Automatic Tag Recommendation for Metadata Annotation Using Probabilistic Topic Modeling

Suppawong Tuarob
Computer Science and Engineering
Pennsylvania State University
University Park, Pennsylvania
suppawong@psu.edu

Line C. Pouchard
Scientific Data Group
Computer Sciences and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee
pouchardlc@ornl.gov

C. Lee Giles
Information Sciences and Technology
Computer Science and Engineering
Pennsylvania State University
University Park, Pennsylvania
giles@ist.psu.edu

ABSTRACT

The increase of the complexity and advancement in ecological and environmental sciences encourages scientists across the world to collect data from multiple places, times, and thematic scales to verify their hypotheses. Accumulated over time, such data not only increases in amount, but also in the diversity of the data sources spread around the world. This poses a huge challenge for scientists who have to manually search for information. To alleviate such problems, ONEMercury has recently been implemented as part of the DataONE project to serve as a portal for accessing environmental and observational data across the globe. ONEMercury harvests metadata from the data hosted by multiple repositories and makes it searchable. However, harvested metadata records sometimes are poorly annotated or lacking meaningful keywords, which could affect effective retrieval. Here, we develop algorithms for automatic annotation of metadata. We transform the problem into a tag recommendation problem with a controlled tag library, and propose two variants of an algorithm for recommending tags. Our experiments on four datasets of environmental science metadata records not only show great promises on the performance of our method, but also shed light on the different natures of the datasets.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

Keywords

Document Annotation, Tag Recommendation, Topic Model

1. INTRODUCTION

Environmental sciences have become both complex and data-intensive, needing access to heterogenous data collected from multiple places, times, and thematic scales. For example, research on bird migration would involve exploring and analyzing observational data such as the migration of animals and temperature shifts across the world, from time to time. While the needs to access such heterogenous data are apparent, the rapid expansion of observational data, in both quantity and heterogeneity, poses huge challenges for data seekers to obtain the right information for their research. Such problems behoove tools that automatically manage, discover, and link big data from diverse sources, and present the data in the forms that are easily accessible and comprehensible.

1.1 ONEMercury Search Service

Recently, DataONE, a federated data network built to facilitate accesses and preservation about environmental and ecological science data across the world, has come to exist and gain increasingly popularity[16]. DataONE harvests metadata from different environmental data providers and make it searchable via the search interface ONEMercury[1], built on Mercury[2], a distributed metadata management system. Figure 1 shows sample screen shots of the ONEMercury search interface (left) and the search result page with search query '`soil`'. ONEMercury offers a full text search on the metadata records. The user can also specify the boundary of locations in which the desired data is collected or published using the interactive graphic map. At the result page, the user can choose to further filter out the results by `Member Node`, `Author`, `Project`, and `Keywords`. The set of keywords used in the system are static-users cannot arbitrarily add new or remove the existing keywords- and managed by the administrator to avoid the emerging of spuriously new keywords. Such keywords are used for manually annotating metadata during the data curation process.

1.2 Challenge and Proposed Solution

Linking data from heterogenous sources always has a cost. One of the biggest problems that ONEMercury is facing is the different levels of annotation in the harvested metadata records. Poorly annotated metadata records tend to be missed during the search process as they lack meaningful keywords. Furthermore, such records would not be compatible with the advanced mode offered by ONEMercury as it

[1]https://cn.dataone.org/onemercury/
[2]http://mercury.ornl.gov/

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.
JCDLâĂŹ13, July 22–26, 2013, Indianapolis, Indiana, USA.
Copyright 20XX ACM 978-1-4503-2077-1/13/07 ...$15.00.

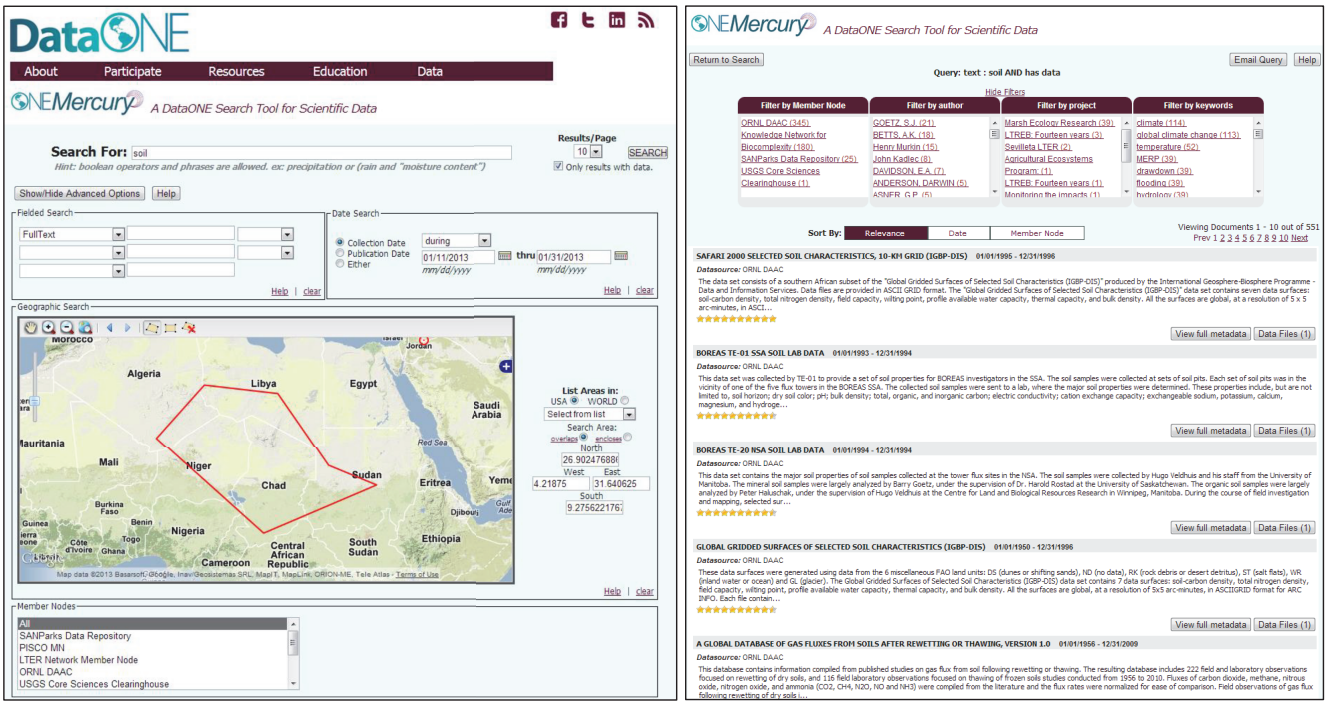

Figure 1: Screen shots of the ONEMercury search interface and result page using query '`soil`'.

requires the metadata records be annotated with keywords from the keyword library. The explosion of the amount of metadata records harvested from an increasing number of data repositories makes it even impossible to annotate them manually by hand, urging the need for a tool capable of automatically annotating these metadata records which are poorly annotated.

