solutions include registry-based and community-based. The former uses a centralized service registry, which can be standard (e.g. Universal Description, Discovery, and Integration (UDDI), which is an XML-based standard for describing, publishing, and finding web services) or non-standard (i.e., programmableweb query platform [1]). It provides a platform for service providers to publish their service descriptions as well as for service users to query the services they need. Community-based solutions allow service discovery to be conducted among homogeneous service groups, where services providing similar functionality belong to the same group. Those groups are formed by applying text mining techniques to learn service functionally from their descriptions.

The above efforts rely on predefined service categories made by domain experts, which exposes high demand on the experts and does not adapt well to the highly dynamic nature of a large-scale web service pool. As a result, tag-based methods have gained their popularity and complement the current service discovery solutions [2], [3]. The idea of those methods is to ask service developers (or providers) to tag their services using the terms that are representative for the service functionality. For example, GEOSPAN, a visual geographic information service, has the tags including aerial, geocoding, GIS, imagery, location, and mapping. Compared with predefined service categories, service tags are more flexible and could be more precise since a service providers usually has the best knowledge of his or her own services.

Content-based tag recommendation has been a useful tool to assist service developers to tag their services [4], [3]. The idea is to automatically derive relevant terms from service descriptions and recommend the top ranked terms as the tags to service providers. This can, especially in a large-scale service space, effectively reduce the efforts of service developers as well as improve the quality of tags. Typical tag recommendation methods learn relevant terms as the candidate tags by leveraging the current data mining solutions, including classification based [5], association rule based [6], and topic modeling based [7]). Classification-based methods usually outperform the others in term of recommendation accuracy since they follow a supervised learning process. That is, they start with a training data set, where the tags of services are already given, to learn the classification model, which can be used to predict the tags of new services. The downside of these methods is that the accuracy of the learning model highly depends on the quality of the training set.

While we could leverage the tags of existing web services assigned by their developers, these developer assigned tags are not always reliable and adequate to build accurate classifiers for tag recommendation. On one hand, ordinary API developers may have difficulty in choosing correct tags from a large candidate pool. Developers may tag their service incorrectly or miss important tags due to the lack of knowledge of all candidate tags. On the other hand, many developers may not want to spend a lot of time examining all candidate tags. Research shows that a well trained tag recommendation system can aid user to annotate tags more precisely and faster [4]. Recent studies suggest given the training data with high quality, supervised learning algorithm is more likely to outperform unsupervised and other methods for tag recommendation [8], [7]. However, to the best of our knowledge, there is no existing work that provides an efficient strategy to improve the quality of the training data. Instead, most works adopt tags annotated by users (or service developers in our case) as training data,

[1] http://www.programmeableweb.com

978-1-5386-0752-7/17 $31.00 © 2017 IEEE
DOI 10.1109/ICWS.2017.37

which can be both inaccurate and incomplete thus are not suitable for training purpose.

To address the challenges outlined above, we develop a novel multi-label active learning approach for web service tag recommendation. The idea of active learning (AL) is to start with base classifier learned from a very small sized training set [9]. It then iteratively selects the most informative services in the rest of the service pool, asks a domain expert to tag it, and evolves the classifier using the new knowledge. The learning stops when the classifier achieves a desired accuracy. Due to the careful selection of the services, the quality of the learned classifier can be significantly improved after each iteration and reaches convergence rapidly. As a result, the training process needs much less tagging effort from domain experts than the traditional classification method. In the proposed approach, AL will be used to train a multi-label classifier, which performs binary classification on a service for each possible label to determine if the tag should be recommended to the service. This allows to suggest different number of tags to a service, which is consistent with the real world situations, i.e., different services may have different number of tags.

We further augment the proposed multi-label active learning with a correlation-aware learning strategy. More specifically, suppose a domain expert is asked to read a service's description in order to decide if a tag should be assigned to the service, he or she may be asked about the service again and again later for different tags if the service is selected for those tags during the active learning process. For a large size service pool and large number of possible tags, the domain expert may have to read the same service description multiple times, which is an ineffective use of their effort. The correlation-aware active learning approach will efficiently learn the correlation relationship among different tags and leverage the correlation information to further reduce experts' tagging effort.

The remainder of the paper is organized as follows. Section 2 highlights the related work of tag recommendation, active learning, and their applications. Section 3 presents proposed correlation-aware multi-label active learning approach for tag recommendation. Section 4 shows the experimental results. Finally, Section 5 concludes the paper.

## II. Related Work

In this section, we review some representative efforts on tag recommendation and compare them to our approach. To better tackle the recommendation issues for web services, we mainly focus on content-based tag recommendation and do not consider user information.

