Expert Systems With Applications 42 (2015) 9587–9602

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Construction and evaluation of ontological tag trees

Chetan Verma[a,b,*], Vijay Mahadevan[a], Nikhil Rasiwasia[a], Gaurav Aggarwal[a], Ravi Kant[a], Alejandro Jaimes[a], Sujit Dey[b]

[a] Yahoo Labs, Bengaluru, India
[b] University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093, United States

ARTICLE INFO

Keywords:
Tag tree
Local search
Optimization
Tag prediction
Classification

ABSTRACT

Several expert systems have been proposed to address the sparsity of tags associated with online content such as images and videos. However most of such systems either necessitate extracting domain-specific features, or are solely based on tag semantics, or have significant space requirements to store corpus based tag statistics. To address these shortcomings, in this work we show how ontological tag trees can be used to encode information present in a given corpus pertaining to interaction between the tags, in a space efficient manner. An ontological tag tree is defined as an undirected, weighted tree on the set of tags where each possible tag is treated as a node in the tree. We formulate the tag tree construction as an optimization problem over the space of trees on the set of tags and propose a novel local search based approach utilizing the co-occurrence statistics of the tags in the corpus. To make the proposed optimization more efficient, we initialize using the semantic relationships between the tags. The proposed approach is used to construct tag trees over tags for two large corpora of images, one from Flickr and one from a set of stock images. Extensive data-driven evaluations demonstrate that the constructed tag trees can outperform previous approaches in terms of accuracy in predicting unseen tags using a partially observed set of tags, as well as in efficiency of predicting all applicable tags for a resource.

1. Introduction

The consumer electronic revolution and the Internet have led to the availability of vast amounts of data including multimedia data such as images and videos. A significant fraction of such data is user generated content, in the form of pictures and videos uploaded onto sites such as Facebook (2015), Flickr (2015) and YouTube (2015). Owing to the fact that there are minimal requirements when uploading the content and that mobile uploads are on the rise, users rarely add any extra information such as a textual description to the content. At best, most images and videos are tagged with certain keywords. As these keywords or tags are sometimes applied to entire albums of images or videos at once, or applied in error, the information provided by such tags is quite noisy.

Some examples of images having incorrect tags (as per human experts) are shown in Fig. 1. The massive scale of data and the lack of useful metadata makes it difficult for users to access data that may be of interest to them (Anand & Mampilli, 2014; Jiang, Qian, Shen, Fu & Mei, 2015; Zheng & Li, 2011).

The social tagging at the above mentioned data sharing websites creates a Folksonomy (Hsieh, Stu, Chen, & Chou, 2009; Kim & Kim, 2014; Sun, Wang, Sun, & Lin, 2011) which mitigates the information overload to some extent by creating non-hierarchical categories or indexes for the retrieval of data. Folksonomies make it scalable to assign labels to large volumes of data in a collaborative manner and are hence more appropriate for such data than traditional taxonomies established by expert cataloguers (Kim & Kim, 2014). At the same time, collaboratively produced folksonomies have several issues, particularly related to incorrect tags and their sparsity (Sun et al., 2011; Uddin, Duong, Nguyen, Qi, & Jo, 2013). While incorrect tags have been discussed earlier, the sparsity in folksonomy arises as a result of lack of incentive for the users to tag the resources comprehensively and completely. As a result, the online resources are typically associated with low number of tags, preventing effective searching and browsing through the available data (Uddin et al., 2013).

In order to address the sparsity in folksonomies, several expert systems have been proposed that recommend or suggest additional tags for a resource based on the tags already associated with the resource (Chen, Liu, & Sun, 2015; Hsieh et al., 2009; Sigurbjörnsson & Van Zwol, 2008; Sun et al., 2011). Most of such works depend on the availability of content-based features such as textual features from

[*] Correspondence to: 1063 Morse Avenue, Apt 3-306, Sunnyvale, CA 94089, United States. Tel.: +1 858 344 9473.

E-mail addresses: cverma@ucsd.edu (C. Verma), vmahadev@yahoo-inc.com (V. Mahadevan), nikux@yahoo-inc.com (N. Rasiwasia), gaggarwa@yahoo-inc.com (G. Aggarwal), rkant@yahoo-inc.com (R. Kant), ajaimes@yahoo-inc.com (A. Jaimes), dey@ece.ucsd.edu (S. Dey).

http://dx.doi.org/10.1016/j.eswa.2015.07.057
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

(a) 'animal', **'car'** (b)**'wedding'**, 'mushroom' (c) 'farms', **'snow'**

Fig. 1. Some examples of incorrect tags given by users on www.flickr.com. (a) An image of a 'cat' tagged as 'car', which most likely is a spelling mistake, (b) an image of a 'mushroom' also tagged as 'wedding' and (c) an image of a 'goat' tagged with 'snow'. The Flickr owner and photo ids of these images are (8656572@N04,4670326818), (35468147887@N01,252474171), and (39405339@N00,5835556089), respectively.

documents or blogs (Chen et al., 2015; Hsieh et al., 2009; Sun et al., 2011), or visual features from images or videos (Xia, Feng, Peng, Wu, & Fan, 2015), and thus cannot be applied to other domains. In addition, extracting and utilizing content based features is known to be computationally expensive and for certain domains, even infeasible (Huang, Fu, & Chen, 2010; Song et al., 2010; Yin, Li, Mei, & Han, 2009; Zanetti, Zelnik-Manor, & Perona, 2008), and so the above works may not be applicable to such domains. Furthermore, expert systems such as Uddin et al. (2013) utilize purely semantic relationships between tags. While semantic relations as obtained from ontologies such as WordNet (Miller, 1995) or Open Directory Project (2015) are an important resource for linguistic and machine learning related problems, such relationships fail to capture the information that is characteristic of an available corpus. Consider for instance a corpus of annotated images from Flickr. The co-occurrence of tags in a given corpus provides interesting insights into the nature of the data. For example, the 2008 Olympics were held in Beijing and as a result, there exist a large number of images in Flickr having '2008' and 'Beijing' as their tags. Such a relation between '2008' and 'Beijing' cannot be obtained from WordNet or similarly formed hierarchies (such as Open Directory Project (2015)) because the semantic relations in the above hierarchies are pre-defined, and do not account for a connection between the two tags. In addition to the above expert systems, works such as Sigurbjörnsson and Van Zwol (2008) capture tag similarities from a given dataset using tag graphs. Tag graphs usually refer to complete graphs representing pair-wise distances or similarities between tags, which are calculated from a given corpus. For certain applications, a threshold is applied and only the most important pair-wise connections are retained. However, storing similarities using tag graphs has several issues. Firstly, the pre-defined threshold value that is chosen to construct them can be arbitrary and there is no clear understanding to what its value should be. The pair-wise edges that have their similarity above the defined threshold are the only ones that are retained in the graph and this leads to completely losing of information of those pairs of concepts or tags that have their similarity below the threshold. Depending on the threshold value, the space requirement of tag graphs can vary as $O(N^2)$ where N is the number of concepts or tags in the tag graph, which can be significantly high for large number of tags. In order to keep a handle on the space requirement, a strict threshold value can be chosen which would result in losing pair-wise similarity information for several pairs of concepts or tags. Lastly, depending on the threshold, it is possible that some concepts or tags are disconnected from the rest. This again implies losing relationship information of the concept or tag with others. Sigurbjörnsson and Van Zwol (2008) estimate the number of tags in Flickr in 2008 to be 3.7 million. Storing each similarity value as a floating point occupying 4 bytes would require more than 27 terabytes just to store the pair-wise relationships.

We attempt to address the above shortcomings in this paper. We use the term ontological tag tree or simply tag tree to denote undirected weighted tree of concepts (or tags) where the relationships between the concept nodes in the tree are defined only in terms of a scalar weight. As compared to tag graphs (Liu, Hua, Yang, Wang, & Zhang, 2009; Sigurbjörnsson & Van Zwol, 2008), ontological tag trees are necessarily trees on the set of tags, i.e., are connected and have no simple cycles. We have chosen a spanning tree to represent the relationships between tags because a spanning tree over the set of tags is necessarily connected and does not lead to losing of information due to possibly disconnected components as in tag graphs. Also, the space requirement of a spanning tree is only $O(N)$ for N tags. For 3.7 million tags (Sigurbjörnsson & Van Zwol, 2008), this implies a significant reduction in the space requirement from 27 terabytes ($O(N^2)$) to less than 50 megabytes ($O(N)$). As a result, expert systems can be implemented even on computing devices that do not have a gigantic memory. Ontological tag trees are constructed using the semantic and the data-driven relations between the tags and hence lead to significantly better performance on data-driven tasks than using solely semantic relationships between tags (Miller, 1995; Uddin et al., 2013). For the constructions of tag trees, we do not utilize content based features, rather we utilize data-driven similarities from tag co-occurrences in the given annotated corpus. As a result, compared to previous expert systems that require extracting and processing content-based (such as visual or textual) features (Chen et al., 2015; Hsieh et al., 2009; Sun et al., 2011; Xia et al., 2015), tag trees can be used to alleviate sparsity in online folksonomies even in domains where extracting domain specific features may be infeasible or inefficient. This also makes the construction approach not married to a single domain such as annotated text documents/blogs or videos or images.

We illustrate the proposed tag tree construction approach using two large image corpora – one obtained from Flickr, and the other obtained from a set of stock images, with the goal of obtaining a tag tree over the set of tags present in these corpora. For these corpora, the co-occurrence count for a pair of tags is defined as the number of images with which both tags are associated. The normalized co-occurrence counts are a measure of how related two tags are. We assume that the concepts or nodes of the tag tree are the tags, and that the tree construction task is to infer the relations between the tags. The task thus becomes a graph learning problem where the nodes of the graph are the tags, and the relations between tags are represented by undirected edges and their weights in the graph. Unlike the relationships given in ontologies, we do not attempt to give semantic interpretations to the relations between tags. To solve the graph learning problem, we formulate an optimization problem on the space of spanning trees of a suitably constructed Similarity Graph that is based on semantic relations between tags, as obtained from WordNet, and on the normalized co-occurrence counts of the corpus. We solve the optimization problem using the 'local search' paradigm by constructing a simple edge exchange based neighborhood on the space of candidate trees. To make the optimization efficient, we initialize our approach using a preliminary tag tree built purely based on semantics from the WordNet hierarchy. The proposed local search based approach is then used to refine the preliminary tag tree based on the corpus statistics.

The evaluation of structures capturing the relationships between different tags or concepts is a difficult task. In the domain of ontologies, there are often no clear quantitative metrics to compare different ontologies that can be built for the same corpus of data. Certain works compare constructed ontologies to a predefined gold standard ontology (Porzel & Malaka, 2004) which is constructed manually. Tag graphs are usually not evaluated explicitly, rather are used in various applications such as tag ranking (Liu et al., 2009). Since manual evaluations are subjective and are not scalable, in this work, we also propose a novel fully automatic framework to evaluate ontological tag trees over tags using the Tag Prediction Accuracy, given an incomplete set of tags for a resource. Furthermore, we also demonstrate that the constructed tag trees can be used to efficiently assign tags to resources in domains where content-based features can be

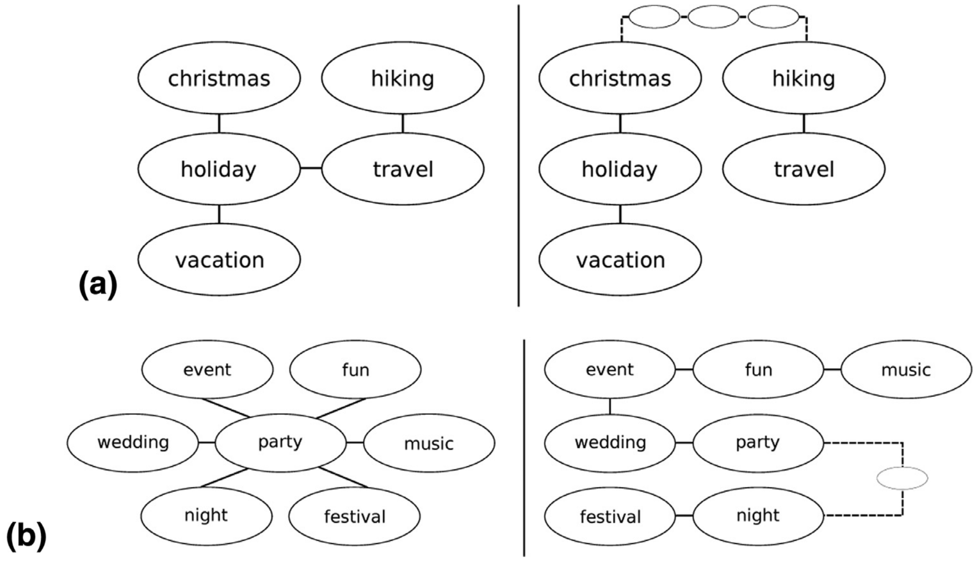


Fig. 2. Two examples of subgraphs built using (left) the proposed data-driven approach and (right) corresponding sub-graphs obtained using WordNet. In example (a), 'holiday' and 'travel' are directly connected using our approach but are separated by multiple hops in the WordNet hierarchy. In example (b), the proposed approach is able to identify 'party' as the central node that connects several other party-related tags. For the proposed approach, objective WAH (3) is utilized as described in Section 3.

derived. Thus as a second evaluation paradigm, we utilize efficiency: for a given resource with no tags, efficiently predict all the applicable tags.