In this paper, we address the problem of automatic annotation of metadata records. Our goal is to build a fast and robust system that annotates a given metadata record with related keywords from a given keyword library. The idea is to annotate a poorly annotated record with keywords associated to the well annotated records that it is most similar with. We propose a solution to this problem by first transforming the problem into a tag recommendation problem with a controlled tag library, where the set of recommended tags is used to annotate the given metadata record, and then propose an algorithm that deals with the problem.

1.3 Problem Definition

We define a document as a tuple of textual content and a set of tags. That is `d = <c, e>`, where `c` is the textual content, represented by a sequence of terms, of the document `d` and `e` is a set of tags associated with the document. Given a tag library T, a set of annotated documents D, and a non-annotated query document q, our task is to recommend a ranked set of K tags taken from T to the query q. A document is said to be annotated if it has at least one tag; otherwise, it is non-annotated. The formal description of each variable is given below:

$$T = \{t_1, t_2, ..., t_M\}; t_i \text{ is a tag.}$$

$$D = \{d_1, d_2, ..., d_N\}; d_i = \langle c_{di}, e_{di} \rangle, e_{di} \subseteq T, \text{and } e_{di} \neq \oslash$$

$$q = \langle c_q, \oslash \rangle$$

1.4 Contributions

This paper has four key contributions as follows:

1. We address a real word problem of linking data from multiple archives faced by ONEMercury. We transform the problem into the tag recommendation problem, and generalize the problem so that the proposed solution can further be applied to other domains.

2. We propose a novel technique for tag recommendation. Given a document query `q`, we first compute the distribution of tags. The top tags are then recommended. We propose two variants of our algorithms: term frequency-inverse document frequency (TF-IDF) based and topic model (TM) based.

3. We crawl environmental science metadata records from 4 different archives for our datasets: the Oak Ridge National Laboratory Distributed Active Archive Center (DAAC)[3], Dryad Digital Repository[4], the Knowledge Network for Biocomplexity (KNB)[5], and TreeBASE: a repository of phylogenetic information[6]. We select roughly 1,000 records from each archive for the experiments.

4. We validate our proposed method using aggressive empirical evaluations. We use document wise 10 fold cross validation to evaluate our methods with 5 evaluation metrics: Precision, Recall, F1, MRR (Mean Reciprocal Rank), and BPref (Binary Preference). These evaluation metrics are extensively used together to evaluate recommendation systems.

[3]http://daac.ornl.gov/
[4]http://datadryad.org/
[5]http://knb.ecoinformatics.org/index.jsp
[6]http://treebase.org/treebase-web/home.html

5. We make our datasets and source code available upon request for research purposes.

2. RELATED WORKS

Literature on document annotation is extensive. Hence we only present the work closely related to ours.

2.1 Automatic Document Annotation

Newman et al.[18] discuss approaches for enriching metadata records using probabilistic topic modeling. Their approach treats each metadata record as a bag of words, and consists of 2 main steps: i) generate topics based on a given corpus of metadata, ii) assign relevant topics to each metadata record. Hence a metadata record is annotated by the top terms representing the assigned topics. They propose 3 variations of their approaches. The first method, which they use as the baseline, uses full vocabulary (every word) from the corpus. The remaining two methods filter out the vocabulary by deleting useless words resulting in more meaningful topics. They compare the three approaches in 3 aspects: % of usable topics, % enhanced records, and average coverage by the top 4 chosen topics. They acquired the datasets from 700 repositories, hosted by OAISter Digital Library. The results show that, overall, the second method performs the best. However, such method requires manual modification of the vocabulary, hence would not scale well. The third method performs somewhere in between.

Bron et al.[4] address the problem of document annotation by linking a poorly annotated document to well annotated documents using TF-IDF cosine similarity. One corpus consists of textually rich documents (A_s) while the other contains sparse documents (A_t). In the paper, they address two research problems: document expansion and term selection. For the document expansion task, each targeted document (a document in sparse set) is mapped to one or more documents in the rich set, using simple cosine-similarity measure. Top N documents are chosen from the rich corpus, and the texts in these documents are added to the targeted documents as supplemental content. The term selection task was introduced because using the whole documents from the source corpus to enrich the targeted document might be too spurious and have a fair chance of topic drifts. This term selection task aims to select only meaningful words from each document in the source corpus to add to the targeted documents. Basically, top K% of the words in each document, ranked by TF-IDF scores, are selected as representative words of the document.

This work has a similar problem setting to ours, except that we aim to annotate a query document with keywords taken from the library, while their approaches extract keywords from the full content of documents.

Witten et al. propose *KEA*, a machine learning based key phrase extraction algorithm from documents [23]. The algorithm can also be applied to annotate documents with relevant keyphrases. Their algorithm first selects candidate keyphrases from the document. Two features are extracted from each candidate keyphrase: TF-IDF score and distance of the first occurrence of the keyphrase from the beginning of the document. A binary NaiveBayes classifier is trained with the extracted features to build a classification model, which is used for identifying important keyphrases. The algorithm is later enhanced by Medelyan et al. [15] to improve the performance and add more functionality such as document annotation and keyphrase recommendation from control vocabulary, where the list of keyphrases to be recommend are already defined in the vocabulary. We use keyphrase recommendation with control vocabulary feature of the *KEA* algorithm as our baseline.