*a) Recommendation Based on Current Tags:* This type of work assumes that a web resource is already tagged by its provider. However, such tags may have serious quality issues, such as they are not sufficient, representative or accurate. This will limit their usefulness when supporting the web resource annotation and query. To address this problem, [6] and [7] derive new tags to a resource based on its current tags assigned by its provider. [7] is an approach based on Latent Dirichlet Allocation (LDA) which is a popular model for topic modeling [10]. It models each web resource as a document consisting of its tags. LDA is used to learn the latent topics from those tags. When a new resource comes in, the system will determine the relevant topics from its current tags, and suggest the top ranked tags in those topics to annotate the resource. [6] is association rule based. It learns and uses the co-occurrence relationship among tags for the recommendation. For example, if it detects that there is enough evidence showing that the web resources using tags $T_1$ also use tags $T_2$, it will suggest $T_2$ to those new resources, which are already assigned tag $T_1$ by their providers.

*b) Recommendation Based on Resource Description:* The efforts under this category do not assume a new web resource to have any tag when it comes. They focus on learning the textual description or the content of the web resource and derive relevant tags from it. [11] proposes a LDA-based method. It extends the basic LDA graphic model by adding tags and their links to latent topics. The posterior probability distribution of topic on terms in documents and the posterior probability distribution of topic on tags for documents are learned and used to determine if a tag should be assigned to a new document. [12] leverages the idea of collaborative filtering into the tag recommendation for web blogs. Web blogs are treated as users and tags are treated as products. It follows the idea that similar webblogs should share the similarity of tags they use. Therefore, given a new web blogs, the tags of the top similar web blogs are retrieved as the candidates. The tags are ranked based on their frequencies of being used by those web blogs. [5] proposes a two-stage process for tag recommendation. The first stage performs a unsupervised learning process and cluster documents into groups based on their similarity. The tags of the documents in a cluster are ranked based on how much they represent those documents. During the second stage, a new document will be classified into one of the previously generated cluster using naive Bayes classifier. The top ranked tags in the corresponding cluster will then be suggested to annotate the document.

The major limitation of these existing work is that they still heavily rely on the tags assigned by users and can thus suffer from their low quality. This can be effectively addressed by the proposed active learning approach.

## III. Tag Recommendation

### A. Settings and Notations

We use a matrix $X \in \mathbb{R}^{m \times n}$ to denote the input data, where the $i$-th service $\mathbf{x}_i$ is represented by the $i$-th row of of $X$, and $X_{ij}$ denotes the term frequencyinverse document frequency (TF-IDF) score of the $j$-th term in $\mathbf{x}_i$'s description. Let $\mathcal{L} = \{1, ..., L\}$ be the set of indices of all possible tags and the binary matrix $Y \in \mathbb{B}^{m \times L}$ represents the tags associated with all services in $X$. Specifically, $Y_{ij} = 1$ if tag $t_j$ appeared in the $i$-th service and 0 otherwise. We use a symmetric matrix $C \in \mathbb{R}^{L \times L}$ to represent the correlation between tags. Specifically, $C_{ij}$ measures the correlation between tags $t_i$ and $t_j$ which is given by one of the correlation functions $Cor(t_i, t_j)$ defined

in the next subsection. $Cor(t_i, t_j)$ should obey the following two laws as a correlation function.

1) Commutative : $Cor(t_i, t_j) = Cor(t_j, t_i)$
2) Diagonal identity: $Cor(t_i, t_i) = \sup\{Cor(t_i, t_j)|j = 1, .., L\}$

### B. Tag Correlation Computation

We explore two different ways, *Jaccard* and *hierarchical clustering*, to compute the correlation between different tags. The correlation information will be exploited by the multi-label active learning algorithm to reduce the human tagging effort.

*1) Jaccard tag correlation: Confidence* is often used to measure the significant level of a given rule in association rule mining algorithms (e.g., Apriori algorithm). It is defined as the probability of observing the rule's consequent (e.g., tag $t_j$) under the condition that the instance (e.g., a service in our case) also includes the antecedent (e.g., tag $t_i$) [13]:

$$Conf(t_i \Rightarrow t_j) = \frac{P(t_i \wedge t_j)}{P(t_i)} = P(t_j|t_i) \tag{1}$$

The above confidence rule provides a means to measure the correlation between tags $t_i$ and $t_j$. However, two problems arise. First, it is a directed measurement, which gives two different values of $Conf(t_i \Rightarrow t_j)$ and $Conf(t_j \Rightarrow t_i)$. As a result, it violates the commutative property we defined for the correlation function. Second, confidence is sensitive to the frequency of consequent in the data and can produce misleading results if the support of the consequent is higher than the confidence [14]. Such heavy consequent effect of a single rule can be avoided by considering a rule on two directions. This motivates us to consider Jaccard coefficient [15] to measure correlation between two tags $t_i$ and $t_j$ .

$$\begin{aligned} JAC(t_i, t_j) &= \frac{P(t_i \wedge t_j)}{P(t_i) + P(t_j) - P(t_i \wedge t_j)} \\ &= \frac{1}{Conf(t_i \Rightarrow t_j)^{-1} + Conf(t_j \Rightarrow t_i)^{-1} - 1} \end{aligned} \tag{2}$$

Eq (2) reveals the relationship between Jaccard coefficient and the two-way confidence measurement. Two tags $t_i$ and $t_j$ will have high correlation if and only if the confidence level of both rule $t_i \Rightarrow t_j$ and $t_j \Rightarrow t_i$ are significant. The marginal probability in (2) is given by the support:

$$P(t_j) = Support(t_j) = \frac{|\{\mathbf{x}_i|Y_{ij} = 1, \mathbf{x}_i \in X\}|}{m} \tag{3}$$

Substituting (3) into (2), we recover the Jaccard coefficient as:

$$JAC(t_i, t_j) = \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \tag{4}$$

where $T_j = \{\mathbf{x}_i|Y_{ij} = 1, \mathbf{x}_i \in X\}$ is the set of services with tag $t_j$.