To summarize, the key contributions of this paper are as follows:

(1) We propose a framework to construct an ontological tag tree over tags in a given corpus. The proposed approach requires constructing a preliminary tag tree using semantics obtained from the WordNet hierarchy. This preliminary tree is then refined to incorporate data specific relations by performing a novel local search operating on local neighborhoods in the space of spanning trees of a defined Similarity Graph over the tags.

(2) We propose a completely automated framework for evaluating ontological tag trees over tags by posing two data-driven tasks. The first task is defined such that it is does not require content based features to be extracted from resources, in order to assign tags to them. It can thus be applied even to corpora where deriving features from resources is either infeasible or ineffective. The second task is applicable to corpora where domain specific features can be extracted from the resources and classifiers or concept detectors can be trained that can map the content based features of the resource to concepts or tags.

(3) We evaluate the constructed tag trees for two large image corpora using the above evaluation framework and show that it outperforms tag trees built using manually created semantic hierarchies such as WordNet, and commonly used approaches using tag graphs of comparable space requirements, in both prediction accuracy and efficiency. We also demonstrate that by using the constructed tag trees, we can achieve a performance that is very close to or better than that of other techniques, while also achieving several orders reduction in the space requirement.

Fig. 2 illustrates a couple of examples for which the proposed approach captures the inter-dependencies between tags in a qualitatively better form than the tree obtained using WordNet alone. Details on how these trees are obtained are provided in Section 3. We first start with a brief discussion on the related literature.

1.1. Related work

In order to discuss the related literature, we study the prior works in terms of works on ontology building, deriving tag relationships, tag recommendation and efficient resource classification, and works on local search paradigm. While we have focused on providing a brief summary of works in these areas that are relevant to our paper, some works may belong to multiple areas.

1.1.1. Ontology building

A commonly used strategy to organize a collection of data is to group it into categories and specify the relationships among the various categories. Ontologies (Fensel, 2001) are often employed to specify predefined relations between categories. Conventionally, constructing an ontology (Gruber, 1995) requires significant manual effort. The concepts or categories of the ontology have to be specified, and the relations between the categories defined, all manually. Furthermore, the ontology has to be updated when data belonging to hitherto unseen categories becomes available. Once the ontology has been specified, data samples must be annotated, again manually, to assign them to one or more categories in the ontology so that rules or classifiers can be learned for that category. Therefore, manual techniques for ontological or taxonomic organization of data become especially challenging and cumbersome when there are large amounts of data. Also, ontologies built for one setting are rarely reusable even in other closely related domains. This necessitates the building of an ontology afresh for each new setting. As data could be from one of an ever increasing pool of knowledge domains, manually constructing ontologies for data in each domain is infeasible. Furthermore, when the data obtained is noisy, as is the case for user generated content on the Internet, the problem is accentuated as more manual effort might be needed to clean up the data followed by ontological organization.

The challenges associated with the manual construction of ontologies has led to efforts that use semi-automatic (Jaimes & Smith, 2003) and fully automatic techniques (Buitelaar, Cimiano, & Magnini, 2005) in domains such as multimedia and text based ontologies respectively. Most existing automatic approaches to ontology construction use text mining techniques to identify the concepts and then define relations between the concepts based on their semantic similarity as obtained from lexical databases such as WordNet (Miller, 1995).

Aside of these, other works on ontology building use some form of clustering to combine similar terms or keywords together to form concepts. First, a similarity metric is defined between tags, words or concepts, and then a hierarchical clustering algorithm is utilized to form a dendrogram with the concepts as the leaves of the formed tree. The hierarchical clustering algorithm can be either bottom-up (agglomerative) or top-down (divisive). Such a procedure creates auxiliary concepts in the tree representing combinations of multiple concepts of interest. For example Neshati, Alijamaat, Abolhassani, Rahimi, and Hoseini (2007) use hierarchical clustering based on a compound similarity measure between words. The similarity score is obtained by using a neural network model on syntactical information and corpus based similarities. However such techniques can only group related concepts together at different hierarchical levels, instead of modeling the inter-dependencies in the form of a graph on the concepts. In Dietz, Vandic, and Frasincar (2012), given a corpus corresponding to a domain, the relations between important concepts are learned with the help of WordNet or by using search engine. Hierarchical clustering is employed to construct a domain specific dendrogram as mentioned above.

Works such as Hearst (1992) and Cimiano, Hotho, and Staab (2005) utilize natural language based grammar rules to learn hierarchies between text entities. Semi-automatic techniques for ontology construction such as Text2Onto (Cimiano & Völker, 2005) assist the user in constructing ontologies from a given set of text based data. Similar techniques have been attempted in the domain of annotated multimedia content, such as images and videos (Jaimes & Smith, 2003). Fully automatic techniques such as OntoLearn, etc. Mani, Samuel, Concepcion, and Vogel (2004), Navigli, Velardi, and Gangemi (2003) and Velardi, Navigli, Cuchiarelli, and Neri (2005) use natural language processing and machine learning to extract concepts and relations from data. For a good review of ontology learning from text see Buitelaar et al. (2005). These works cannot be applied outside of the domain of natural language, since they depend at least in part upon grammatical speech.

1.1.2. *Deriving tag relationships*

The organization of tags or concepts obtained from different domains has also been explored. Tag clustering has been employed in systems such as Flickr Clusters (2015) and studies (Begelman, 2006) show that it is helpful as a means to allow users to explore the information space of tags. For annotated images, Schmitz (2006) proposed the application of a co-occurrence based subsumption model from Sanderson and Croft (1999), to learn whether a tag subsumes another. Griffin and Perona (2008) use the category confusion matrix to cluster similar categories together in a hierarchical manner. To construct an ontology for a set of tags, Djuana, Xu, and Li (2011) map the tags to WordNet and leverages WordNet's hierarchy. Tag graphs have been utilized for various applications such as tag ranking (Liu et al., 2009) to represent the pair-wise similarities or distances between tags. While several works use tag graphs as complete graphs on the set of tags, others choose set of edges that have their distance lower than a heuristically chosen threshold (Heymann & Garcia-Molina, 2006). In general, tag graphs have $O(N^2)$ edges with correspondingly large storage requirement for large values of N. For example, Sigurbjörnsson and Van Zwol (2008) estimate the number of tags in Flickr in 2008 to be 3.7 million. Storing each similarity value as a floating point occupying 4 bytes would require more than 27 terabytes to store the pair-wise relationships, as compared to under 50 megabytes as required by the proposed tag tree. This eliminates the need to have computing devices with gigantic memory in order to operate on a large number of tags or concepts for tasks such as tag prediction, resource annotation, etc.

In the domain of annotated images, there exist works that determine semantic relationships between concepts using visual features (Wu, Hua, Yu, Ma, & Li, 2008) and using visual features and tags (Katsurai, Ogawa, & Haseyama, 2014). The approaches proposed in these works are specific to the domain of images and require visual-feature based representation of images. As mentioned in Huang et al. (2010), Song et al. (2010), Zanetti et al. (2008) and Yin et al. (2009), extracting and utilizing content based features can be computationally expensive and even infeasible in certain domains. In addition to above, the space requirement of these works varies as $O(N^2)$ where N is the number of tags or concepts. Our work is different from the above works since we propose a tag tree construction approach which is not dependent on the availability of content-based features from the resources and has a space requirement of only $O(N)$.

1.1.3. *Tag recommendation and efficient resource classification*

While several expert systems such as Anand and Mampilli (2014), Jiang et al. (2015), Kim and Kim (2014), Uddin et al. (2013) and Zheng and Li (2011) address the broader problem of information overload, others focus on addressing the sparsity of online folksonomies through approaches such as tag recommendation and efficient resource classification. In this section we provide a brief summary of the existing literature in the latter category, since such expert systems are closer to our evaluation tasks.

In our first evaluation task, we attempt to predict certain tags associated with a resource while having visibility to other tags associated with the same resource. There exist works in literature that utilize domain-specific features to associate tags to a resource or to determine the relevance of tags to a given resource. For instance, Li, Snoek, and Worring (2009) use visual similarity to determine neighbors of a test images and then aggregates their tags by voting. Wu, Hoi, Jin, Zhu, and Yu (2009) learn a distance metric to determine images that are close to a given image based on the visual content, and then determine tag relevance for the image. Hsieh et al. (2009) build a desktop collaborative tagging system to enable collaborative workers to tag their offline documents. Chen et al. (2015) approach tag recommendation as a translation problem to translate the textual description to tags, while Sun et al. (2011) use language modeling to recommend tags for blogs and documents. These works require extraction of domain-specific, in this case visual and textual features from the resource (images or blogs/documents) and cannot be applied to other domains or corpora where deriving features from resources is either infeasible or ineffective (Huang et al., 2010; Song et al., 2010; Yin et al., 2009; Zanetti et al. (2008). Aside of these works, works such as Uddin et al. (2013) use semantic similarity between tags as obtained from WordNet, to address information overload. As discussed in Section 1, purely semantics based systems fail to capture data-specific characteristics, thus leading to poor performance on data-driven tasks. The tag prediction task as proposed in our paper is defined such that it can even be applied to corpora where deriving content based features from resources is either ineffective or infeasible. The proposed task is somewhat similar to the tag recommendation application in Katsurai et al. (2014) and in Sigurbjörnsson and Van Zwol (2008). However, the procedure for tag recommendation in Katsurai et al. (2014) and Sigurbjörnsson and Van Zwol (2008) requires manual labeling of tags to measure the performance. Since manual labeling is subjective, irregular and not scalable to large numbers of testing resources, we define a tag prediction task where given an incomplete set of tags associated with a resource, we attempt to predict the rest of the tags. Using such an approach, our evaluation is completely automated and does not require any human assistance or labeling. As mentioned above, Katsurai et al. (2014) require visual features to obtain the concept similarities. We compare the performance of the constructed tag trees with the symmetric sum based tag recommendation approach as outlined in Sigurbjörnsson and Van Zwol (2008) and show that we achieve almost similar performance with several orders reduction in the space requirement.

In the second data-driven evaluation task, we utilize tag trees to associate tags to resources based on their content. Specifically, for domains where content-based features can be extracted, we show that using the constructed tag trees, it is possible to determine which concept detectors should be tested for the test resource, thereby making the resource classification more efficient. Compared to Li and Snoek (2013), our approach does not require training of faster and efficient classifiers for this task and can utilize pre-trained binary classifiers as concept detectors for different tags. The image annotation application in Wu et al. (2008) associates a test image with tags based on applying all concept detectors on the test image and using the predicted image likelihood under the Dual Cross-Media Relevance model (Liu et al., 2007). Compared to Wu et al. (2008), our approach does not require applying all concept detectors corresponding to the tags, rather our objective is to determine the concept detectors to apply on a given test image. In order to demonstrate that the proposed approach for the second data-driven task is not applicable to only a single domain, we provide evaluation results based on two types of modalities – visual, and textual. We have used two large sized image corpora to demonstrate the above evaluation tasks. The tasks in Section 4 are defined such that they can be used to evaluate the constructed tag trees. Most of the works as discussed above do not offer a way to do so.

1.1.4. Local search paradigm

The use of local search methods in combinatorial optimization has a long history (Aarts & Lenstra, 1997). The paradigm has been extensively studied (Aarts & Korst, 1988; Johnson, Papadimitriou, & Yannakakis, 1988) due to its practical success on many NP hard problems and also for the insights it provides on the structure of discrete optimization problems. The use of exchange neighborhoods was introduced by Croes (1958) and Lin (1965) for solving the Traveling Salesman Problem and has since been successfully applied to a wide variety of problems. See Aarts and Lenstra (1997) for a comprehensive survey.

We formulate an automated approach for building an ontological tag tree with $(N - 1)$ edges for N tags using WordNet followed by a data-driven refinement. To our knowledge, the formulation of the ontological tag tree construction as an optimization problem on the space of spanning trees and its solution using the 'local search' paradigm is completely novel. We use a variant of the edge-exchange method to construct the neighborhood on the solution space.

Verma et al. (2014) present the preliminary results using the proposed approach. We next discuss the problem statement addressed in this paper, followed by the proposed tree construction approach.

2. Problem statement

We assume that we are given a corpus $\mathscr{C}$ of annotated resources, where each resource is associated with a variable number of tags. The corpus contains:

- Set of resources: $\mathcal{R} = \{r_l\}$ where $l = 1$ to $|\mathscr{C}|$.
- Set of tags in the corpus: $\mathcal{T} = \{t_j\}$ where $j = 1 - N$.
- A binary tag association matrix $\mathcal{B}$: $\mathcal{B}(r,j) = 1$ if tag t_j is associated with resources r, and 0 otherwise.