2.2 Automatic Tag Recommendation

Since we have a tag recommendation problem, we briefly cover related literature. Tag recommendation has gained substantial amount of interest in recent years. Most work, however, focuses on personalized tag recommendation, suggesting tags to a user's object based on the user's preference and social connection. Mishne et al. [17] employ the social connection of the users to recommend tags for weblogs, based on similar weblogs tagged by the same users. Wu et al.[24] utilize the social network and the similarity between the contents of objects to learn a model for recommending tags. Their system aims towards recommending tags for Flickr photo objects. While such personalized schemes have been proven to be useful, some domains of data have limited information about authors (users) and their social connections. Liu et al. [12] propose a tag recommendation model using Machine Translation. Their algorithm basically trains the translation model to translate the textual description of a document in the training set into its tags. Krestel et al.[11] employ topic modeling for recommending tags. They use the Latent Dirichlet Allocation algorithm to mine topics in the training corpus, using tags to represent the textual content. They evaluate their method against the association rule based method proposed in [7].

3. DATASETS

We obtain 4 different datasets of environmental metadata records for the experiments: the Oak Ridge National Laboratory Distributed Active Archive Center (DAAC)[7], Dryad Digital Repository (DRYAD)[8], the Knowledge Network for Biocomplexity (KNB)[9], and TreeBASE: a repository of phylogenetic information (TreeBASE)[10]. The statistics of the datasets including the number of documents, total number of tags, average number of tags per document, number of unique tags (tag library size), tag utilization, number of all words (dataset size), and average number of word per document, are summarized in Table 1. Tag utilization is the average number of documents where a tag appears in, and is defined as $\frac{\#\ all\ tags}{\#\ unique\ tags}$. The tag utilization measure quantifies how often, on average, a tag is used for annotation.

The Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) is one of the NASA Earth Observing System Data and Information System (EOSDIS) data centers managed by the Earth Science Data and Information System (ESDIS)[11] Project, which is responsible for providing scientific and other users access to data from NASA's Earth Science Missions. The biogeochemical and ecological data provided by ORNL DAAC can be categorized into four groups: Field Campaigns, Land Validation, Regional and Global Data, and Model Archive. After raw data is collected, the data collector describes the data and

[7] http://daac.ornl.gov/
[8] http://datadryad.org/
[9] http://knb.ecoinformatics.org/index.jsp
[10] http://treebase.org/treebase-web/home.html
[11] http://earthdata.nasa.gov/esdis

	# Docs	#All Tags	Avg Tags/Doc	#Uniq. Tags	Tag Util.	#All Words	Avg Words/Doc
DAAC	978	7,294	7.46	611	11.937	101968	104.261
DRYAD	1,729	8,266	4.78	3,122	2.647	224,643	129.926
KNB	24,249	254,525	10.49	7,375	34.511	1535560	63.324
TreeBASE	2635	1838	0.697	1321	1.391	30054	11.405

Table 1: Statistics of the 4 datasets.

annotates it using topic-represented keywords from the topic library.

Dryad is a nonprofit organization and an international repository of data underlying scientific and medical publications. The scientific, educational, and charitable mission of Dryad is to promote the availability of data underlying findings in the scientific literature for research and educational reuse. As of January 24, 2013, Dryad hosts 2570 data packages and 7012 data files, associated with articles in 186 journals. Metadata associated with each data package is annotated by the author with arbitrary choices of keywords.

The Knowledge Network for Biocomplexity (KNB) is a national network intended to facilitate ecological and environmental research on biocomplexity. For scientists, the KNB is an efficient way to discover, access, interpret, integrate and analyze complex ecological data from a highly-distributed set of field stations, laboratories, research sites, and individual researchers. Each data package hosted by KNB is described and annotate with keyword from the taxonomy by the data collector.

TreeBASE is a repository of phylogenetic information, specifically user-submitted phylogenetic trees and the data used to generate them. TreeBASE accepts all types of phylogenetic data (e.g., trees of species, trees of populations, trees of genes) representing all biotic taxa. Data in TreeBASE are exposed to the public if they are used in a publication that is in press or published in a peer-reviewed scientific journal, book, conference proceedings, or thesis. Data used in publications that are in preparation or in review can be submitted to TreeBASE but are only available to the authors, publication editors, or reviewers using a special access code. TreeBASE is produced and governed by the The Phyloinformatics Research Foundation, Inc[12].

In our setting, we assume that the documents are independently annotated, so that the tags in our training sets represent the gold-standard. However, some metadata records may not be independent since they may be originated from the same projects or authors, hence annotated with similar styles and sets of keywords. To mitigate such problem, we randomly select a subset of 1,000 annotated documents (except DAAC dataset, which only has 978 documents of land terrestrial ecology, hence we select them all.) from each archive for our experiments. We combine all the textual attributes (i.e. `Title`, `Abtract`, `Description`) together as the textual content for the document. We preprocess the textual content in each document by removing 664 common stop words and punctuation, and stemming the words using the Porter2's[13] stemming algorithm.

4. PRELIMINARIES

Our proposed solution is built upon the concepts of Cosine Similarity, Term Frequency-Inverse Document Frequency (TF-IDF), and Latent Dirichlet Allocation (LDA). We briefly introduce them here to fortify readers' background before going further.

4.1 Cosine Similarity

In general, cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. Given two vectors A and B, the cosine similarity is defined using a dot product and magnitude as:

$$CosineSim(A,B) = \frac{A \cdot B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{N} A_i \times B_i}{\sqrt{\sum_{i=1}^{N}(A_i)^2} \times \sqrt{\sum_{i=1}^{N}(B_i)^2}}$$

`CosineSim(A,B)` outputs `[0,1]`, with 0 indicating independence, and the value in between indicates the level of similarity. In information retrieval literature, the cosine similarity is heavily used to calculate the similarity between two vectorized documents.

4.2 Term Frequency-Inverse Document Frequency

TF-IDF is used extensively in the information retrieval area. It reflects how important a term is to a document in a corpus. TF-IDF has two components: the term frequency (TF) and the inverse document frequency (IDF). The TF is the frequency of a term appearing in a document. The IDF of a term measures how important the term is to the corpus, and is computed based on the document frequency, the number of documents in which the term appears. Formally, given a term t, a document d, and a corpus (document collection) D:

$$tf(t,d) = \sqrt{count(t,d)}$$
$$idf(t,D) = \sqrt{log\left(\frac{|D|}{|d \in D; t \in d|}\right)}$$
$$TFIDF(t,d,D) = TF(t,d) \cdot IDF(t,D)$$