*2) Hierarchical clustering based tag correlation:* One issue of using Jaccard coefficient as the correlation function is that it can only discover the correlation between two tags that co-occur in some services. This may result in a correlation matrix that is very sparse especially when performing active learning as only a small number of services will be tagged. For instance, consider two tags $t_A$ and $t_B$ and let $T_A$ and $T_B$ to denote the sets of services that they are assigned to. Assume that $T_A = 1, 2, 3$ and $T_B = 4, 5, 6$. We have $JAC(t_A, t_B) = 0$ as they are not shared by any service. However, if $t_C$ with $T_C = 2, 3, 4, 5$ is observed, it is reasonable to believe that $t_A$ and $t_B$ may still be correlated as they both correlate with $t_C$. To accommodate this situation, we propose to employ a hierarchical clustering algorithm to capture the indirect correlation between tags, which help generate a dense correlation matrix of tags. Hierarchical clustering starts by treating each tag as an individual cluster. The distance between two clusters $A$ and $B$ is measured by their linkage. Specifically, in this paper we use Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [16] as a linkage function:

$$UPGMA(A, B) = \frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} Dist(x, y) \tag{5}$$

The algorithm repeatedly identifies one pair of clusters with minimum linkage and merges them as a new cluster until there is only one cluster left. When clusters $A$ and $B$ are merged, we use the linkage of them to update the correlation between all tags from A and B.

$$HC(t_i, t_j) = UPGMA(A, B)^{-1}, \forall t_i \in A, \forall t_j \in B \tag{6}$$

The correlation matrix generated using the hierarchical clustering approach will be more dense as the correlation score between two tags is computed using the reciprocal of linkage between their immediate parent clusters.

### C. Multi-label Active Learning for Tag Recommendation

We start by introducing the basic rational of a classical pool-based active learning model. We then present how to extend this basic model to multi-label settings for tag recommendation. An active learning model is initialized by using a few human labeled data samples to training a base classifier. It then applies the base classifier to a pool of unlabeled data samples and chooses one or a small batch of data samples from the pool. A human user will be queried to label the selected samples. The trained classifier will then be updated using the newly added labeled data samples and be applied again to the unlabeled data sample pool. This process continues until it reaches a predefined classification accuracy over a validation set. The effectiveness of active learning comes from its ability to choose the most informative data samples from the pool. One commonly used criterion is to choose the ones that the current classifier is the most uncertain to make a decision. By labeling these data samples, the classifier is expected to be improved the most in the next iteration. Support Vector Machines (SVMs) have been used in active learning as they provide a convenient way to choose uncertain data samples

from the unlabeled data pool [9]. For a typical single-label binary-class problem, a straightforward way is to choose the sample closest to the current decision boundary.

A SVM that performs binary classification does not serve the purpose of tag recommendation for two reasons. First, there will be a large number of potential tags, i.e., $L \gg 2$. Second, each service is typically assigned more than one tags. Thus, we need to develop an active learning model that works with a multi-class and multi-label setting. We propose to employ a binary relevant muli-label classifier that is comprised of $L$ binary classifiers $h_1$,...,$h_L$ to address the need of tag recommendation. In particular, each classifier $h_i$ is trained using the input matrix $X$ as the predictor and the $i$-th column of $Y$ as response, independently. The task of $h_i$ is to predict the relevance between all services and tag $t_i$, which will be performed by a SVM.

During the multi-label active learning process, each binary classifier $h_j$ takes turn to sample the most informative data $\mathbf{x}$ and form a query: Does tag $t_j$ belong to service $\mathbf{x}$? The query will be answered by an expert. After that $\mathbf{x}$ along with its label will be added to the training set of $h_j$. Notice that all queries are order independent and thus the algorithm can be easily executed in a distributed fashion.