We define an ontological tag tree as an undirected weighted tree on the set of tags $\mathcal{T}$. This implies that the tag tree is connected and has no simple cycles. The task is to arrange the set of tags $\mathcal{T}$ in an ontological tag tree.

3. Construction of ontological tag tree

In order to construct the tag tree, we propose an approach that starts with a preliminary tag tree obtained using the semantics encoded in WordNet hierarchy. We follow this by a corpus statistics based tag tree refinement.

Construction of the WordNet based preliminary tree is described below.

3.1. Constructing WordNet-based preliminary tag tree

We follow the approach outlined in Djuana et al. (2011) to derive the semantic relations between the set of tags $\mathcal{T}$. This is done in two stages. In the first stage, disambiguation for the meaning of the tags is done by selecting the most popular concept (synonym set, or synset) for every tag. For example a tag 'turkey' can be mapped to the bird, or the country. In WordNet, since the synset corresponding to Turkey, the bird, has a higher frequency count than the synset corresponding to the Republic of Turkey, the former synset would be selected to map to the tag 'turkey'. Then in the second stage, in order to find the relationships between different tags, all links between the mapped concepts are found through the WordNet hierarchy for semantic relationships 'is-a' or 'part-of'. Since we are only interested in a tag tree which has undirected edges between the tags, we ignore the directions of the edges in the obtained hierarchy, which could otherwise help distinguish more generic concepts or tags from more specific ones.

The resulting undirected graph in general has cycles and is usually disconnected, forming disjoint clusters of tags. In order to construct a tree from the above undirected graph, we first break the cycles and then connect disjoint segments in a greedy manner using inter-tag semantic distances as obtained from WordNet Library (Rita WordNet Library, 2015). The semantic distance between two synsets in WordNet is defined as:

$$\frac{\mathrm{MHP}}{\mathrm{HPR} + \mathrm{MHP}} \qquad (1)$$

where MHP $=$ minimum hops to common parent, and HPR $=$ hops from common parent to Root of hierarchy. We define the semantic distance between tags i and j as the semantic distance between the respective WordNet synsets. The steps of the procedure for obtaining a preliminary ontological tag tree starting from WordNet hierarchy are summarized in Algorithm 1, which returns a tree over the set of tags $\mathcal{T}$.

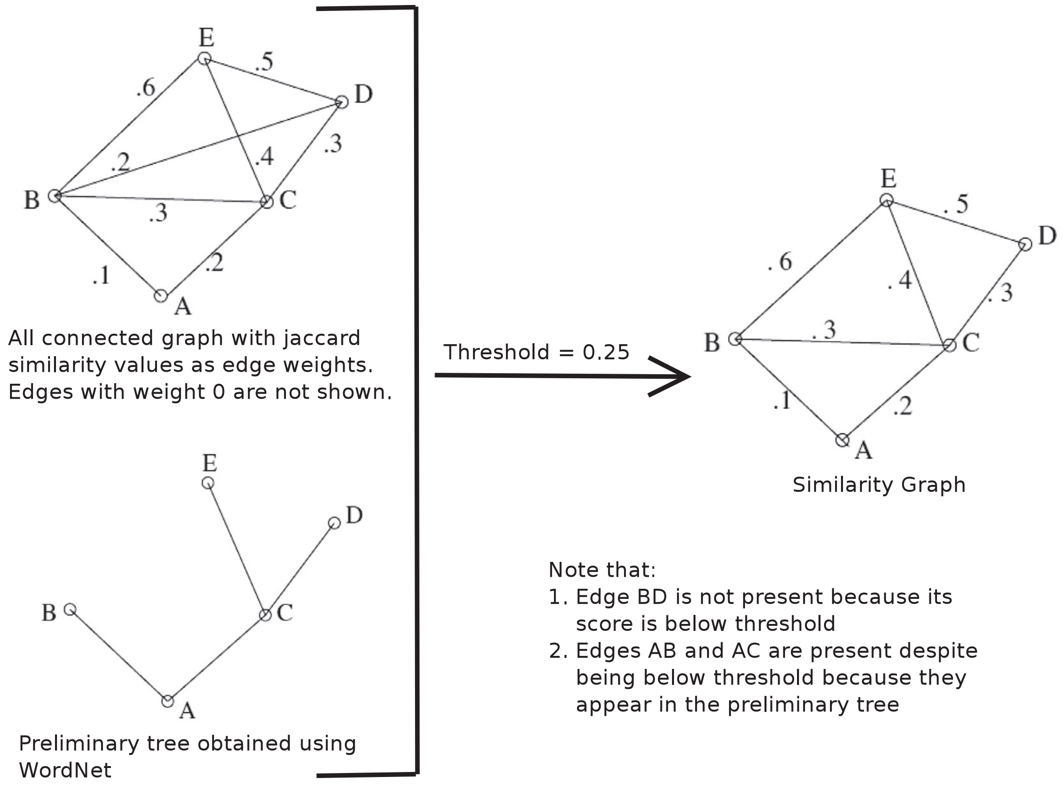
Algorithm 1
Constructing preliminary tag tree using WordNet hierarchy.

```
Input:
• WordNet based hierarchy $\mathcal{H}_\mathcal{T}$ for given set of tags $\mathcal{T}$ representing
semantic relationships such as 'is-a' or 'part-of'. $\mathcal{H}_\mathcal{T}$ is directed graph and
can be disconnected.
• Pair-wise semantic distances for tags in $\mathcal{T}$ from (1)
Breaking cycles:
Obtain $H_{undirected}$ as the undirected version of $\mathcal{H}_\mathcal{T}$
Loop While cycles exist in $H_{undirected}$
    • Find one cycle in $H_{undirected}$. $E_{cycle}$ = edges in the obtained cycle
    • Remove the edge $e \in E_{cycle}$ with largest distance (1) from $H_{undirected}$
EndWhile
Connecting disjoint components in $H_{undirected}$:
Loop While $H_{undirected}$ is disconnected
    • Obtain pair-wise distances between tags from (1)
    • Set distances between tags in same component, to $\infty$
    • Connect tags with least distance. These will form edges between the
disjoint components.
EndWhile. Output: tree $T_W$
```

Once the preliminary tag tree based on WordNet, T_W, is constructed as described, we refine it using a data driven approach based on the co-occurrence statistics of the tags. We describe this below.

3.2. Co-occurrence based tag tree refinement

The preliminary tag tree is refined by accounting for those tags that strongly co-occur in the corpus but are not linked in the WordNet based tag tree. To achieve this, we first define the similarity score of two tags using Jaccard similarity:


E
.5
.6
D
.2
.4
.3
B
.3
C
.1
.2
A
All connected graph with jaccard similarity values as edge weights. Edges with weight 0 are not shown.
Threshold = 0.25
E
. 5
D
. 6
. 4
. 3
B
. 3
C
.1
.2
A
Similarity Graph
E
D
B
C
A
Preliminary tree obtained using WordNet

Note that:
1. Edge BD is not present because its score is below threshold
2. Edges AB and AC are present despite being below threshold because they appear in the preliminary tree

Fig. 3. Building the Similarity Graph for threshold $\tau = 0.25$.

Definition 1. [Jaccard similarity]. The Jaccard similarity $J(s_1,s_2)$ between two sets s_1,s_2 is defined to be equal to

$$\frac{|s_1 \cap s_2|}{|s_1 \cup s_2|}. \tag{2}$$

Let $\mathcal{T} = \{t_j\}_{j=1}^{N}$ be the set of tags and $\mathcal{R} = \{r_l\}_{l=1}^{|\mathscr{C}|}$ be the set of resources. For each tag t_j we construct the set $s_j = \{r: r \in \mathcal{R}$ and $\mathcal{B}(r,j) = 1\}$. Let $J_\mathcal{T}$ be an $N \times N$ matrix such that $J_\mathcal{T}(i, j) = J(s_i,s_j)$. We will call $J_\mathcal{T}$ the *Jaccard Matrix* of $\mathcal{T}$.

We augment the preliminary tag tree based on WordNet, T_W, to construct a *Similarity Graph* as follows. We construct the graph with vertex set $V = \{v_1,\ldots,v_N\}$ where the v_j corresponds to the tag t_j. We start with T_W, which is a tree on V constructed using WordNet as described in Algorithm 1. Additionally, given a threshold τ, $0 \le \tau \le 1$, we join v_i,v_j with an edge if $J_\mathcal{T}(i, j) \ge \tau$, and the edge weight of edge (v_i,v_j) is set to be $J_\mathcal{T}(i, j)$. We call the resulting graph $\mathcal{G}_\mathcal{T}$, the *Similarity Graph* of $\mathcal{T}$ since it captures the tree based on semantic similarity, and additional edges based on corpus based Jaccard similarity. Fig. 3 shows an illustrative example for obtaining Similarity Graph based on a threshold τ.

Given the Similarity Graph $\mathcal{G}_\mathcal{T}$, the objective of the refinement stage is to find a tree in the space of spanning trees of the Similarity Graph $\mathcal{G}_\mathcal{T}$ which minimizes a defined objective function. Below we define and motivate two different objective functions based on corpus statistics, for tag tree construction.

(1) Weighted Average Hops (WAH)

$$\sum_i \sum_{j,j<i} J_\mathcal{T}(i, j) d_{i,j}, \tag{3}$$

where $d_{i,j}$ represents the number of hops between tag t_i and tag t_j in the tag tree. The motivation for such a score is that it is lower when tags *i* and *j* with high $J_\mathcal{T}(i, j)$ are separated by fewer hops as compared to tags with low $J_\mathcal{T}(i, j)$. The above objective function is equal to the sum of the pair-wise hops between all pairs of tags weighted by the corresponding Jaccard similarity $J_\mathcal{T}(i, j)$. Dividing the sum by $\Sigma_{i,j<i} J_\mathcal{T}(i, j)$ would be equal to the weighted average number of pair-wise hops where the weights are normalized Jaccard similarities. Since the value $\Sigma_{i,j<i} J_\mathcal{T}(i, j)$ is a constant for a given set of tags $\mathcal{T}$, we have removed the scaling factor from the objective function. For a general graph $\mathcal{G}_\mathcal{T}$, the problem of minimizing the weighted average number of hops has been established to be an NP hard problem (Garey & Johnson, 1979).

(2) Similarity Approximation (SA)

$$\sum_i \sum_{j,j<i} w_{i,j} |J_\mathcal{T}(i, j) - S_T(i, j)|, \tag{4}$$

where $S_T(i, j)$ represents the similarity between tags t_i and t_j estimated using tag tree *T*. A very close problem is that of approximating a given distance matrix through spanning trees, which has been established to be NP hard (Eckhardt, Kosub, Maa, Täubig, & Wernicke, (2005). The objective function in (4) is the weighted L1 norm of the difference between the Jaccard Matrix $J_\mathcal{T}$ and the *Estimated Similarity Matrix* S_T. The weights $w_{i,j}$ are taken to be the co-occurrence counts of tags t_i and t_j and are useful to establish relative importance between different pairs of tags in the objective function. While $S_T(i, j)$ can be calculated in several ways for a given tag tree *T*, we define $S_T(i, j)$ as

$$S_T(i, j) = \prod_{e \in \mathfrak{B}_{i,j}} S(e), \tag{5}$$

where $\mathfrak{B}_{i,j}$ is the path in tag tree *T* connecting tags t_i and t_j and $S(e)$ is equal to the Jaccard similarity between the tags that edge e connects. Such a definition for $S_T(i, j)$ ensures that it lies between 0 and 1 and no rescaling is required in order to compare $S_T(i, j)$ values with $J_\mathcal{T}(i, j)$.

Note that the trees in (3) and (4) are constrained to be spanning trees over the Similarity Graph $\mathcal{G}_\mathcal{T}$. The local search based approach to minimize either of the above objective functions is described next.

3.3. Optimization based on local search

Given the Similarity Graph $\mathcal{G}_\mathcal{T}$ for a set of tags $\mathcal{T}$, our objective is to construct a spanning tree on $\mathcal{G}_\mathcal{T}$ such that the defined objective function is minimized. Since finding spanning trees of $\mathcal{G}_\mathcal{T}$ that optimally minimize either (3) or (4) is a hard problem, we propose an approach to obtain local optimum through the local search paradigm.

Local search: Local search algorithms provide a local optimum to an optimization problem. This is done by moving from one solution to another, in the search space of candidate solutions.

For the problem of constructing an ontological tag tree, we define a simple edge-exchange based neighborhood on the space of spanning trees of the graph $\mathcal{G}_\mathcal{T}$ as follows. Given two spanning trees T_1, T_2 we say that T_2 is a neighbor of T_1 if it can be obtained from T_1 by the following process:

(1) Pick an edge $e_1 \in \mathcal{G}_\mathcal{T} \backslash T_1$ and add it to T_1.
(2) In the (unique) cycle thus formed in T_1 containing e_1 pick the edge, say e_2 with minimum weight (i.e., Jaccard similarity of the tags e_2 connects). Remove e_2 from T_1.