We can then construct a TF-IDF vector for a document `d` given a corpus `D` as follows:

$$TFIDF(d,D) = \langle TFIDF(t_1,d,D), \cdots, TFIDF(t_n,d,D) \rangle$$

Consequently, if one wishes to compute the similarity score between two documents d_1 and d_2, the cosine similarity can be computed between the TF-IDF vectors representing the two documents:

$$DocSim_{TF-IDF}(d_1,d_2,D) = CosineSim\left(TFIDF(d_1,D), TFIDF(d_2,D)\right) \quad (1)$$

[12]http://www.phylofoundation.org/
[13]http://snowball.tartarus.org/algorithms/english/stemmer.html

4.3 Latent Dirichlet Allocation

In text mining, the Latent Dirichlet Allocation (LDA) [3] is a generative model that allows a document to be represented by a mixture of topics. Past literature such as [10, 20, 21] demonstrates successful usage of LDA to model topics from given corpora. The basic intuition of LDA for topic modeling is that an author has a set of topics in mind when writing a document. A topic is defined as a distribution of terms. The author then chooses a set of terms from the topics to compose the document. With such assumption, the whole document can be represented using a mixture of different topics. LDA serves as a means to trace back the topics in the author's mind before the document is written. Mathematically, the LDA model is described as follows:

$$P\left(t_i|d\right) = \sum_{j=1}^{|Z|} P\left(t_i|z_i = j\right) \cdot P(z_i = j|d)$$

$P(t_i|d)$ is the probability of term t_i being in document d. z_i is the latent (hidden) topic. $|Z|$ is the number of all topics. This number needs to be predefined. $P(t_i|z_i = j)$ is the probability of term t_i being in topic j. $P(z_i = j|d)$ is the probability of picking a term from topic j in the document d.

Essentially, the LDA model is used to find $P(z|d)$, the topic distribution of document d, with each topic is described by the distribution of term $P(t|z)$. After the topics are modeled, we can assign a distribution of topics to a given document using a technique called *inference*. A document then can be represented with a vector of numbers, each of which represents the probability of the document belonging to a topic.

$$Infer(d, Z) = \langle z_1, z_2, ..., z_Q \rangle; |Z| = Q$$

Where Z is a set of topics, d is a document, and z_i is a probability of the document d falling into topic i. Since a document can be represented using a vector of numbers, one can then compute the topic similarity between two documents d_1 and d_2 using cosine similarity as follows:

$$\begin{aligned} DocSim_{TM}(d_1, d_2, Z) = \\ CosineSim\left(Infer(d_1, Z), Infer(d_2, Z)\right) \end{aligned} \tag{2}$$

5. METHOD

The metadata annotation problem is transformed into the tag recommendation with a controlled tag library. A document is a tuple of textual information and a set of tags, i.e. $\langle text, tags \rangle$. A document query is a document without tags, $\langle text, \oslash \rangle$. Specifically, given a tag library $T = \langle t_1, t_2, ..., t_m \rangle$, a document corpus $D = \langle d_1, d_2, ..., d_n \rangle$, and a document query q, the algorithm outputs a ranked list $T_K^* = \langle t_1, t_2, ..., t_K \rangle$, where $t_i \in T$, of K tags relevant to the document query q.

Our proposed algorithm comprises 2 main steps:

STEP1 $P(t|q, T, D, M)$, the probability of tag t being relevant to q, is computed for each $t \in T$. M is the document similarity measure, which can be either *TF-IDF* or *TM*.

STEP2 Return top K tags ranked by the $P(t|q, T, D, M)$ probability.

$P(t|q, T, D, M)$ is the normalization of the relevance score of the tag t to the document query q, and is defined below:

$$\begin{aligned} P(t|q, T, D, M) &= \frac{TagScore_M(t, q, D)}{\sum_{\tau \in T} TagScore_M(\tau, q, D)} \\ TagScore_M(t, q, D) &= \sum_{d \in D} DocSim_M(q, d, D) \cdot isTag(t, d) \end{aligned}$$

$TagScore_M(t, q, D)$ calculates the tag score determining how relevant the tag t is to document query q. This score can be any real non-negative number. $DocSim_M(q, d, D)$ measures the similarity between two documents, i.e. q and d, given a document corpus D and returns a similarity measure ranging between [0,1]. $isTag(t, d)$ is a binary function that returns 1 if $t \in d.tags$ and 0 otherwise. We propose two approaches to compute the document similarity: *Term Frequency-Document Inverse Frequency (TF-IDF)* based ($DocSim_{TF-IDF}(q, d, D)$) and *Topic Modeling (TM)* based ($DocSim_{TM}(q, d, D)$). These two approached are described in the next subsections.

5.1 TF-IDF based $DocSim_{TF-IDF}(q, d, D)$

The TF-IDF based document similarity measure relies on the term frequency-inverse document frequency principle discussed in Section 4.2. The function aims to quantify the content similarity based on term overlap between two documents. In order to compute the IDF part of the scheme, all the documents in D are first indexed. Hence the training phase (preprocess) involves indexing all the documents. We then compute the similarity between the query q and a source document d using $DocSim_{TF-IDF}(q, d, D)$ as defined in Equation 1.

5.2 TM based $DocSim_{TM}(q, d, D)$

The TM based document similarity measure utilizes topic distributions of the documents using the LDA algorithm as described in Section 4.3. The algorithm further extracts the semantic reposing within a document captured by its topic distribution. With this knowledge in mind, one can measure the semantic similarity between two documents by quantifying the similarity between their topic distributions. Indeed, our proposed TM based algorithm transforms the topic distribution of a document into a numerical vector, wherein Cosine similarity is used to compute the topic similarity between two documents using Equation 2.

6. EVALUATION AND DISCUSSION

We evaluate our methods using the tag prediction protocol. We artificially create a test query document by removing the tags from an annotated document. The task is to predict the removed tags. There are two reasons behind the choosing of this evaluation scheme:

1. The evaluation can be done fully automatically. Since our datasets are large, manual evaluation (i.e. having human identify whether a recommended tag is relevant or not) would be infeasible.

2. The evaluation can be done against the existing gold standard established (manually tagged) by expert annotators (i.e. data collectors, project principal investigators, etc.) who have good understanding about the

data, while manual evaluation could lead to evaluation biases.

We evaluate our TF-IDF and TM based algorithms against the baseline KEA document annotation algorithm with controlled vocabulary. In our setting, the tag library is used as the vocabulary in the KEA algorithm. The document-wise 10 fold cross validation is performed, where each dataset is first split into 10 equal subsets, and for each fold $i \in \{1, 2, 3, ..., 10\}$ the subset i is used for the testing set, and the other 9 subsets are combined and used as the source (training set). The results of each fold are summed up and the averages are reported.