Take one classifier $h_k$ as an example, the process of sampling in active learning can be described as follow:

**Algorithm 1** Active Learning of a Binary Classifier $h_k$

**Require:** input matrix $X$
**Ensure:** a trained $h_k$

1: Create a copy $X^{(k)}$ of $X$ and divide $X^{(k)}$ into a smaller part $X^{(k)}_{train}$ and a larger part $X^{(k)}_{candidate}$.
2: Let expert label the relevance between $X^{(k)}_{train}$ and tag $t_k$ and store the result into vector $\mathbf{y}^{(k)}$.
3: **while** stop condition != TRUE **do**
4: Train a binary SVM on $\left\{(X^{(k)}_{train}, \mathbf{y}^{(k)})\right\}$.
5: Sample a service $\mathbf{x}$ from $X^{(k)}_{candidate}$ using

$$\mathbf{x} = \underset{\mathbf{x}\in X^{(k)}_{candidate}}{\arg\min} \; |\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b| \qquad (7)$$

where $\alpha$ and $b$ are parameters of the SVM trained from $X^{(k)}_{train}$, $N$ is number of services in $X^{(k)}_{train}$, and $K(\cdot,\cdot)$ is the kernel function.
6: Let the expert label the relevance between $\mathbf{x}$ and $t_k$ then add $\mathbf{x}$ and its label to $X^{(k)}_{train}$ and $\mathbf{y}^{(k)}$, respectively.
7: **end while**

### D. Correlation-aware Active Learning

In this section we propose an augmentation of the multi-label active learning model by learning/leveraging tag correlations to further reduce human experts' tagging effort.

In general, the labor cost of an expert in active learning consists of two parts: comprehension time (CT) and decision time (DT). CT refers to the time that an expert needs to spend on reading, analysing, and understanding the content of the service description. We assume that CT increases with the length of a service description. That is, the longer a service description is, the more time an expert needs to read and digest it. DT refers to the time an expert needs to spend on annotating all candidate tags. Once an expert reads a service description and fully understands its functionality, to add a few additional tags to be considered by an expert does not increase DT significantly. The fact that $CT$ is usually much longer than $DT$ motivates us to merge multiple queries into one in order to let experts annotate multiple tags at the same time.

Suppose that classifier $h_i$ samples a service $\mathbf{x}$ and predicts to assign tag $t_i$ to the service. Before going to the next classifier $h_{i+1}$ to sample another service, we check whether there are any other tags that are highly relevant to $t_i$ so that we can have the expert to annotate them simultaneously. We propose to identify the tags most relevant to $t_i$ by computing the joint probability $P(x, t_i, t_j), \forall j \neq i$. Choosing a most relevant tag to be annotated along with $t_i$ can be achieved as by maximizing $P(x, t_i, t_j)$:

$$\begin{aligned} t_j &= \arg\max_{j\in\mathcal{L}, j\neq i} P(x, t_i, t_j) \\ &= \arg\max_{j\in\mathcal{L}, j\neq i} P(x)P(t_i|x)P(t_j|t_i) \\ &= \arg\max_{j\in\mathcal{L}, j\neq i} P(x)\frac{P(t_i, x)}{P(x)}\frac{P(t_i, t_j)}{P(t_i)} \\ &= \arg\max_{j\in\mathcal{L}, j\neq i} P(x|t_i)P(t_i, t_j) \\ &= \arg\max_{j\in\mathcal{L}, j\neq i} P(t_i, t_j) \end{aligned} \qquad (8)$$

The joint probability $P(t_i, t_j)$ can be approximated using tag correlation, i.e., $P(t_i, t_j) \approx C_{i,j}$. The process of leveraging tag correlation to further optimize querying of a human expert is illustrated in Figure 1.

## IV. Experiment

We present a comprehensive empirical study to demonstrate the proposed correlation-aware multi-label active learning model for service tag recommendation.

### A. Dataset Description and Experiment Setup

We collected 2000 real web services from www.programmableweb.com. For each service, we selected three attributes: Title, Summary, and Description, and concatenated them to form content of a service. To ensure sufficient content for services in both the training and testing sets, we remove services whose content is less than 50 terms. Based on the frequency of usage, we divided all 504 tags that have been used in those services into three categories: hot tags (frequency $\geq$ 20), medium tags (10 $\leq$ frequency $<$ 20) and rare tags (frequency $<$ 10). Figure 2 shows the distribution of the top 100 frequent tags.

To ensure sufficient statistical strength, we chose the 20 most frequent tags as candidate tags for recommendation. These tags are *search, social, tools, science, financial, sms,*