Starting from a spanning tree T_0 of $\mathcal{G}_\mathcal{T}$ as an initial solution, we explore all neighbors of T_0 to determine which neighbor minimizes the defined objective function. The winning neighbor is then considered as the next solution and its neighbors are explored until no further benefit is seen in the objective function. The steps of the local search based ontological tree construction are listed in Algorithm 2. The output is the locally optimal tag tree T_{opt}. Note that T_0 is taken as T_W as obtained from Algorithm 1. Fig. 4 shows one iteration of the proposed local search based approach for the objective function in (3).

Algorithm 2
Ontological tree construction algorithm.

```
Input:
• Similarity Graph $\mathcal{G}_\mathcal{T}$ for a given set of tags $\mathcal{T}$
• Initial Solution: $T_0$.
• Pair-wise Jaccard similarities between tags, i.e., $J_\mathcal{T}$ (i,j) Initialization:
$S = T_0$
Loop:
  • $E_{Candidates}$: = set of edges present in $\mathcal{G}_\mathcal{T}$ and not in $S$. For each edge $e$
in $E_{Candidates}$
      • Add edge $e$ in $S$ to get graph $G$.
      • $E_{Cycles}$: = set of edges in the cycle formed in $G$.
      • Remove edge $e'$ with lowest weight (i.e., Jaccard
      similarity of connecting tags) from $E_{Cycles}$:$e' \neq e$
      • $S_{Neighbor}$ = spanning tree thus formed
      • Calculate objective function at $S_{Neighbor}$
  EndFor
  • Select neighbor giving best objective function as $S_{Next}$
  • If $S_{Next}$ improves objective function over $S$
      $S = S_{Next}$
  • Else Stop iterating
End Loop. Output: locally optimal spanning tree $T_{opt} = S$
```

3.4. Effect of initializing using WordNet

The benefit of using WordNet to initialize the proposed local search based approach is that as compared to random initializations, the former helps achieve a preferred value of the objective function faster. The typical way to attain a lower value of the objective function for an optimization problem as defined in Section 3.2 would be to run the local search using randomly constructed tree on the set of tags as initialization, and picking the best tree across several such runs based on the tag tree that has lowest objective function. However this requires running the local search several times which can be large considering that the number of spanning trees on a set of N tags varies as N^{N-2} (Cayley, 1894). The preliminary tree as constructed using WordNet offers a more meaningful initialization to the local search by capturing certain types of relationships between the tags that are dictated by their semantics. Table 1 provides the statistics of the objective function of tag trees constructed by running the proposed local search based approach 20 times with random initializations. As can be seen, a single run using WordNet based preliminary tag tree leads to a much better objective function (4). Also, the resulting tag tree using WordNet leads to a better performance than the best across 20 runs with random initializations. Note that for Table 1, the performance of tag trees is measured using Average Tag Prediction Accuracy as discussed in Section 4.

4. Evaluation

In this section, we describe the experimental setup for the construction and evaluation of ontological tag trees. For the evaluation, we define tag prediction and efficient classification tasks as discussed later in this section. The experiments are conducted on two large corpora of images, the details of which are given below.

4.1. Datasets

To test the robustness of our approach to build ontological tag trees for domains with varying degrees of tag noise, we use two different image corpora – one from Flickr, composed primarily of user generated content, and one from a professionally curated stock photo agency.

- *Flickr images:* Flickr (2015) is a popular image and video hosting website where users can upload images and associate them with annotations such as titles, tags and descriptions, among others. As Flickr primarily contains user generated content, tags are often noisy, irrelevant to image content or even completely absent. We utilize 500,000 images for training and 100,000 images for testing. All these images are licensed under Creative Commons copyright licenses.
- *Stock images corpus:* To evaluate the proposed approach on less noisy data, we take a corpus of stock photos that are professionally annotated, and hence are accompanied with a variety of accurate annotations - such as keywords, captions, etc. For this corpus, we use the set of keywords to build the ontological tag tree, and refer to them as 'tags'. We utilize more than 350,000 images for training and close to 70,000 images for testing. The textual captions are used in the efficient classification task as shown in Section 4.4.

Training images are used for adapting the WordNet based preliminary tag tree obtained using Algorithm 1 to the given corpus using the local search based approach described in Algorithm 2. Training images are also used for specific required tasks such as training of classifiers. Testing images are used to evaluate the constructed tag trees. There is no overlap between training and test sets.

4.2. Effect of local search based optimization

We first demonstrate how the proposed local search based approach helps in improving the objective function as defined in Section 3. Fig. 5 shows the variation of the objective function in (3) with the number of iterations on Flickr tag corpus. The objective function value of the WordNet based preliminary tag tree for Flickr corpus before the proposed refinement is 357.2 that becomes 167.1 using the local search based refinement in 68 iterations. Median of the Jaccard similarity values is taken as the threshold τ for candidate selection in the proposed refinement algorithm. Similar improvement is observed for the Stock images corpus, for which the objective function values

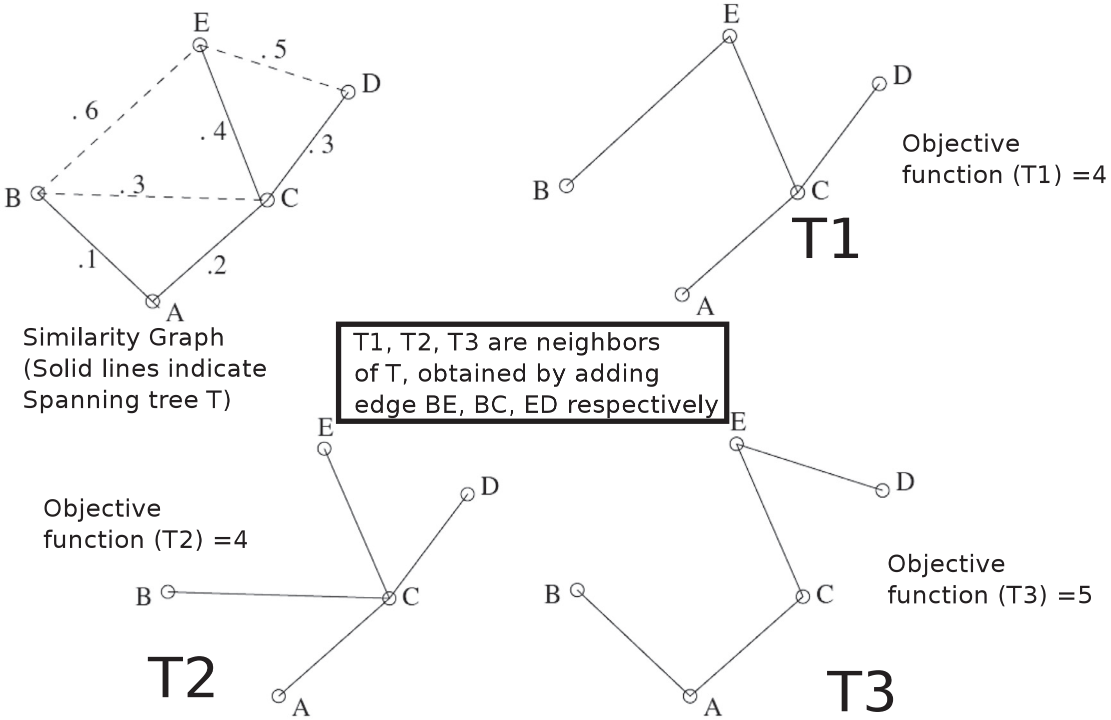


Fig. 4. One iteration of the proposed approach. Eq. (3) is utilized to calculate the objective function for neighbors of a tree T. Ties are broken arbitrarily. Since both $T1$ and $T2$ have an objective function (3) value of 4, we choose $T1$ and proceed to next iteration.

Table 1
Effect of initializing the proposed local search based approach using WordNet. A total of 30 tags from stock image corpus (described in Section 4) are used.

	Random initialization			Using WordNet
	Min	Mean	Max	
Objective function ((4) × 10^6)	6.4	13.7	47.7	0.6
Performance (%)	24.2	43.8	49.2	50.9

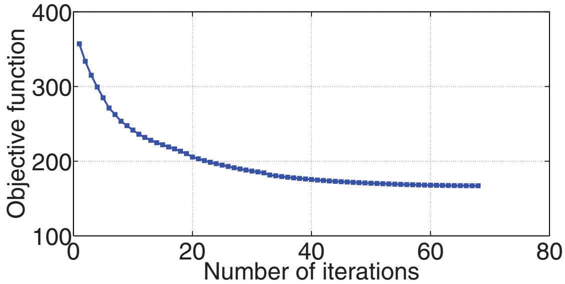


Fig. 5. The variation of the objective function (3) with the number of iterations for a sample run on Flickr tag corpus.

improves from 224.5 to 99 after 23 iterations. For both corpora, the iterations are terminated once no further improvement is observed in the objective function. We describe below the tasks defined to evaluate the constructed tag trees.

4.3. Tag prediction task

The tag prediction task is similar to the tag recommendation task in Katsurai et al. (2014) and in Sigurbjörnsson and Van Zwol (2008). However as outlined in Section 1.1, the proposed task does not require manual labeling for the resources (images) to evaluate the proposed tag tree construction approach. In order to do so, we divide the tags associated with a resource into a seen and an unseen set of tags and use the latter to evaluate the predicted tags. We demonstrate this approach through experiments conducted on image corpora. Let an image i in the corpus be tagged with the set of tags $\mathcal{T}_i$, such that $|\mathcal{T}_i| = N_{Tags}$. Assume that out of these N_{Tags} tags, only a subset $\mathcal{T}_{i,Seen}$ are observed, with $|\mathcal{T}_{i,Seen}| = N_{Seen}$. The objective of the tag prediction task is to predict the remaining $(N_{Tags} - N_{Seen})$ tags, i.e., $\mathcal{T}_i \backslash \mathcal{T}_{i,Seen}$. Let $\mathbb{P}_i$ be the set of $(N_{Tags} - N_{Seen})$ tags predicted for image i assuming that $\mathcal{T}_{i,Seen}$ is observed. Note that the prediction assumes the total number of tags for the image, N_{Tags}, to be known. Performance of tag prediction can be measured by the *Tag Prediction Accuracy*, defined as follows:

$$\text{Tag Prediction Accuracy} = \frac{|\{\mathcal{T}_i \backslash \mathcal{T}_{i,Seen}\} \cap P_i|}{|\{\mathcal{T}_i \backslash \mathcal{T}_{i,Seen}\}|} \tag{6}$$

We now discuss the approach we follow to obtain the set of predicted tags $\mathbb{P}_i$ when the set of tags $\mathcal{T}_{i,Seen}$ is seen, by utilizing a given ontological tag tree.

4.3.1. Utilizing ontological tag tree for tag prediction

Consider the tag tree T, built over the set of $\mathcal{T}$ tags in a corpus. For image i with N_{Seen} number of seen tags, each tag $t \in \{\mathcal{T} \backslash \mathcal{T}_{i,Seen}\}$ is given a proximity score s_t based on its proximity from the seen tags, as per T. Specifically,

$$s_t = \sum_{t' \in \mathcal{T}_{i,Seen}} dist(t, t'), \tag{7}$$

where $dist(t,t')$ is the distance between tags t and t' in T calculated as shown in Section 4.3.2. A lower proximity score for a tag t indicates that it is closer in a cumulative sense to the set of observed tags $\mathcal{T}_{i,Seen}$. The tags are ordered in the increasing order of s_t, and the first $(N_{Tags} - N_{Seen})$ tags, i.e. those corresponding to the least values of s_t, are chosen as the set of predicted tags $\mathbb{P}_i$.

4.3.2. Methods compared

We compare the following methods in the tag prediction task:

(1) *Random:* As the name suggests, this baseline method randomly picks $(N_{Tags} - N_{Seen})$ tags from the set $\mathcal{T} \backslash \mathcal{T}_{i,Seen}$.

(2) *WordNet:* This baseline approach uses the semantics based ontological tag tree constructed from WordNet hierarchy using the procedure described in Algorithm 1. The edge weights are assigned to be semantic distances as obtained from Rita WordNet Library (2015).

(3) *Google Similarity Distance:* Google Similarity Distance (Cilibrasi & Vitanyi, 2007) has been used to construct tag graphs in applications such as tag ranking (Liu et al., 2009). As mentioned in Section 1.1, a threshold is used to discard certain edges in tag graphs. We choose a threshold such that for a tag graph with N nodes (or tags), there are exactly $(N - 1)$ edges remaining, so that the tag graph thus formed has same number of edges and space requirement as the tag tree learnt from proposed approach. Edge weights for the tag graph are taken to be the Google Similarity Distance as defined by Cilibrasi and Vitanyi (2007).

(4) *LS Weighted Average Hops (LS-WAH):* Here we construct a tag tree using the proposed local search based approach, to minimize Weighted Average Hops (3) in Section 3.2. If an edge exists between tags t_i and t_j then the weight of the edge connecting them is given to be $(1 - J_{\mathcal{T}}(i, j))$.