For the TF-IDF based algorithm, we use LingPipe[14] to perform the indexing and calculating the TF-IDF based similarity. For the TM based algorithm, the training process involves modeling topics from the source using LDA algorithm as discussed in Section 4.3. We use the Stanford Topic Modeling Toolbox[15] with the collapsed variational Bayes approximation[2] to identify topics in the source documents. For each document we generate uni-grams, bi-grams, and tri-grams, and combine them to represent the textual content of the document. The algorithm takes two input parameters: the number of topics to be identified and the maximum number of the training iterations. After some experiments on varying the two parameters, we fix them at 300 and 1,000 respectively. The inference method proposed by Asuncion et al. [2] is used to assign a topic distribution to a given document. The evaluation is done on a Windows 7 PC with Intel Core i7 2600 CPU 3.4 GHz and 16GB of ram.

6.1 Evaluation Metrics

This section presents the evaluation metrics used in our tasks, including precision, recall, F1, Mean Reciprocal Rank (MRR), and Binary Preference (Bpref). These metrics, when used in combination, have shown to be effective for evaluation of recommending systems[8, 13, 25].

6.1.1 Precision, Recall, F1

Precision, recall, and F1 (F-measure) are well-known evaluation metrics in information retrieval literature [14]. For each document query in the test set, we use the original set of tags as the ground truth T_g. Assume that the set of recommended tags are T_r , so that the correctly recommended tags are $T_g \bigcap T_r$. Precision, recall and F1 measures are defined as follows:

$$precision = \frac{|T_g \bigcap T_r|}{|T_r|}, recall = \frac{|T_g \bigcap T_r|}{|T_g|}, F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

In our experiments, the number of recommended tags ranges from 1 to 30. It is wise to note that better tag recommendation systems tend to rank correct tags higher than the incorrect ones. However, the precision, recall, and F1 measures do not take ranking into account. To evaluate the performance of the ranked results, we employ the following evaluation metrics.

6.1.2 Mean Reciprocal Rank (MRR)

MRR[22] measure takes ordering into account. It measures how well the first correctly recommended tag is ranked. The reciprocal rank of a query is the multiplicative inverse of the rank of the first correctly recommended tag. The mean reciprocal rank is the average of the reciprocal ranks of the results of the query set Q. Formally, given a testing set Q, let $rank_q$ be the rank of the first corrected answer of query $q \in Q$, then MRR of the query set Q is defined as follows:

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{rank_q}$$

If the set of recommended tags does not contain a correct tag at all, $\frac{1}{rank_q}$ is defined to be 0.

6.1.3 Binary Preference (Bpref)

Bpref measure considers the order of each correctly recommended tag [5]. Let S be the set of recommended tags by the system, R be the set of corrected tags , $r \in R$ be a correct recommendation, and $i \in S - R$ be an incorrect recommendation. The Bpref is defined as follows:

$$Bpref = \frac{1}{|R|} \sum_{r \in R} 1 - \frac{|i \text{ ranked higher than } r|}{|S|}$$

Bpref can be thought of as the inverse of the fraction of irrelevant documents that are retrieved before relevant ones. Bpref and mean average precision (MAP) are similar when used with complete judgments. However, Bpref normally gives a better evaluation when used in a system with incomplete recommendations.

6.2 Results

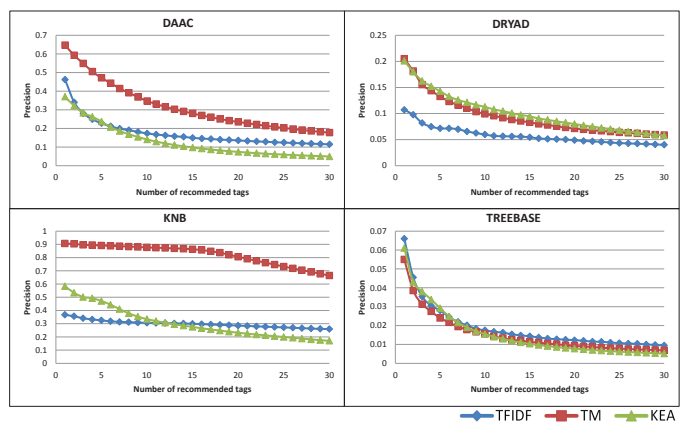

Figure 2: Precision of the TF-IDF, TM, KEA (baseline) algorithms on the 4 datasets.

Figures 2,3,4 plot the precision@K, recall@K, F1@K respectively evaluated at the top K recommended tags of the proposed TF-IDF and TM based algorithms against the baseline KEA algorithm on each dataset. Figure 5 summarizes the precision versus recall on each dataset.

According to the results, our proposed algorithms outperform the baseline KEA algorithm on the DAAC and KNB datasets (TM based approach outperforms at every K and TF-IDF based approach outperforms at larger K). This is because the tags used to annotate DAAC and KNB documents are drawn from the libraries of topics. Hence there is a high chance that a tag is reused for multiple times, resulting in high tag utilization. Since our algorithms tend to

[14]http://alias-i.com/lingpipe/
[15]http://nlp.stanford.edu/software/tmt/tmt-0.4/

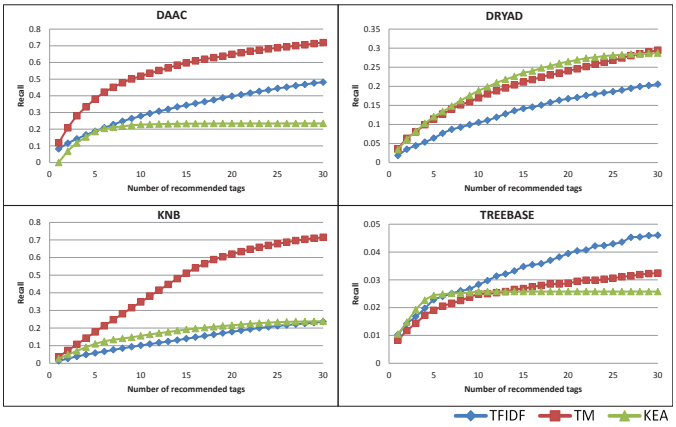

Figure 3: Recall of the TF-IDF, TM, KEA (baseline) algorithms on the 4 datasets.