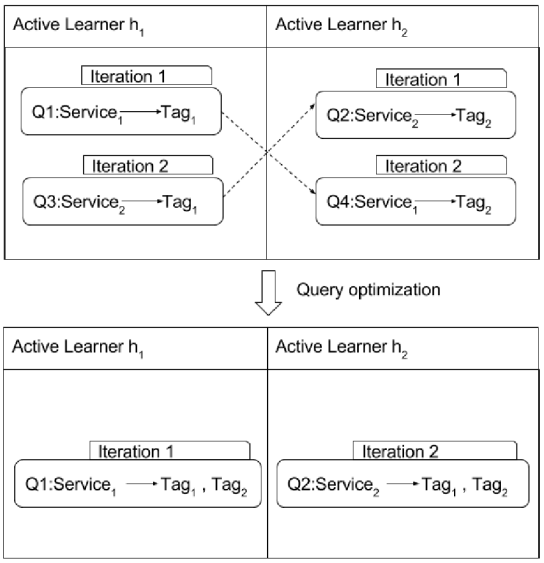


Fig. 1. Correlation-aware Active Learning

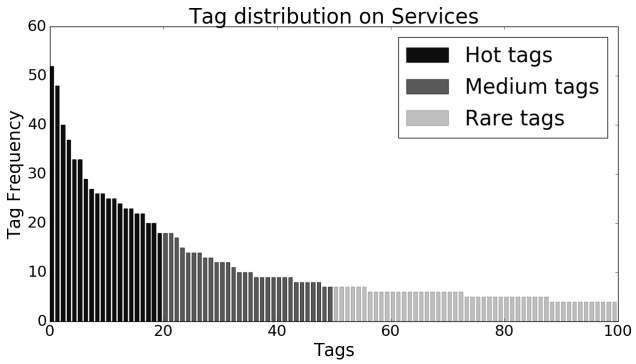


Fig. 2. Distribution of the Top 100 Tags

*internet, mapping, payment, mobile, enterprise, reference, shopping, telephony, photo, government, marketing, messaging, utility*, and *video*. We randomly selected 600 services for training and the rest 1400 for testing. The training and testing services have been converted into a bag-of-word matrix using TF-IDF scoring, where each row of the matrix represent an individual service while each column represents a term from the dictionary of the corpus. Terms will be filtered out if they occur either in too many services (with document frequency $\geq 0.75$) or in too few services (i.e., less than 5).

TABLE I
SUMMARY OF THE DATASET

| Data Set | m | n | L | LCARD | PUNIQ | PMAX |
|---|---|---|---|---|---|---|
| Train | 600 | 812 | 20 | 2.96 | 0.0033 | 0.033 |
| Test | 1400 | 812 | 20 | 3.20 | 0.0015 | 0.026 |

We use Label Cardinality (LCARD) [17] to measure the *multilabledness* of the dataset. It gives the average number of tags attached to each service. Given a label matrix $Y \in \mathbb{B}^{m \times L}$, LCARD is defined as:

$$LCARD = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{L} Y_{i,j} \tag{9}$$

To further discover the correlation between tags, we introduce two more measurements: Proportion of Unique Label Combinations (PUNIQ) and Proportion of label sets with the Maximum frequency (PMAX) [18]:

$$PUNIQ = \frac{\left|\left\{i | \sum_{j=1, j\neq i}^{L} (Y^T Y)_{i,j} = 0, i \in \mathcal{L}\right\}\right|}{m} \tag{10}$$

$$PMAX = \frac{\max\{(Y^T Y)_{ij} | i, j \in \mathcal{L}\}}{m} \tag{11}$$

PUNIQ is an indicator of the population of absolute isolated tags in the data set. A high PUNIQ implies tags are correlated loosely. On the other hand, PMAX reflects the upper bound of the correlation strength of tags in the data set. The correlation aware active learning will be more effective if $PMAX > PUNIQ$. Table I summarizes the major characteristics of our data and Table II lists the notations used in the later part of the analysis.

TABLE II
SUMMARY OF NOTATIONS

| Notation | Description |
|---|---|
| BRAL | Binary relavent based active learning |
| HCL/HCH | Hierarchical clustering based correlation aware approach with low/high threshold(0.1/0.65) |
| JACL/JACH | Jaccard coefficient based correlation aware approach with low/high threshold(0.1/0.65) |

Due to the sparsity of the tag space, the classification accuracy is no longer a proper measurement for multi-tag recommendation. A recommendation system can achieve a high accuracy by simply not recommending any tag. We instead use the F-measure, which is the harmonic mean between precision and recall to evaluate model performance:

$$\begin{aligned} \text{F-measure} &= \frac{1}{L}\sum_{i}^{L} F(\hat{Y}_i, Y_i) \\ &= \frac{1}{L}\sum_{i}^{L} \frac{2P(\hat{Y}_i, Y_i)R(\hat{Y}_i, Y_i)}{P(\hat{Y}_i, Y_i) + R(\hat{Y}_i, Y_i)} \end{aligned} \tag{12}$$

where $\hat{Y}$ is the predict result while $\hat{Y}_i$ represents the recommended result for tag $t_i$. The precision function $P()$ and recall function $R()$ are given by:

$$P(\hat{Y}_i, Y_i) = \frac{\sum(\hat{Y}_i \wedge Y_i)}{\sum \hat{Y}_i}, \quad R(\hat{Y}_i, Y_i) = \frac{\sum(\hat{Y}_i \wedge Y_i)}{\sum Y_i} \tag{13}$$

It has already been shown that a supervised learning model trained using high-quality training data can achieve very accurate tag recommendation [8], [7]. Therefore, instead of focusing on comparing the F-measure with other existing tag recommendation approaches, we mainly aim to evaluate the effectiveness of the proposed active learning models. The goal is to show that they can provide an efficient and cost-effective means to achieve a high-quality training set that can be used to build a supervised model for accurate tag recommendation.