(5) *LS Similarity Approximation (LS-SA):* This tag tree is constructed using the proposed approach with objective corresponding to Similarity Approximation as outlined in (4) in Section 3.2. The edge weights are assigned as in method 4 above.

(6) *Symmetric sum based:* In order to compare the performance of the proposed tag tree construction approaches with that of other tag recommendation approaches that do not use tag trees or tag graphs or any visual features, we also provide the performance of the symmetric sum based approach as proposed in Sigurbjörnsson and Van Zwol (2008). Note that the space required to store the pair-wise similarities in Sigurbjörnsson and Van Zwol (2008) is $O(N^2)$ while the proposed tag tree construction requires $O(N)$ space to store the tag tree.

The prediction task is performed using the approach described in Section 4.3.1. For methods numbered 2, 3, and 4 above, $dist(t_i,t_j)$ as required in (7) is calculated by adding distances of edges in path connecting tags t_i and t_j. For LS-SA method, $dist(t_i,t_j)$ is defined as $\{1 - S_T(i, j)\}$ where $S_T(i, j)$ is calculated for an ontological tag tree T as per (5). This is because the tag tree construction approach for LS-SA method utilizes product based Similarity Approximation (5) and so it is appropriate to use same approach to estimate similarities based on a tree, and hence to calculate $dist(t_i,t_j)$. Similarly, for Symmetric sum based method (Sigurbjörnsson & Van Zwol, 2008), $dist(t_i,t_j)$ is defined as $\{1 - J_{\mathcal{T}}(i, j)\}$ where $J_{\mathcal{T}}(i, j)$ is the Jaccard similarity between tag t_i and t_j as defined in Section 3.2. Note that this makes (7) similar to the sum based scoring approach in Sigurbjörnsson and Van Zwol (2008).

Flickr corpus: We begin by choosing the top 150 most popular tags in a sample of Flickr images. After selecting only those tags that also occur in the WordNet database, we are left with 117 tags. These comprise the set $\mathcal{T}$. The total number of tags in an image, varies from 0 to 6 for most Flickr images. Since we need at least one tag to be seen and at least one to be predicted, we vary N_{Tags} from 2 to 6. For each value of N_{Tags}, test images are selected which contain exactly N_{Tags} tags. For each such image i, all combinations of N_{Seen} tags are selected to comprise the observed tag set $\mathcal{T}_{i,Seen}$. Predictions are made as discussed in Section 4.3.1 and performance of tag prediction task is measured using (6). N_{Seen} is varied from 1 through $(N_{Tags} - 1)$. Given values for N_{Tags} and N_{Seen}, the Average Tag Prediction Accuracy is the Tag Prediction Accuracy using (6) averaged across test images. We define Overall Tag Prediction Accuracy as the mean of Average Tag Prediction Accuracy across all combinations of N_{Tags} and N_{Seen}.

Table 2
Overall Tag Prediction Accuracy (%) for various methods for tag prediction task on Flickr and Stock images corpora.

Tag prediction method	Flickr corpus	Stock images corpus
Random tag prediction	2.29	13.21
WordNet	7.95	22.67
Google Similarity Distance	34.02	40.05
LS-WAH	31.53	48.06
LS-SA	44.52	50.86
Symmetric sum based	47.13	54.71

Table 3
Average Tag Prediction Accuracies (in %) obtained using Random method on Flickr corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5
2	1.57	–	–	–	–
3	2.34	1.30	–	–	–
4	2.67	1.55	0.73	–	–
5	2.19	1.68	1.01	0.78	–
6	6.80	5.39	3.36	2.41	0.58

Table 4
Average Tag Prediction Accuracies (in %) obtained using WordNet method on Flickr corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5
2	7.05	–	–	–	–
3	8.17	2.88	–	–	–
4	11.16	5.27	5.39	–	–
5	16.50	10.05	13.10	0.84	–
6	14.72	10.85	9.46	2.97	0.87

Table 5
Average Tag Prediction Accuracies (in %) obtained using Google Similarity Distance method on Flickr corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5
2	22.07	–	–	–	–
3	17.01	5.50	–	–	–
4	23.70	16.67	11.66	–	–
5	51.51	47.09	45.28	43.15	–
6	54.55	48.28	44.68	41.85	37.30

Table 6
Average Tag Prediction Accuracies (in %) obtained using LS-WAH method on Flickr corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5
2	16.13	–	–	–	–
3	11.53	5.51	–	–	–
4	14.91	10.96	7.17	–	–
5	42.76	42.94	39.69	36.95	–
6	49.37	51.48	49.00	48.89	45.69

The Overall Tag Prediction Accuracies for the various methods compared are shown in Table 2. Tables 3–8 show the Average Tag Prediction Accuracies for various combinations of N_{Tags} and N_{Seen} for the individual methods discussed in Section 4.3.2 for Flickr corpus. For comparison, the Average Tag Prediction Accuracies marginalized over N_{Tags} are shown in Fig. 6. Note that Google Distance refers to the method corresponding to Google Similarity Distance as outlined above. It can be seen that the LS-SA method outperforms all others and has its performance very close to that of the Symmetric sum based approach (Sigurbjörnsson & Van Zwol, 2008). It should be noted that the latter utilizes pair-wise similarities between tags as derived from the corpus and thus has space requirement of $O(N^2)$ which as discussed in Section 1.1 is very high for large number of tags, i.e., for large N. Compared to this, the proposed approach has

Table 7
Average Tag Prediction Accuracies (in %) obtained using LS-SA method on Flickr corpus.

$N_{\mathrm{Seen}} \rightarrow N_{\mathrm{Tags}} \downarrow$	1	2	3	4	5
2	25.87	–	–	–	–
3	20.34	21.65	–	–	–
4	26.09	28.96	26.43	–	–
5	52.33	54.79	54.82	53.16	–
6	59.57	61.78	62.10	60.81	59.07

Table 8
Average Tag Prediction Accuracies (in %) obtained using Symmetric sum method on Flickr corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5
2	25.62	–	–	–	–
3	21.19	22.17	–	–	–
4	28.69	30.88	28.12	–	–
5	55.6	57.88	57.12	54.27	–
6	62.77	65.48	67.05	66.35	63.75

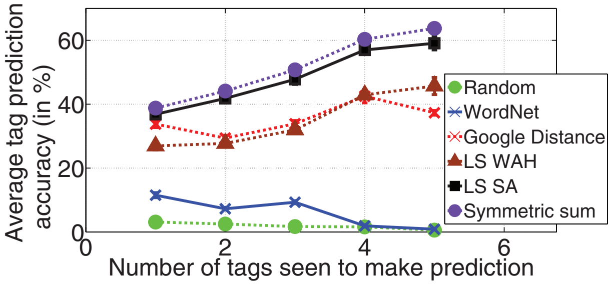


Fig. 6. Average Tag Prediction Accuracies marginalized over N_{Tags} for various methods for the tag prediction task on Flickr corpus. Note that Google Distance refers to the method corresponding to Google Similarity Distance as outlined in Section 4.3.2.

Table 9
Average Tag Prediction Accuracies (in %) obtained using Random method on Stock images corpus.

$N_{Seen} \rightarrow N_{\mathrm{Tags}} \downarrow$	1	2	3	4	5	6	7	8	9
2	1	–	–	–	–	–	–	–	–
3	7	5.28	–	–	–	–	–	–	–
4	11.3	5.8	4.9	–	–	–	–	–	–
5	14.9	10.1	6.8	6.1	–	–	–	–	–
6	19.1	12.9	10.1	9.5	3.1	–	–	–	–
7	21.5	16.1	14.6	10.1	13.1	2.5	–	–	–
8	20.7	17.8	11.8	12.7	6.4	9.6	2.3	–	–
9	28.6	25.5	23	19.8	19.4	12.4	7	5.7	–
10	28.3	27.6	23.5	23.5	17.7	13.6	16.7	7.6	7.4

only $O(N)$ space requirement. We will discuss in Section 6 the reason why LS-SA leads to construction of a tree that outperforms the LS-WAH method. Google distance based method has performance close to that of LS-WAH while the tag tree based on WordNet hierarchy does not work very well for tag prediction task. As expected, random tag prediction performs the worst.

Stock images corpus: As in the Flickr corpus, we first select the set of most popular tags from the corpus of stock images that are also in WordNet. This produces a set $\mathcal{T}$, of 30 tags. Tables 9–14 show the Average Tag Prediction Accuracies for various combinations of N_{Tags} and N_{Seen} for the tag prediction methods discussed in Section 4.3.2. The Average Tag Prediction Accuracies marginalized over N_{Tags} are shown in Fig. 7, plotted as a function of N_{Seen}. Note that if N_{Tags} is kept constant, increasing N_{Seen} reduces the number of unseen tags (i.e., $\mathcal{T}_i \backslash \mathcal{T}_{i,Seen}$), thus reducing the random chance of predicting a correct unseen tag from $\mathcal{T} \backslash \mathcal{T}_{i,Seen}$. As a result of this phenomenon, a drop

Table 10
Average Tag Prediction Accuracies (in %) obtained using WordNet method on Stock images corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5	6	7	8	9
2	1.9	–	–	–	–	–	–	–	–
3	8.9	3.3	–	–	–	–	–	–	–
4	14.2	10.9	3.4	–	–	–	–	–	–
5	21.3	16.9	14.7	8.5	–	–	–	–	–
6	28.4	22.8	22	16.7	9.8	–	–	–	–
7	32.3	28.4	28.4	24.8	22.4	12.6	–	–	–
8	35.7	33.3	32.7	30.7	27.6	23.8	13	–	–
9	36.5	34.5	32.7	31.9	28.7	26.3	23.4	15.5	–
10	37.4	35.2	32.9	30.8	28	26.1	22.3	18.5	9.9

Table 11
Average Tag Prediction Accuracies (in %) obtained using Google Similarity Distance method on Stock images corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5	6	7	8	9
2	3.5	–	–	–	–	–	–	–	–
3	11.2	8.4	–	–	–	–	–	–	–
4	24	26.6	21.9	–	–	–	–	–	–
5	32.3	39.8	39.7	35.2	–	–	–	–	–
6	36.6	43.9	46.5	46.5	44.5	–	–	–	–
7	39.4	44.8	47.8	49.5	49.5	48.5	–	–	–
8	43.6	45.2	47.3	48.5	49.1	49	48.7	–	–
9	41.9	43.5	44.4	45.4	45.9	45.8	45.2	44.9	–
10	41.6	42.7	43.2	42.8	42.6	42	40.7	39.5	38.7

Table 12
Average Tag Prediction Accuracies (in %) obtained using LS-WAH method on Stock images corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5	6	7	8	9
2	7.8	–	–	–	–	–	–	–	–
3	14.6	15.4	–	–	–	–	–	–	–
4	20.5	22.6	22	–	–	–	–	–	–
5	36.2	33.2	32.8	31.3	–	–	–	–	–
6	49	47.6	45.2	43.3	41.5	–	–	–	–
7	59.6	59.4	58.7	57.1	54.4	51.4	–	–	–
8	62.3	63.5	63.2	62.8	61.4	58.2	54.9	–	–
9	56.6	59.7	59.2	58.7	57.9	56.3	52.8	49.5	–
10	53.5	57.6	57.4	56.8	56.1	54.7	52.9	48.5	44.4

Table 13
Average Tag Prediction Accuracies (in %) obtained using LS-SA method on Stock images corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5	6	7	8	9
2	7.2	–	–	–	–	–	–	–	–
3	13.38	11.1	–	–	–	–	–	–	–
4	28.4	27	21.1	–	–	–	–	–	–
5	44.5	41.9	38.5	35	–	–	–	–	–
6	53.3	53.3	51.3	49.5	46.9	–	–	–	–
7	61.7	63.2	62.6	60.8	58.3	56	–	–	–
8	64.3	64.8	65.8	65.3	63.1	60.2	57.4	–	–
9	59	60.2	60.8	61.1	60.8	58	55.3	52.3	–
10	55.7	57.7	57.9	57.8	57.6	57	53.7	50.6	47.1

Table 14
Average Tag Prediction Accuracies (in %) obtained using the Symmetric sum approach on Stock images corpus.