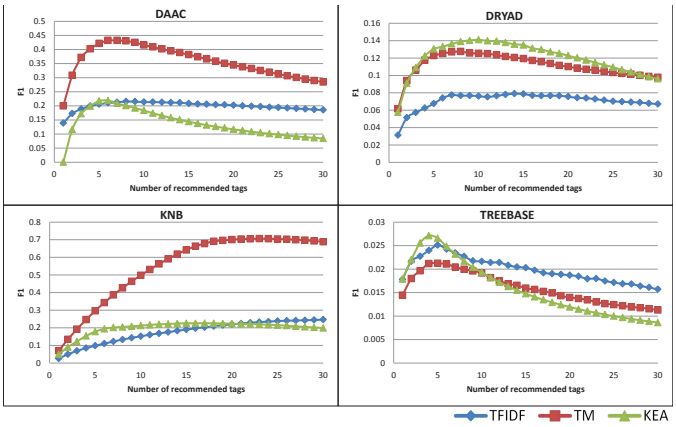

Figure 4: F1 of the TF-IDF, TM, KEA (baseline) algorithms on the 4 datasets.

give higher weight to tags that have been used frequently, datasets with high tag utilization (such as DAAC and KNB) tend to benefit from our algorithms.

However, our proposed algorithms tend to perform worse than the baseline on the DRYAD dataset. This is because, tags in each DRYAD document are manually made up at the curation process. Manually making up tags for each document results in a large size of tag library where each tag is used only a few times, leading to the low tag utilization. Datasets with low tag utilization do not tend to benefit from our proposed algorithms since the probability distribution given to the tags tends to be uniform and not very discriminative.

All the algorithms perform poorly on the TreeBASE dataset. This is because TreeBASE documents are very sparse (some do not even have textual content), and have very few tags. From the dataset statistics, each document on the TreeBASE dataset has only 11 words and only 0.7 tags on average. Such sparse texts lead to weak relationship when finding textually similar documents in the TF-IDF based approach, and the poor quality of the topic model used by the TM based approach. The small number of tags per document makes it even harder to predict the right tags.

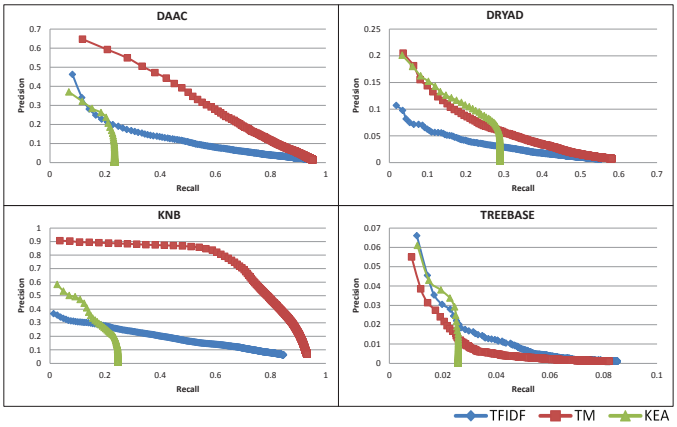

Figure 5: Precision vs Recall of the TF-IDF, TM, KEA (baseline) algorithms on the 4 datasets.

Dataset	Method	MRR	Bpref	ALT (s)	ATT (s)
DAAC	TFIDF	0.5649	0.8183	5.29	0.99
	TM	0.7546	0.9005	2430.82	49.63
	KEA	0.5109	0.2337	75.45	6.50
DRYAD	TFIDF	0.2022	0.4404	6.32	1.19
	TM	0.3264	0.4934	4486.09	83.76
	KEA	0.3423	0.2851	102.59	8.85
KNB	TFIDF	0.4944	0.6659	6.06	1.23
	TM	0.9226	0.9100	1159.81	49.60
	KEA	0.6823	0.2431	64.11	5.61
TREEBASE	TFIDF	0.0893	0.0695	6.15	1.08
	TM	0.0750	0.0636	401.50	16.99
	KEA	0.0745	0.0257	6.26	1.04

Figure 6: MRR, BPref, Average Learning Time (ALT) and Average Test Time (ATT) of TF-IDF, TM, KEA (baseline) algorithms on the 4 datasets.

Figure 6 lists the MRR, BPref, average learning time (in seconds) per fold, and average testing time (in seconds) per fold of the proposed TF-IDF and TM based algorithms against the baseline KEA algorithm on each dataset. MRR quantifies how the first correct recommendation is ranked. In terms of MRR, our TM based algorithm performs the best on the DAAC and KNB datasets, TF-IDF based algorithm performs the best in the TreeBASE dataset, and the KEA algorithm performs the best on the DRYAD dataset. The TM based algorithm achieves the notable MRR scores of 0.75 and 0.92 on the DAAC and KNB datasets receptively, and outperforming the baseline by 47.70% and 33.22% respectively.

Bpref measures the ranking of all the correctly recommended keywords. In terms of Bpref, our TM algorithm performs the best on the DAAC, DRYAD, and KNB datasets with the Bpref scores of 0.90, 0.49, and 0.91 respectively. The TF-IDF based algorithm performs the best on the TreeBASE dataset. Similar to the MRR results, notable BPref scores are achieved by the TM based algorithm on the DAAC and KNB datasets, outperforming the baseline by 285.32% and 274.33% respectively.

Figure 7 shows sample results of our proposed TF-IDF/TM based algorithms and the KEA algorithm against the 15 actual ground-truth tags associated to an DAAC metadata ti-