### B. Correlation Among Tags

We visualize correlation matrices achieved by $JAC$ and $HC$ in Figure 3. As can be seen, $HC$ is less sensitive to data sparsity and leads to a more dense matrix than $JAC$. By analyzing the discovered tag correlations, we identify three interesting types of correlations based on the intrinsic relationships of tags: complementary, synonyms, and hierarchical (see Table III for example tags in each type).

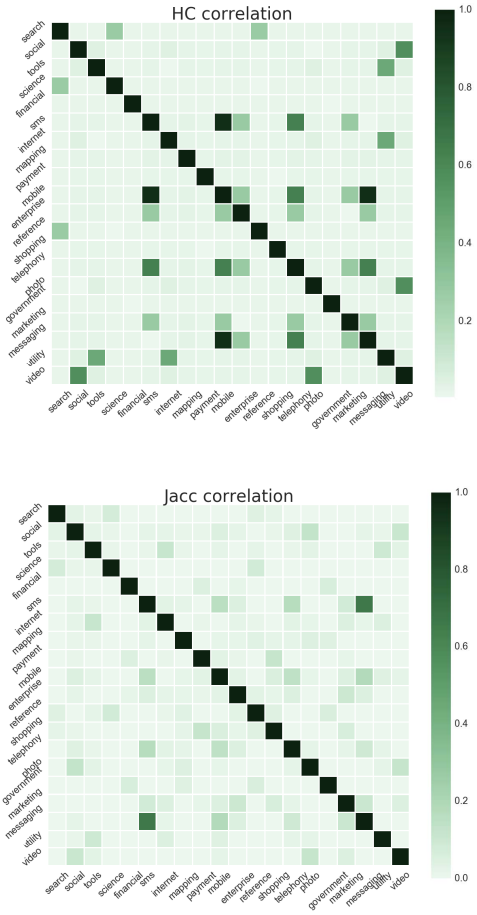


Fig. 3. Tag correlations

TABLE III
CORRELATION TYPES

| Correlation type | Example tag pairs |
|---|---|
| Complementary | (video,photo),(messaging,photo) |
| Synonym | (tools,utility),(search,reference) |
| Hierarchical | (mobile,telephony),(science,search),(messaging,sms) |

### C. Effectiveness of Multi-label Active Learning

In this set of experiments, we report the result of proposed multi-label active Learning appraoch, referred to as Binary Relevant based Active Learning (BRAL), for tag recommendation. We demonstrate its effectiveness by comparing it with a random sampling based approach. Both approaches start with 120 services along with their tags as the initial training pool. The rest 480 services and their tags serve as the candidate training pool, which consists of $480 \times 20 = 9600$ unlabeled service-tag pairs. At each iteration, BRAL forms a query by choosing one service-tag pair from the candidate training pool through uncertainty sampling while random sampling forms the query by randomly choosing one service-tag pair from the pool. After the query is answered, the service-tag pair is added to the training pool and F-measure of the model on the test data is recorded. We also present the best performance of the model, which is achived when all service-tag pairs in the candidate pool are annotated. From Figure 4, we observe a significantly faster convergence rate of BRAL as compared to random sampling. Therefore, we only need to rely on human experts to tag a small subset of informative services to achieve a satisfactory performance. In contrast, to achieve an equally good performance, a much larger number of services need to be labeled if active learning is not used.

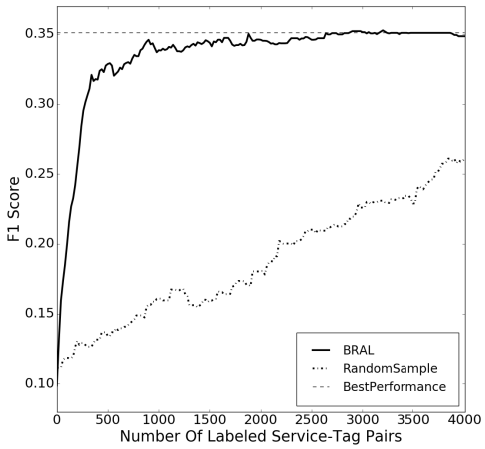


Fig. 4. Binary Relevant based Active Learning Vs. Random Sampling

### D. Correlation-aware Active Learning

We consider two strategies to generate the tag correlation matrix: statically computing tag correlations before the active learning process or dynamically updating the correlation matrix along with active learning. The static strategy is suitable when some prior knowledge of tags can be obtained from some other resources. The dynamic strategy, on the other hand, assumes no prior knowledge on tags, making it more widely applicable. However, as tag correlations are continuously calculated, the correlation information at the early phase may not be very accurate. It is worth to note that our proposed active learning process allows us to accurately compute the tag correlations by tagging a very small number of services.