$N_{Seen} \rightarrow N_{Tags} \downarrow$	1	2	3	4	5	6	7	8	9
2	11.1	–	–	–	–	–	–	–	–
3	16.9	17.7	–	–	–	–	–	–	–
4	32.1	32.5	29.7	–	–	–	–	–	–
5	45.1	47.8	46.4	43.1	–	–	–	–	–
6	52.7	56	56.3	55.5	53.2	–	–	–	–
7	62.1	64.6	64.9	63.9	62	59.9	–	–	–
8	64.5	68	68.1	67.8	65.8	63.1	60.5	–	–
9	60.6	63.9	64.9	64.9	64	61.5	58.5	55.9	–
10	58	61.5	62.9	63.2	63.1	61.6	58.9	55	52.3

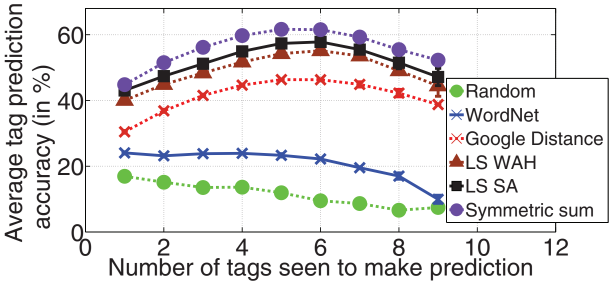


Fig. 7. Average Tag Prediction Accuracies marginalized over N_{Tags} for various methods for the tag prediction task on Stock images corpus.

in performance for random prediction and other methods can be seen with increasing N_{Seen}.

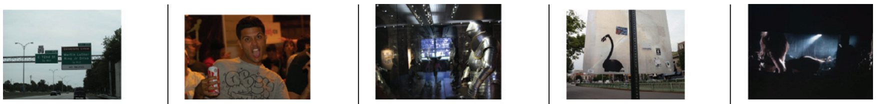
We provide examples of some test images from the Flickr corpus that had $N_{Tags} = 5$ and $N_{Seen} = 2$. Fig. 8 shows a few test images for which LS-SA method made 100% correct predictions. Fig. 9 shows test images that gave 0% Tag Prediction Accuracies. The corresponding sets of see ($\mathcal{T}_{i,Seen}$), unseen ($\mathcal{T}_i \backslash \mathcal{T}_{i,Seen}$), and the predicted tags ($\mathbb{P}_i$) are also provided.

The results indicate that the proposed local search paradigm based approach has successfully adapted the ontological tag tree obtained from WordNet to the Flickr or stock images corpus. We can obtain performance better than or close to that of other tag prediction approaches, while having several orders of savings in the space requirement. We next describe the second data-driven task used for evaluation of tag trees.

4.4. Efficient classification task

We consider the problem of efficiently associating tags with resources based on the resource content. For domains where it is feasible to extract features from the resource and to train concept detectors, we show how tag trees can be utilized to determine which concept detectors should be applied on a given test resource. We demonstrate this using annotated image corpora and utilize different modalities to represent the content of a given resource. Given a test image i from the corpus without any associated tags or keywords, we predict which of the N tags are applicable to the image. By letting each of the N tags correspond to a category or class, this is equivalent to a multi-class, multi-label classification task. Let us assume that one-vs-all binary classifiers are available for each tag class – these take the image instance i as input and can predict $P(j|i)$ where class j corresponds to tag t_j. We can use this to predict whether or not a tag t_j should be associated with image i based on whether $P(j|i)$ is greater than an appropriate threshold θ or not.

The naive approach to predict all tags applicable to i would be to test each of the N classifiers on i and accumulate those tags for



(a) Seen: highway, road **Unseen**: route, shield, sign

(b) Seen: austin, band **Unseen**: music, texas, tx

(c) Seen: england, europe **Unseen**: london, travel, uk

(d) Seen: art, dc **Unseen**: graffiti, street, washington

(e) Seen: concert, england **Unseen**: london, music, uk

Fig. 8. Example images where LS-SA method gave 100% Tag Prediction Accuracy when the first two tags were seen and the next three were unseen and predicted. The Flickr owner and photo ids of these images are (dougtone@7975042008), (elchupacabra@7023118527), (jeffwilcox@159730021), (daquellamanera@4678084101), and (martinrp@3832812191), respectively.

(a) Seen: film, france **Unseen**: holiday, sky, snow **Predicted**: paris, europe, bw

(b) Seen: china, family **Unseen**: live, summer, usa **Predicted**: photography, christmas, photo

(c) Seen: canada, ocean **Unseen**: red, sky, sunset **Predicted**: sea, beach, water

(d) Seen: autumn, black **Unseen**: light, macro, night **Predicted**: white, nature, bw

(e) Seen: car, green **Unseen**: photo, washington, white **Predicted**: red, spring, nature

Fig. 9. Example images where LS-SA method gave 0% Tag Prediction Accuracy when the first two tags were seen and the next three were unseen and predicted. Also provided are the tags that were predicted by the LS-SA method. The Flickr owner and photo ids of these images are (king-edward@4061393892), (familymwr@4928996212), (alejandroerickson@7730525250), (wwarby@5145467790), and (1968-dodge-charger@5507716438), respectively. Seen: ($\mathcal{T}_{i,Seen}$); unseen: ($\mathcal{T}_i \backslash \mathcal{T}_{i,Seen}$); and predicted: ($\mathbb{P}_i$) as per Section 4.3.

which the predictions by the corresponding classifiers exceed θ. Some works such as Li and Snoek (2013) propose techniques to make classification of images faster. However in order to do this, these works require re-training of classifiers and cannot utilize existing classifiers for each tag. Contrary to these, in the proposed approach, we can utilize pre-trained classifiers corresponding to different tags. Other works such as the image annotation application in Wu et al. (2008) associate a test image with tags based on applying all concept detectors on the test image and then combining the predictions. However such approaches require applying all N classifiers to a given test image and are hence inefficient for a large number of tags. This process can be made more efficient by using the ontological tag tree on the set of tags. Once certain labels have been predicted for an image i, one can utilize the tree structure to decide which tags are more or less likely to be associated with the image. Thus the choice of the classifiers to test next on the image i, can be made in a more efficient manner. Therefore, a tag tree that captures the relations between the tags more effectively is expected to lead to more efficient performance by reaching the correct number of predicted tags with fewer number of binary classifications performed. The efficient classification task is formulated to measure the classification efficiency in such a setting.

This evaluation task is formulated as follows. Given N binary classifiers, the jth classifier predicts probability $P(j|i)$ of tag t_j being associated with a test image i. K binary classifications are performed for image i and the set of tags that are predicted positive among those K classifications, comprise the set of predicted tags $\mathbb{P}_{i,K}$ for image i for given K. The ground truth set of tags that are associated with image i as per the corpus, is denoted $\mathcal{T}_i$. We define performance of the classification task based on the Tag Recall as defined below.

$$\text{Tag Recall} = \frac{|\mathcal{T}_i \cap P_{i,K}|}{|\mathcal{T}_i|} \tag{8}$$

Based on the above definition, the Tag Recall is monotonically increasing with K: as more than K classifications are performed, the cardinality of the set of predicted labels can remain constant or increase, leading to a non-decreasing value for the Tag Recall.

The most naive way of choosing the order in which the N classifications are performed, would be to choose each binary classifier randomly. A more sophisticated approach can be adopted by choosing the classifiers based on the decreasing order of their class priors in the training corpus, i.e., choosing the classifier corresponding to the most frequently occurring tag first, followed by the next popular tag, and so on. We refer to these methods as Random and Prior-based, respectively. Below we discuss how the classifiers can be chosen based on their priors and a given ontological tag tree.

4.4.1. Utilizing ontological tag tree for classification

Intuitively, the procedure for using an ontological tag tree to decide how to choose the classifiers is similar to the approach for the tag prediction task. For image i, once certain tags are predicted as present, the tags in their proximity (as per the tag tree) have a higher chance of being present too, and the corresponding classifiers should be tested sooner. Similarly, the predicted absence of tags brings down the chance that tags in their proximity will be present. We define below a priority score for the tags that have not been tested for, based on their priors and the predictions for the tags that have been tested for. For image i and tag tree T, we define

$$\text{Proximity}(j) = P_{prior}(j) + \sum_{c \in C_{tested}} (P(c|i) - \theta) \times (1 - dist'(c, j)) \tag{9}$$

where $dist'(c, j)$ is the distance $dist(c, j)$ as defined in Section 4.3.1 for various methods, normalized with respect to its maximum value such that $dist'(c, j)$ varies between 0 and 1. $P_{prior}(j)$ is the prior probability of tag t_j, and C_{tested} is the set of tags that image i has been tested for. Algorithm 3 utilizes (9) and presents the algorithm employed to decide the order of classifications for each image i. For a given image i and given K, the set of labels predicted as present, i.e., $\mathbb{P}_{i,K}$ can be obtained using Algorithm 3.

Algorithm 3.
Algorithm for efficient classification by using ontological tag tree.

```
Input:
• $C_j$: classifier for label $j$ corresponding to tag $t_j$ ($\forall j \in \mathcal{T}$), that can provide $P(j|i)$ for image $i$; threshold $\theta$
• $P_{prior}(j)$: prior probability for tag $t_j$
• $dist'(t,t')$: normalized inter-tag distances calculated based on given tag tree as outlined in Section 4.4.1
• $K$: number of binary classifications to perform
Initialize: $L_{tested} = [\ ]$
Loop While $|L_{tested}| \leq K$
    • Calculate priority score for all tags in ($\mathcal{T} \setminus L_{tested}$) using (9)
    • Test classifier $C_{\hat{j}}$ corresponding to tag $t_{\hat{j}}$ with highest calculated priority score to get $P(\hat{j}|i)$.
    • $L_{tested} = L_{tested} \cup \hat{j}$.
EndWhile. Output: $\mathbb{P}_{i,K} = \{j : P(j|i) > \theta, j \in L_{tested}\}$
```

Flickr corpus: We use the same set of 117 Flickr tags as described in Section 4.3. In order to train image classifiers 500,000 Flickr training images are used. Each tag t_j is treated as a class. Positive training images are those that are associated with the corresponding tag t_j and negative training images are those that are not. We use a ratio of 1:10 for number of positive instances to number of negative instances for each class. SIFT features (Lowe, 2004) are used to train binary SVM classifiers for each class. We provide results for the Random and Prior-based methods as discussed in Section 4.4, and for methods 2 through 6 described in Section 4.3.2 using the approach outlined in Section 4.4.1. For $\theta = 0.5$, the Tag Recall observed with varying number of classifiers tested, i.e., K, are shown in Fig. 10. Note that Sigurbjörnsson and Van Zwol (2008) do not utilize the symmetric similarities for an application as proposed in Section 4.4; however we utilize the symmetric Jaccard similarities as per Algorithm 3 and compare against our approach for an understanding of the trade-off between performance and space requirement. It is clear that for a given K, our approach outperforms the tag tree constructed from WordNet, the tag graph using Google Distance, and the ones that select classifiers randomly or solely based on prior. Performance of the LS-SA method is very close to that of the Symmetric sum method despite having several orders of savings in the space requirement as discussed in Sections 1.1 and 4.3. For Tag Recall of around 4.4%, the tag tree based on proposed approach (LS-SA) requires only 11 classifications compared to 21 for LS-WAH, 31 for Google Distance based, 51 for WordNet, 36 Prior based, and 61 when the classifiers are randomly selected. These correspond to 91%, 182%, 363%, 227%, 454% additional classifications used by these methods respectively, compared to the proposed method with Similarity Approximation (LS-SA). The

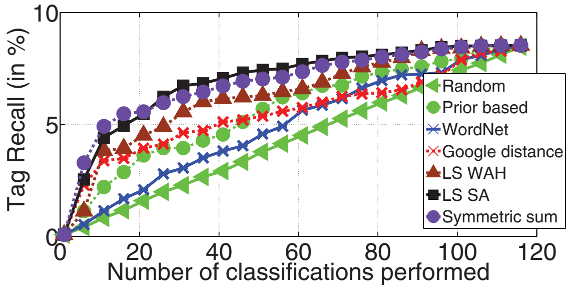


Fig. 10. Tag Recall obtained with respect to number of classifications performed for Flickr corpus. θ is chosen to be 0.5.

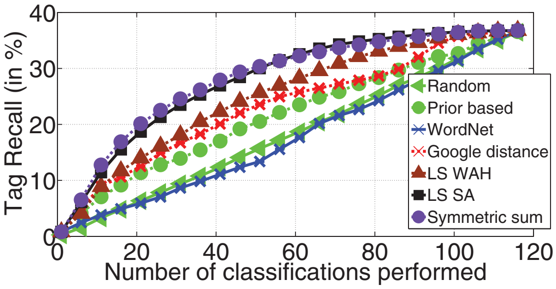


Fig. 11. Tag Recall obtained with respect to number of classifications performed for Flickr corpus. θ is chosen to be 0.33.

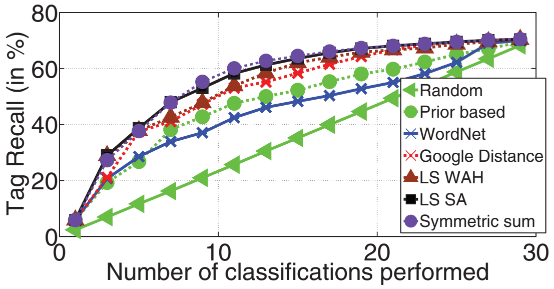


Fig. 12. Tag Recall obtained with respect to number of classifications performed for Stock images corpus. θ is chosen to be 0.5.