Actual Tags	TFIDF	TM	KEA(Baseline)
[1]albedo [2]land cover [3]veget cover [4]veget index [5]leaf area meter [6]terra morn equatori cross time satellit [7]noaa nation ocean amp amp atmospher administr [8]plant characterist [9]steel measur tape [10]canopi characterist [11]modi moder resolut imag spectroradiomet [12]leaf characterist [13]avhrr advanc high resolut radiomet [14]field investig [15]reflect	[1]**field investig** [2]analysi [3]**land cover** [4]comput model [5]**reflect** [6]**veget cover** [7]biomass [8]primari product [9]**steel measur tape** [10]weigh balanc [11]precipit amount [12]**canopi characterist** [13]**leaf characterist** [14]water vapor [15]quadrat sampl frame [16]rain gaug [17]surfac air temperatur [18]air temperatur [19]meteorolog station [20]human observ	[1]**land cover** [2]**modi moder resolut imag spectroradiomet** [3]**terra morn equatori cross time satellit** [4]**field investig** [5]**veget cover** [6]**reflect** [7]**veget index** [8]**leaf characterist** [9]**canopi characterist** [10]**plant characterist** [11]**albedo** [12]**steel measur tape** [13]**avhrr advanc high resolut radiomet** [14]**noaa nation ocean amp amp atmospher administr** [15]**leaf area meter** [16]analysi [17]comput model [18]noaa [19]avhrr [20]popul distribut	[1]model [2]geograph distribut [3]classif [4]lba [5]amazonia [6]area [7]south america [8]ecolog [9]**reflect** [10]calibr [11]field investig [12]speci [13]factor [14]sequenc [15]hawaiian island [16]genera [17]fern [18]systemat [19]**steel measur tape** [20]correl

Figure 7: Comparison of the recommended keywords by the TF-IDF, TM, and KEA (baseline) algorithms on a sample document "ISLSCP II IGBP DISCOVER AND SIB LAND COVER, 1992-1993". The first column lists the actual tags. The bold, underlined terms are correctly recommended items.

tled "*ISLSCP II IGBP DISCOVER AND SIB LAND COVER, 1992-1993*"[16]. Our TM based algorithm performs well on this particular example by capturing all the actual tags within the top 15 recommended tags.

6.3 TM vs TF-IDF Based Approaches

According to the results, our TM based approach performs better than TF-IDF based approach on DAAC, DRYAD, and KNB datasets, in terms of precision, recall, and F1 measure, while the TF-IDF based approach performs better on the TreeBASE dataset. Since the only difference between the two proposed methods is the document similarity function $DocSim(q, d, D)$, which computes the similarity score between the query document q and a source document $d \in D$, the analysis on the differences between the two document similarity measures could provide explanation about the performance difference.

The TF-IDF document similarity measures the cosine similarity between two TF-IDF vectors representing the two documents. Loosely speaking, the TF-IDF document similarity measures the quantity of term overlap, where each term has a different weight, in the two documents.

The TM based approach first derives a set of topics from the document source, each of which is represented by a distribution of terms. The ranked terms in each topic bare coherent semantic meanings. Figure 8 provides an example of the top 10 terms in sample 9 topics derived from the DAAC dataset using the LDA algorithm with 300 topics and 1,000 iterations. Once the set of topics has been determined, a document is assigned a distribution of topics using the inference algorithm mentioned in Section 4.3. The TM document similarity then measures the cosine similarity between the topic distribution vectors representing the two documents. Loosely speaking, the TM document similarity quantifies the topic similarity between the two documents.

The difference in performance of both the proposed methods could be impacted by the semantic representation of each document. It is evident from the experimental results on the DAAC, DRYAD, and KNB datasets, that representing a document with a mixture of topics lead to a more accurate semantic similarity interpretation, leading to better recommendation. However, the reason why the TM based approach performs worse than the TF-IDF based approach on the TreeBASE dataset could be because the documents in such dataset are very sparse (Each TreeBASE document has only 11 words on average). Such sparsity could lead to a poor set of topics, consisting of idiosyncratic word combinations.

Hence we recommend the TM based algorithm for datasets whose documents are rich in textual documents, and the TF-IDF based algorithm approach for those with textually sparse documents.

6.4 Limitations

Regardless of the promising performance, our proposed document annotation algorithms have the following limitations:

1. Our algorithms rely on the existence of a good document source (training set). The quality of the resulting annotation directly reflects the quality of the annotation of each document in the training data. Fortunately, the current ONEMercury system only retrieves the metadata from the archives wherein each metadata document is manually provided by Principal investigators and data managers. In the future, however, the system may expand to collect the metadata from the sources in which the data records may have poor or no annotation. Such problems urge the need for a

[16] $http://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id = 930$

Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic 9
amazon	pine	aboveground	park	carbon dioxide	soil moisture	plant	bigfoot	environment
river	team	year	mongu	ssa	site average	water potential	modi product	radar
basin	black spruce	mortal	safari	energy	moisture bett	determin	evergreen	mosaic
ecology	chamber	woody biomass	zambia	exchange	neutron	leaf optic	land cover	geotiff format
floodplain	methane	hardwood	wet season	soil temperature	airborn	radiometer	bigfoot project	band
band	team collect	brown	photosynthetic	team	fife experiment	plant water	nasa	backscatter image
inundate	chamber flux	eastern	kalahari	tower flux	gamma	grass	flux tower	synthesize
meter	trace gas	softwood	botswana	vapor	gravimeter	leaf water	grassland	topograph
scale	plant	schroeder	store	water vapor	water content	leaf tissue	evergreen needleleaf	srtm
mosaic	spruce	commercial	activity	flux measure	gas flux	summer	ecology	develop

Figure 8: Top 10 terms in sample 9 topics derived using LDA algorithm from the DAAC dataset.

method that allows the automatic annotator trained with a high-quality training dataset to annotate the documents in different datasets. We plan to investigate into such cross-source recommendation in our future work.

2. Our TM based algorithm needs to model topics from scratch every time a significant amount of new documents are added to the training corpus, so that the modeled topics can reflect the new documents added. Since our TM based algorithm utilizes the traditional LDA algorithm to model topics, wherein incremental training is not a feature, we plan to explore into methods such as [1] and [9] which may enable our algorithm to adaptively model the topics from a dynamic corpus.

3. Regardless of the promising performance of our proposed TM based algorithm, the scalability can be an issue when it comes to mining topics from a larger corpus of documents. The scalability issues of our TM based algorithm is discussed in detail in the next sub section.

6.5 Scalability of the TM Approach

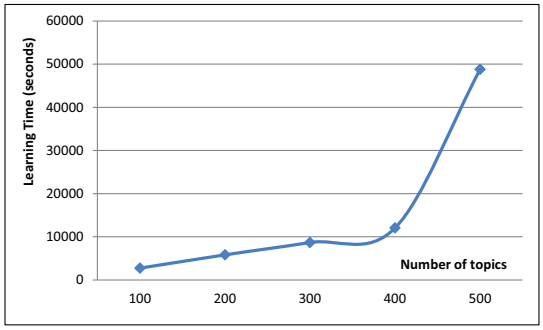

Figure 9: Learning time in seconds of the TM based algorithm as a function of numbers of topics.