To test the effectiveness of correlation-aware active learning, we experiment with two thresholds, 0.1 and 0.65 on the joint probability (see section III-D). A lower threshold allows the model to merge more queries for labor saving. However, it might harm the model performance as the merged queries may violate the uncertainty sampling rule of active learning. Figure 5 summarizes the results. In general, most correlation aware approaches can achieve a similar F-measure by labeling

much less number of service-tag pairs. We will provide a detailed analysis on labor efficiency in the next section. Meanwhile, while choosing a low threshold may allow us to merge more queries, it may hurt the accuracy of the model. For example, when the correlation matrix is statically computed using hierarchical clustering (HCL in the top figure), the F-measure is obviously lower than BRAL. We will provide a detailed analysis of this behavior in the next section.

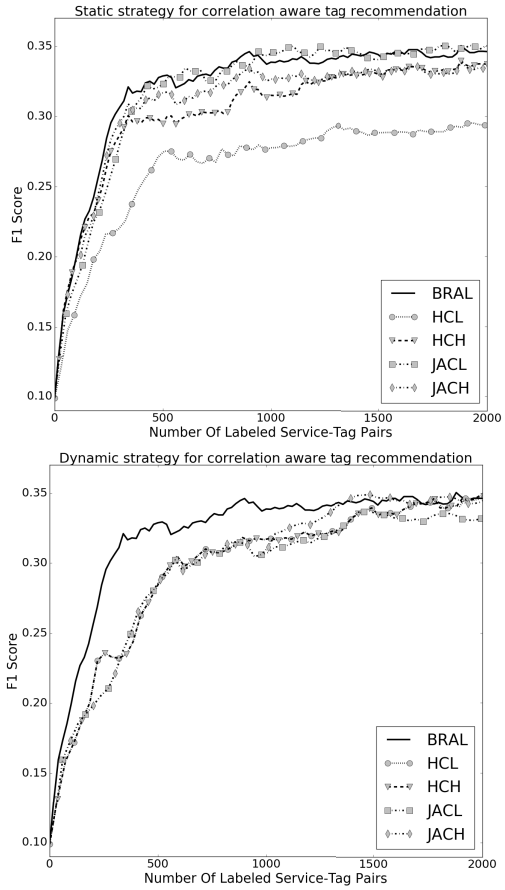


Fig. 5. Effectiveness of leveraging tag correlations

### E. Analysis of Label Efficiency

The goal of correlation aware query optimization is to save labor cost in active learning by merging two service-tag pairs into one. In order to evaluate the effectiveness of labor saving, we label $N = 4000$ service-tag pairs in total and record the number of merged queries as $MQ$. We assume the comprehension time for expert to understand the description of all services are identical to a constant which is proportional to the average length of all services. Then a merged query can be seen as labeling two service-tag pairs at labor cost of on standard query. As a result, we define the total labor cost (TLC) for an active learning trail as:

$$TLC = (N - MQ) + \frac{MQ}{2} = N - 0.5MQ \qquad (14)$$

We use $TLC@F_K$ which is the TLC of the model when its performance first achieves F-measure=K to evaluate the labor efficiency of the active learning model. Figure 6 shows that when tag correlation is dynamically updated given the current tagging results, all four correlation aware approaches outperform BRAL with less labor cost to achieve the same prediction performance. When tag correlation is statistically computed (referred as the global correlation matrix), both HCL and HCH cost more labor to converge than BRAL. The reason is that hierarchical clustering tends to generates a dense matrix to encourage aggressive exploration among tags. As a result, the correlation matrix may contain more noises than the matrix from Jaccard coefficient. When considering all the data, the noises are introduced at the beginning of the active learning and will have an accumulated effect for the entire trail. The relationship between correlation matrices from dynamic and static strategies is presented in Figure 7, where the distance between global correlation matrix $A$ and dynamic correlation matrix $B$ is defined as the Frobenious norm of their (relative) difference: $Distance(A, B) = \frac{\|A-B\|_F}{\|A\|_F}$. As can be seen, dynamic correlation matrices from both JACH and HCH quickly approach their corresponding global matrices. However, the correlation matrix from HCH at the beginning of the active learning is twice far away from its global matrix as compared to JACH.

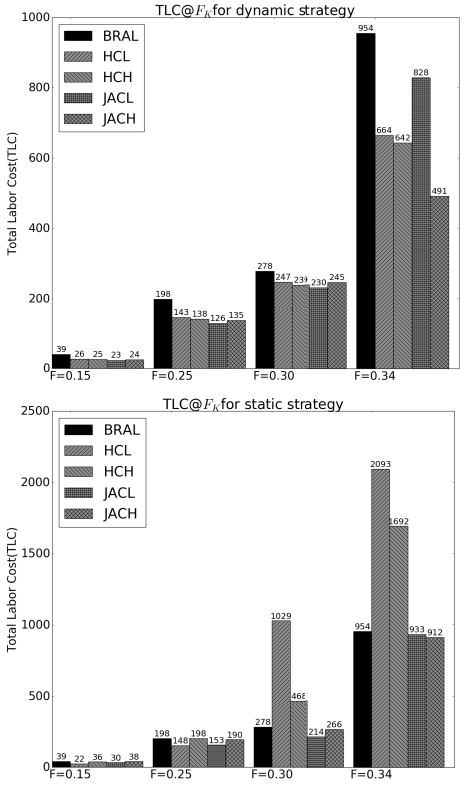


Fig. 6. TLC @$F_K$

### F. Examples of Recommended Tags

Note that our model is evaluated using tags provided by individual service developers. As discussed earlier, these tags may be incomplete. We inspect our tag recommendation result