WordNet based method performs worse than the Prior-based method, implying that the using semantics based distance with priors is less efficient than using priors alone. Visual-feature based image classification is a hard problem and the Tag Recall for 117 classes even after all 117 classifiers are tested, is observed to be less than 10%. This is also a result of the individual classifiers and of the chosen threshold θ. Note that the Tag Recall (8) is different from recall of a class. For $\theta = 0.5$, when all the classifiers are tested, the average recall over all classes is 10.5% and the average precision is 15.9%.

A more lenient (i.e., lower) θ would lead to higher Tag Recall as can be seen in Fig. 11. $\theta = 0.3$ corresponds to an average recall of 37.7% and average precision of 4.9%. A lower θ also implies less relative weight given to $P_{Prior}(j)$ in (9) and as a result, the WordNet method performs worse than even the Random method.

Stock images corpus: We perform experiments on the same corpus that was described in Section 4.3. In order to show the applicability of our approach to different modalities that can be used to represent a given resource, we utilize the textual caption accompanying the image i to represent the image. A TF-IDF based bag-of-words representation is used for each image, followed by binary SVM classifiers for each class. As in the case of Flickr, we present variation of Tag Recall with K for various methods for $\theta = 0.5$ in Fig. 12. $\theta = 0.5$ corresponds to an average recall of 73.6% and average precision of 75.6%. Our approach is seen to be able to better decide which classifiers to test, for a given K and as a result, leads to higher Tag Recall for a fixed K. For Tag Recall of around 48%, the tag tree based on the proposed approach (LS-SA) requires only 7 classifications compared to 9 for LS-WAH, 9 for Google Distance based, 15 for WordNet, 11 for only prior based and 21 when the classifiers are randomly selected. These correspond to 29%, 29%, 114%, 57%, 200% additional classifications used by these methods respectively, compared to the proposed method with Similarity Approximation (LS-SA).

It should be noted that the goodness of predictions i.e., the precision and recall obtained when all classifiers are tested, is essentially a property of the set of classifiers, which are the same for various methods used for this task. Based on the experiments, we have demonstrated that using ontological tag trees constructed through the proposed approach makes selection of classifiers much more efficient as compared to using purely semantics based tag tree or tag graphs built using commonly used techniques. We have also shown that we can achieve a performance almost as high as other approaches despite having several orders less space requirement.

5. Robustness analysis

We provide an analysis of the robustness of the proposed approach for constructing ontological tag trees. For the purpose of this section, we refer to the resource tag data using which the tag tree is constructed, as training data. The resource tag data over which the constructed tag tree is tested for evaluation purpose, is referred to as the test data. As can be seen in Section 4, the LS-SA method has consistently outperformed LS-WAH across different corpora and for both data-driven tasks. Therefore, in this section we only study the robustness of the LS-SA method, i.e., the proposed local search based approach using Similarity Approximation based objective function (4). In order to evaluate the constructed tag trees under various scenarios, we provide evaluation using tag prediction task as detailed in Section 4.3. Annotated image corpora are used for this purpose. We study the robustness of proposed approach with respect to label noise, difference between training and test data, and the size of training data. These are discussed in detail below.

5.1. Robustness to label noise in training data

As described in Section 1, a vast majority of the user generated data available over the Internet has noisy labels (tags) associated with the resources. We study the effect of different degrees of label noise on the tag tree constructed from such a noisy corpus. Fundamentally, we attempt to understand how robust the proposed tag tree construction approach is, to different degrees of label noise. We also attempt to answer questions such as – how much noise is too much for tag tree construction?

We work with stock image corpus since stock images are professionally curated and have little to no label noise, thus providing a better control on the amount of noise in the experimental data. We select top 150 tags (keywords) from this corpus and remove those that do not occur in Flickr or in WordNet. This gives a set $\mathcal{T}$ of 108 tags. Each stock image has on an average 3 tags amongst $\mathcal{T}$. A total of 100,000 stock images are used in training set and 50,000 stock images in the test set. In order to simulate varying degrees of label noise in training set, we replace stock images in the training data with artificially created images with noisy tags. The total number of training images (stock images and noisy images) is kept constant at 100,000. The artificially created noisy images are defined as images having exactly 3 randomly chosen tags from $\mathcal{T}$. The test set is not varied. Fig. 13 shows the variation of Overall Tag Prediction Accuracy (%) as defined in Section 4.3, with number of images that were from stock images in the training set. It can be seen that even with 87.2% noisy images in the training set, the performance of the constructed tag tree (39%) is very close to the performance when there are no noisy images at all (40%). Even when 99.2% of the images in training set are noisy, the Overall Tag Prediction Accuracy is 31.4%. An explanation for such performance even at high levels of noise is that the noisy images have uniformly random distribution of tags. The overall effect of adding noisy images to a corpus can be understood as adding certain intensity of white noise to the co-occurrence counts between tags. Unless the noise intensity dominates the corpus statistics, the relative order of pair-wise connections between tags remains fairly unchanged. In other words, since the noisy images have no strong biases towards specific tag-pairs, the performance of tag trees constructed using such a hybrid training set is not severely affected.

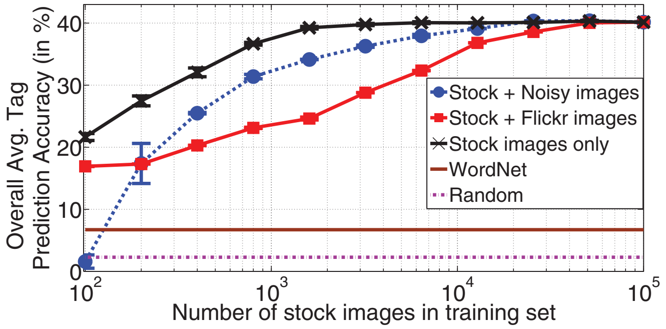


Fig. 13. Overall tag prediction score (in %) obtained by proposed approach in Section 4.3.1. The training set used to construct an ontological tag tree is formed by using certain number of stock images, with (A) Flickr images, or (B) noisy images, or (C) none. Testing of the constructed tag tree is done on images of stock photos only. For comparison, the Overall Tag Prediction Accuracies for Random method, and WordNet method (as outlined in Section 4.3.2) are also provided.

5.2. Robustness to difference between training and test data

In several machine learning applications, the data over which a model is trained has certain amount of distributional difference as compared to the data over which it is tested. Looking at the construction of tag tree as model training, the degree to which a tag tree will be effective on a test set is a function of the difference between the test set and the training set using which the tag tree is constructed. Here we study how the performance of a tag tree varies for different degrees of difference between the training and the test sets. We utilize the same stock images corpus as used in Section 5.1. In order to vary the difference between training and test sets, we replace varying number of stock images in training set with images from Flickr corpus (Section 4.1). The total number of images (stock images and Flickr images) used for training of tag tree is kept constant at 100,000. Fig. 13 shows the variation of Overall Tag Prediction Accuracy (%) with number of images that were from stock images in the training set. It is interesting to note that for the same number of stock images, adding completely noisy images leads to better performance than adding Flickr images. For instance, for the case when 99.2% of training images are from Flickr, performance of the constructed tag tree (23.1%) is seen to be substantially worse than when 99.2% of training images are noisy (31.4%). The reason for this is that the tag distribution in Flickr corpus is not random, unlike in the case of noisy images. As a result, there are strong relationships between tags in Flickr corpus that are absent in stock image corpus. Constructing a tag tree on such a hybrid corpus attempts to capture corpus statistics of varying numbers of Flickr images and stock images, and in the process, is less effective at capturing the relationships that were present in the test data that has only stock images.

5.3. Size of training data

In this section, we study the effect of size of training set on construction of tag trees. Fig. 13 shows the variation of Overall Tag Prediction Accuracy (%) with number of stock images in the training data. Here the size of training set is equal to the number of stock images in it and there are no Flickr images or noisy images. The performance of constructed tag tree improves with size of training data and almost saturates after certain number of training images. This is quite similar to the variation in performance of most machine learning models with the size of data over which they are trained. The performance of constructed tag tree from only 800 stock images is very close to the performance when 100,000 images are used. Even when only 100 stock images are used, the constructed tag tree performs much better than the tag trees using 100 stock images with 99,900 Flickr or noisy images. Based on above, one can conclude that for the construction of ontological tag trees, it is better to use fewer training images than a larger set which is noisy or dissimilar to the test set.

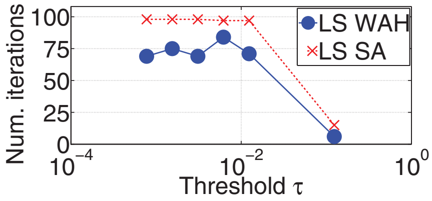


Fig. 14. Variation of number of iterations with τ.

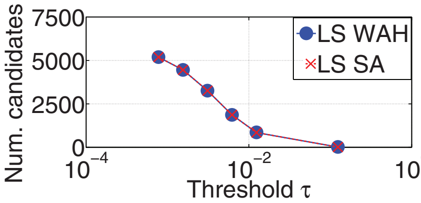


Fig. 15. Variation of number of candidate neighbors with τ.

6. Discussion

In this section, we provide a discussion on the key insights and observations made during the course of study on ontological tag trees.

As outlined in Section 3, the motivation for the objective function (3) in the proposed approach is that in order to get a lower value for (3), it is more important to separate the tags t_i and t_j having high $J_{\mathcal{T}}(i, j)$ with less number of hops, as compared to the tags with low $J_{\mathcal{T}}(i, j)$. However, separating the latter set of tags by fewer hops would also lower the value of (3). Since (3) attempts to minimize the weighted number of hops between tag pairs, a local search in the space of all possible trees on $\mathcal{T}$ would lead to connecting most tags to a central tag, thus forming a star graph. Such a minimum weighted hops spanning tree problem on an all connected graph can be solved in polynomial time using Gomory–Hu trees (Panigrahi, 2008); however star graphs need not reflect the true relationship between tags in a corpus. In order to ensure that the proposed approach does not induce artificial structures because of bias of the objective function, we constrain the search space to be spanning trees of a suitably constructed Similarity Graph with the help of a threshold τ, as outlined in Section 3.2. Experimentally, constraining as above does succeed in ensuring that we do not get a star graph. However upon visual inspection, the proposed approach yields star subtrees connected through their centers. The objective function (4) in comparison has no such bias for shorter trees and as a result, the results for proposed approach using (4) outperform those using (3). Figs. 14–17 show the variation of different aspects of constructed tag trees with τ for 117 Flickr tags. Fig. 15 shows that with increasing τ, the number of tag trees that are eligible candidates in an iteration of the local search, drops drastically for both LS-WAH and LS-SA. As discussed above, a lenient τ leads to the LS-WAH based local search converging to star graphs which are an artifact of the procedure and not of the true relationships between the tags. As a result, for lower τ, the performance of LS-WAH based approach degrades substantially as can be seen in Fig. 17. For large values of τ, the number of candidate neighbors at each iteration of the local search becomes less. This constrains the local search and prevents it from attaining a lower value for the objective function (4). This leads to a drop in the performance of LS-SA as can be seen in Fig. 17. We have chosen a threshold as the median of the pair-wise Jaccard similarity values since it offers a convenient trade-off between number of candidates and performance for LS-SA.

At each iteration of the proposed approach, there are $O(N^2)$ neighbors of a tree to explore, where $N = | \mathcal{T} |$. Calculating the pair-wise

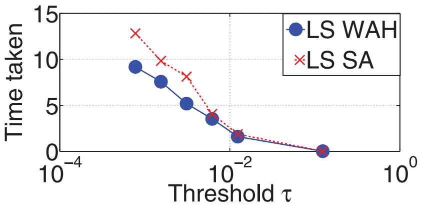


Fig. 16. Variation of time (hours) taken by local search to converge.

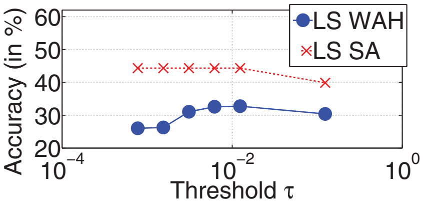


Fig. 17. Variation of the Average Tag Prediction Accuracy with τ.

distances (or hops) and computing (3) or (4) require $O(N^2)$ operations per neighbor (Ford & Fulkerson, 2010). Thus, the complexity of the proposed approach, for either objective function, is $O(N^4)$ per iteration. A bottleneck of the proposed tag tree construction approach is the time taken for local search to terminate. Fig. 16 shows the variation of the time taken by the proposed local search based approach, with respect to τ. Matlab on a 2.9 GHz, 8GB RAM 4 core processor is used. One way to reduce the time taken at each iteration is to reduce the number of neighbors. This is achieved by constraining the solution space to spanning trees of a Similarity Graph. While the order of complexity at each iteration remains at $O(N^4)$, the time taken is much less. For instance, when τ is chosen as median of the pair-wise Jaccard similarity values as in Section 4, the time taken is nearly half of the time taken when $\tau = 0$, which corresponds to having search space as all possible trees on $\mathcal{T}$. Thus, for the proposed approach using objective function (3), constraining the local search to spanning trees of a Similarity Graph is necessary to ensure that we do not get a trivial solution. Doing so also makes the local search faster. As compared to this, for the proposed approach using objective function (4), using a Similarity Graph only makes the optimum search faster as can be seen in the reduced number of candidates in Fig. 15 and the reduced number of iterations to converge, as shown in Fig. 14.