Scalability issues should be taken into account since the algorithms will eventually be cooperated as part of the ONEMercury system, which currently hosts much larger datasets than the ones we use in the experiments. This section presents two scalability issues presented in the TM based algorithm: the increase in number of topics and the increase in size of the corpus.

We examine the scalability issues our TM based algorithm on the KNB dataset, using the Stanford Topic Modeling Toolbox with collapsed variational Bayes approximation and fixed 1,000 iterations, on the same machine we use for earlier experiments.

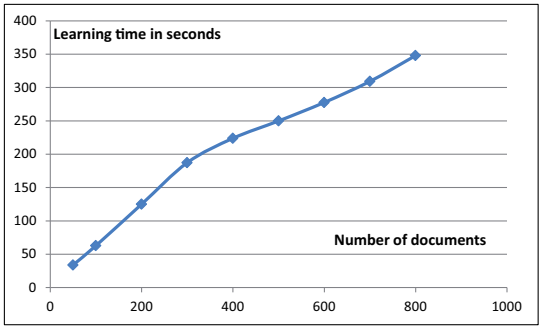

Figure 10: Learning time in seconds of the TM based algorithm as a function of numbers of training documents.

As the data grows larger, new topics emerge, urging the need for a new model that captures such increasing variety of topic. Figure 9 plots the training time (in seconds) as the function of number of topics. The training time grows approximately linearly with the number of topics up to 400 topics. The program runs out of physical memory, however, at 500 topics, leading to a dramatic increase in the training time. Hence, this study points out a more memory-efficient topic model algorithm should be explored.

Another scalability concern lies with the projected increase in the number of training documents. Figure 10 shows the training time of the TM based algorithm as the number of document increases. The results also show a linear scale with the number of training documents. Note that the experiment is only done with up to 1,000 documents, while there are roughly 47 thousands, and definitely increasing in the future, metadata records in the current system. Even with the current size of the ONEMercury repository, the algorithm would take approximately 5.3 hours to model topics, which is not feasible in practice. Hence a large-scale parallel algorithm such as MapReduce[6] should be investigated.

7. CONCLUSION AND FUTURE WORK

This paper presents a set of algorithms for automatic annotation of metadata. We are motivated by the real world problems faced by ONEMerucy, a search system for environmental science metadata harvested from multiple data archives. One of the important problems includes the different of levels of curation of metadata from different archives,

which means that the system must automatically annotate poorly annotated metadata records. We treat each metadata record as a tagged document, and then transform the problem into the tag recommendation problem with a controlled tag library.

We propose two algorithms for tag recommendation, one based on term frequency-inverse document frequency (TF-IDF) and the other based on topic modeling (TM) using the Latent Dirichlet Allocation. The evaluation is done on 4 different datasets of environmental metdata using the tag prediction evaluation protocol, against the well known KEA document annotation algorithm. The results show that our TM based approach yields better results on datasets characterized to have high tag utilization and rich in textual content such as DAAC and KNB than those which do not (i.e. DRYAD abd TreeBASE), though with the cost of longer learning times. The scalability issues of the TM based algorithm necessitate investigation into more memory-efficient and scalable approaches. Finally, future steps could be implementing an automatic metadata annotation algorithm on the ONEMercury search service or explore online tagging[19].

8. ACKNOWLEDGMENTS

We gratefully acknowledge useful comments from Natasha Noy, Jeffery S. Horsburgh, and Giri Palanisamy. This work has been supported in part by the National Science Foundation (DataONE: Grant #OCI-0830944) and Contract No. De-AC05-00OR22725 of the U.S. Department of Energy.

9. REFERENCES

[1] L. AlSumait, D. Barbar, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM*, pages 3–12. IEEE Computer Society, 2008.

[2] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[4] M. Bron, B. Huurnink, and M. de Rijke. Linking archives using document enrichment and term selection. In *Proceedings of the 15th international conference on Theory and practice of digital libraries: research and advanced technology for digital libraries*, TPDL'11, pages 360–371, 2011.

[5] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 25–32, New York, NY, USA, 2004. ACM.

[6] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[7] 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*, SIGIR '08, pages 531–538, New York, NY, USA, 2008. ACM.

[8] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach. Recommending citations: translating papers into references. CIKM '12, pages 1910–1914, New York, NY, USA, 2012. ACM.

[9] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 663–672, New York, NY, USA, 2010. ACM.

[10] S. Kataria, P. Mitra, and S. Bhatia. Utilizing context in generative bayesian models for linked corpus. In *AAAI'10*, pages –1–1, 2010.

[11] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 61–68, New York, NY, USA, 2009. ACM.

[12] Z. Liu, X. Chen, and M. Sun. A simple word trigger method for social tag suggestion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1577–1588, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[13] Z. Liu, W. Huang, Y. Zheng, and M. Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[15] O. Medelyan and I. H. Witten. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, pages 296–297, New York, NY, USA, 2006. ACM.

[16] W. Michener, D. Vieglais, T. Vision, J. Kunze, P. Cruse, and G. Janée. DataONE: Data Observation Network for Earth - Preserving Data and Enabling Innovation in the Biological and Environmental Sciences. *DLib Magazine*, 17(1/2):1–12, 2011.

[17] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 953–954, New York, NY, USA, 2006. ACM.

[18] D. Newman, K. Hagedorn, C. Chemudugunta, and P. Smyth. Subject metadata enrichment using statistical topic models. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '07, pages 366–375, New York, NY, USA, 2007. ACM.

[19] 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*, SIGIR '08, pages 515–522, 2008.

[20] S. Tuarob, L. C. Pouchard, N. Noy, J. S. Horsburgh, and G. Palanisamy. Onemercury: Towards automatic annotation of environmental science metadata. In *Proceedings of the 2nd International Workshop on Linked Science*, LISC '12, 2012.

[21] S. Tuarob and C. S. Tucker. Fad or here to stay: Predicting product market adoption and longevity using large scale, social media data. In *Proc. ASME 2013 Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, IDETC/CIE '13, 2013.

[22] E. M. Voorhees. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[23] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM.

[24] L. Wu, L. Yang, N. Yu, and X.-S. Hua. Learning to tag. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 361–370, 2009.

[25] T. Zhou, H. Ma, M. Lyu, and I. King. Userrec: A user recommendation framework in social tagging systems. In *Proceedings of AAAI*, pages 1486–1491, 2010.