TABLE IV
EXAMPLES OF RECOMMENDED TAGS

| Service description | Original tags | Newly predicted tags |
|---|---|---|
| ZNISMS is one of the SMS providers in India, providing a number of services for sending standard and bulk SMS messages... | sms, telephony | messaging |
| ...Matrix SMS Gateway for group messaging provides businesses a hosted messaging platform to add SMS capability to any system,...querying message status and cellular carrier lookup... | sms, telephony, mobile | messaging, search |
| ODP is a Chinese site that provides search and aggregation of top journals and articles for a variety of subjects... | China, news | search |
| Use Jott Links to blog from your cell phone wherever you are. Use Jott Links for social networking and micro-blogging... | messaging | social, mobile |
| Shift PlatformMarketing Tools PlatformShift is an enterprise collaboration platform for social media marketers... | advertising, collaboration, enterprise, marketing, project-management | social, tools |

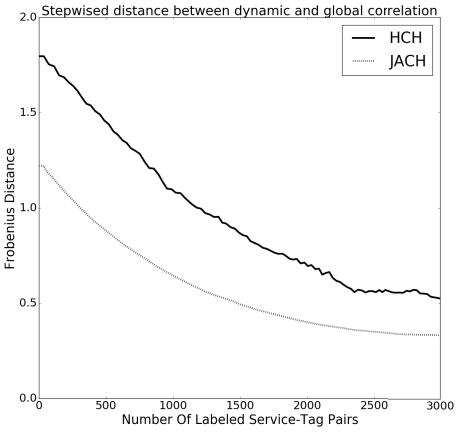


Fig. 7. Convergence of dynamic correlation computation

and find that our tag recommendation system is able to recommend those missing tags. To provide some evidence on this, we list some of examples in Table IV. Note that those recommended tags are not considered as positive predictions when we evaluate the performance of the model, which implies the performance of the proposed model can work even better in practice than the performance we reported previously.

## V. CONCLUSION

We present a novel mutli-label active learning approach for web service tag recommendation. Through choosing the most informative service-tag pairs, the proposed approach leads to an efficient and cost-effective way for building a high-quality training set. Such a dataset is critical to train highly accurate multi-label classification models for service tag recommendation. By leveraging the correlations among different tags, we further reduce the labor cost in the active learning process while maintaining a high recommendation performance. A comprehensive experimental study over real-world web service data demonstrates the effectiveness of the proposed approach.

## REFERENCES


[1] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed, "Deploying and managing web services: issues, solutions, and directions," *The VLDB Journal*, vol. 17, pp. 537–572, May 2008. [Online]. Available: http://dx.doi.org/10.1007/s00778-006-0020-3

[2] B. Sigurbjörnsson and R. Van Zwol, "Flickr tag recommendation based on collective knowledge," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 327–336.

[3] T. Liang, L. Chen, J. Wu, and A. Bouguettaya, "Exploiting heterogeneous information for tag recommendation in API management," in *IEEE International Conference on Web Services, ICWS 2016, San Francisco, CA, USA, June 27 - July 2, 2016*, 2016, pp. 436–443. [Online]. Available: http://dx.doi.org/10.1109/ICWS.2016.63

[4] M. Ames and M. Naaman, "Why we tag: motivations for annotation in mobile and online media," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 971–980.

[5] Y. Song, L. Zhang, and C. L. Giles, "Automatic tag recommendation algorithms for social recommender systems," *ACM Trans. Web*, vol. 5, no. 1, pp. 4:1–4:31, Feb. 2011. [Online]. Available: http://doi.acm.org/10.1145/1921591.1921595

[6] P. Heymann, D. Ramage, and H. Garcia-Molina, "Social tag prediction," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 531–538. [Online]. Available: http://doi.acm.org/10.1145/1390334.1390425

[7] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 61–68.

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

[9] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[11] X. Si and M. Sun, "Tag-lda for scalable real-time tag recommendation," *Journal of Computational Information Systems*, vol. 6, no. 1, pp. 23–31, 2009.

[12] G. Mishne, "Autotag: a collaborative approach to automated tag assignment for weblog posts," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 953–954.

[13] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Acm sigmod record*, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[14] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," in *Acm Sigmod Record*, vol. 26, no. 2. ACM, 1997, pp. 265–276.

[15] P.-N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," *Information Systems*, vol. 29, no. 4, pp. 293–313, 2004.

[16] R. R. Sokal, "A statistical method for evaluating systematic relationships," *Univ Kans Sci Bull*, vol. 38, pp. 1409–1438, 1958.

[17] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.

[18] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.