Intuitively, using (4) as the objective function minimizes the weighted L1 norm of the difference between pair-wise Jaccard similarity and the similarity estimated using a tag tree, thus trying to approximate the normalized symmetric second order statistics of a corpus using a tree on the set of tags. We have chosen weighted L1 norm since it was observed to lead to the best results across different corpora among other variants such as L1, L2 and weighted L2 norm. Implementation of the similarity estimated using a tag tree (5) can be conveniently done by using Logarithms and Ford Fulkerson Algorithm (Ford & Fulkerson, 2010). Also, since the LS-SA approximates the pair-wise similarities $J_{\mathcal{T}}(i, j)$ in $O(N)$ as compared to $O(N^2)$, approaches such as Sigurbjörnsson and Van Zwol (2008) that utilize $J_{\mathcal{T}}(i, j)$ lead to equal or only marginally better performance, despite having several orders higher space requirement.

7. Conclusions and future work

In this section we summarize the research contributions of our paper, key practical insights, and advantages and limitations as compared to existing expert systems. We also discuss limitations of our work, and provide several suggestions for future research directions.

We have proposed ontological tag trees to enable expert systems address the sparsity of folksonomies effectively and in a space efficient manner. Ontological tag trees or tag trees are defined as simple trees on the set of tags in a corpus. The construction of tag trees is formulated as an optimization problem on corpus based statistics, and is solved through a novel local search based approach. We have shown that this approach can be used to build tag trees for tags obtained from two corpora, one composed of noisily annotated Flickr images and the other composed of cleanly annotated stock images. To validate the utility of the constructed ontological tag tree, we proposed two evaluation tasks involving tag prediction in images. We have demonstrated that for the task of predicting the unseen tags of a given image with a partially observed set of tags, the proposed ontological tag trees outperform those constructed using only semantic relationships, or tag graphs constructed using commonly used techniques that have comparable space requirements. In the second task, we have shown that the tag tree obtained from the proposed approach makes the process of using appropriate classifiers to tag an untagged test image more efficient. Robustness analysis shows that the proposed approach is fairly robust to tag noise and differences between the training and test set distributions.

Compared to previous expert systems, our work offers significant advantages. In addition, several key insights can be derived from our study. Since expert systems such as Uddin et al. (2013) based on only semantic relationships fail to capture the data-specific relationships between tags, their performance is significantly lower on data-driven tasks than that of ontological tag trees constructed using the proposed approach. This is demonstrated through evaluations in Section 4. In addition, we show that even though ontological tag trees have space requirement of only $O(N)$ for N tags, as compared to $O(N^2)$ for tag graphs, tag trees can provide equal or better performance than several existing expert systems on the evaluation tasks. Particularly, compared to Sigurbjörnsson and Van Zwol (2008) which may require 27 terabytes of space to store pair-wise similarities, our approach requires less than 50 megabytes, and still achieves almost equal performance. Ontological tag trees can thus offer a very convenient method for expert systems to capture corpus based relationships for tasks such as tag prediction, and efficient resource classification. The significant savings in space requirement facilitates practical deployment of the expert systems even on computing devices that do not have gigantic memory available. The third key advantage of using ontological tag trees is that their construction does not depend on the availability of content-based features (such as textual or visual features). Thus as compared to previous expert systems such as Chen et al. (2015), Hsieh et al. (2009), Li et al. (2009), Sun et al. (2011) and Wu et al. (2009), our approach can help alleviate sparsity of online folksonomies even in domains where extracting content-based features may be inefficient or infeasible (Huang et al., 2010; Song et al., 2010; Yin et al., 2009; Zanetti et al., 2008). Lastly, as discussed in Section 6, the performance of the LS-SA approach is significantly better than that of the LS-WAH approach. It can thus be recommended that the LS-SA approach be used to construct ontological tag trees.

While the proposed ontological tag trees can help expert systems achieve high performance in a space efficient manner, there are certain limitations of the proposed approach. As discussed in Section 6, the time taken per iteration of the local search based tag tree construction approach varies as $O(N^4)$. For large number of tags, the total time taken to construct tag trees can be high. In addition, the tag trees only capture undirected weighted relationships between tags. As a result, any directed relationships that may be present in the corpus (for example subsumption relationships (Schmitz, 2006)) are lost by the constructed tag trees.

We provide several suggestions for future work directions based on our work. Since the tag tree construction approach has a limitation of high time requirement, one direction of future work could be to make the proposed approach faster by choosing a higher or an adaptive τ, with the possible trade-off being increasing the constraint on the local search, thereby achieving a sub-optimal tag tree. Another direction of future work could be to keep a low τ but develop distributed and possibly approximated versions of the proposed approach, such that available distributed clusters, for example Amazon EC2 cloud server (Amazon EC2, 2015) can be leveraged. In the current work, the LS-SA approach can be thought of as approximating the symmetric second order corpus statistics. An extension of the work could be to construct tag structures that approximate higher order corpus statistics. In addition, future work could utilize the constructed tag trees for other practical applications such as tag recommendation: given set of tags associated with a test images, which additional tags would you associate with the image? Removal of potentially noisy tags can also be studied based on whether a tag appears too far in the tree from other tags associated with a given image. Lastly, a hierarchical taxonomy could be constructed in future, where the edges between the nodes are directed, by adopting similar techniques and using the co-occurrence data from the corpus.

Acknowledgments

This research was partially supported by an internship at Yahoo Labs, and partially by the Center for Wireless Communications at University of California San Diego.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2015.07.057.

References

Aarts, E., & Korst, J. (1988). *Simulated annealing and Boltzmann machines*. John Wiley and Sons Inc.

Aarts, E. E. H., & Lenstra, J. K. (1997). *Local search in combinatorial optimization*. Princeton University Press.

Amazon EC2 (2015). http://aws.amazon.com.

Anand, D., & Mampilli, B. S. (2014). Folksonomy-based fuzzy user profiling for improved recommendations. *Expert Systems with Applications, 41*, 2424–2436.

Begelman, G. (2006). Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the collaborative web tagging workshop at ACM conference on world wide web*. ACM.

Buitelaar, P., Cimiano, P., & Magnini, B. (2005). *Ontology learning from text: methods, evaluation and applications*: Vol. 123. IOS press.

Cayley, A. (1894). *The collected mathematical papers of Arthur Cayley*: Vol. 7. The University Press.

Chen, X., Liu, Z., & Sun, M. (2015). Estimating translation probabilities for social tag suggestion. *Expert Systems with Applications, 42*, 1950–1959.

Cilibrasi, R. L., & Vitanyi, P. M. (2007). The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering, 19*, 370–383.

Cimiano, P., Hotho, A., & Staab, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligent Research (JAIR), 24*, 305–339.

Cimiano, P., & Völker, J. (2005). Text2onto. *Natural Language Processing and Information Systems* (pp. 227–238). Springer.

Croes, G. (1958). A method for solving traveling-salesman problems. *Operations Research, 6*, 791–812.

Dietz, E., Vandic, D., & Frasincar, F. (2012). Taxolearn: a semantic approach to domain taxonomy learning. In *Proceedings of web intelligence and intelligent agent technology* (pp. 58–65). IEEE.

Djuana, E., Xu, Y., & Li, Y. (2011). Constructing tag ontology from folksonomy based on WordNet. In *Proceedings of IADIS international conference on internet technologies and society*.

Eckhardt, S., Kosub, S., Maaß, M. G., Täubig, H., & Wernicke, S. (2005). Combinatorial network abstraction by trees and distances. *Algorithms and Computation* (pp. 1100–1109). Springer.

Facebook (2015). www.facebook.com.

Fensel, D. (2001). *Ontologies*. Springer.

Flickr (2015). http://www.flickr.com.

Flickr Clusters (2015). http://www.flickr.com/photos/tags/flickr/clusters.

Ford, D., & Fulkerson, D. R. (2010). *Flows in networks*. Princeton University Press.

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability*: Vol. 174. New York: Freeman.

Griffin, G., & Perona, P. (2008). Learning and using taxonomies for fast visual categorization. In *Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies, 43*, 907–928.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of conference on computational linguistics*. ACL.

Heymann, P., & Garcia-Molina, H. (2006). Collaborative creation of communal hierarchical taxonomies in social tagging systems, *Technical report*, Stanford.

Hsieh, W.-T., Stu, J., Chen, Y.-L., & Chou, S.-C. T. (2009). A collaborative desktop tagging system for group knowledge management based on concept space. *Expert Systems with Applications, 36*, 9513–9523.

Huang, C., Fu, T., & Chen, H. (2010). Text-based video content classification for online video-sharing sites. *Journal of the American Society for Information Science and Technology, 61*, 891–906.

Jaimes, A., & Smith, J. R. (2003). Semi-automatic, data-driven construction of multimedia ontologies. In *Proceedings of IEEE international conference on multimedia and expo*. IEEE.

Jiang, S., Qian, X., Shen, J., Fu, Y., & Mei, T. (2015). Author topic model based collaborative filtering for personalized POI recommendation. *IEEE Transactions on Multimedia, 17*, 907–918.

Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (1988). How easy is local search? *Journal of Computer and System Sciences, 37*, 79–100.

Katsurai, M., Ogawa, T., & Haseyama, M. (2014). A cross-modal approach for extracting semantic relationships between concepts using tagged images. *IEEE Transactions on Multimedia, 16*, 1059–1074.

Kim, H., & Kim, H.-J. (2014). A framework for tag-aware recommender systems. *Expert Systems with Applications, 41*, 4000–4009.

Li, X., & Snoek, C. G. (2013). Classifying tag relevance with relevant positive and negative examples. In *Proceedings of ACM international conference on multimedia*. ACM.

Li, X., Snoek, C. G., & Worring, M. (2009). Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia, 11*, 1310–1322.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 44*, 2245–2269.

Liu, D., Hua, X.-S., Yang, L., Wang, M., & Zhang, H.-J. (2009). Tag ranking. In *Proceedings of ACM conference on world wide web*. ACM.

Liu, J., Wang, B., Li, M., Li, Z., Ma, W., Lu, H., & Ma, S. (2007). Dual cross-media relevance model for image annotation. In *Proceedings of ACM conference on multimedia*. ACM.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*, 91–110.

Mani, I., Samuel, K., Concepcion, K., & Vogel, D. (2004). Automatically inducing ontologies from corpora. *Corpus, 9*, 19–024.

Miller, G. A. (1995). WorldNet: a lexical database for English. *Communications of the ACM, 38*, 39–41.

Navigli, R., Velardi, P., & Gangemi, A. (2003). Ontology learning and its application to automated terminology translation. *IEEE Intelligent Systems, 18*, 22–31.

Neshati, M., Alijamaat, A., Abolhassani, H., Rahimi, A., & Hoseini, M. (2007). Taxonomy learning using compound similarity measure. In *Proceedings of IEEE web intelligence*. IEEE.

Open Directory Project (2015). http://www.dmoz.org.

Panigrahi, D. (2008). Gomory–Hu trees. *Encyclopedia of algorithms* (pp. 364–366). Springer.

Porzel, R., & Malaka, R. (2004). A task-based approach for ontology evaluation. In *Proceedings of ECAI workshop on ontology learning and population*.

Rita WordNet Library (2015). http://www.rednoise.org/rita/wordnet/documentation.

Sanderson, M., & Croft, B. (1999). Deriving concept hierarchies from text. In *Proceedings of SIGIR conference on research and development in information retrieval* (pp. 206–213). ACM.

Schmitz, P. (2006). Inducing ontology from Flickr tags. In *Proceedings of collaborative web tagging workshop at ACM conference on world wide web*.

Sigurbjörnsson, B., & Van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *Proceedings of ACM conference on world wide web*. ACM.

Song, Y., Zhao, M., Yagnik, J., & Wu, X. (2010). Taxonomic classification for web-based videos. In *Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE.

Sun, K., Wang, X., Sun, C., & Lin, L. (2011). A language model approach for tag recommendation. *Expert Systems with Applications, 38*, 1575–1582.

Uddin, M. N., Duong, T. H., Nguyen, N. T., Qi, X.-M., & Jo, G. S. (2013). Semantic similarity measures for enhancing information retrieval in folksonomies. *Expert Systems with Applications, 40*, 1645–1653.

Velardi, P., Navigli, R., Cuchiarelli, A., & Neri, R. (2005). Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies, *Ontology learning from text: methods, applications and evaluation: vol. 123* (pp. 92–106). IOS Press.

Verma, C. K., Mahadevan, V., Rasiwasia, N., Aggarwal, G., Kant, R., Jaimes, A., & Dey, S. (2014). Construction of tag ontological graphs by locally minimizing weighted average hops. In *Proceedings of companion publication of ACM conference on world wide web*.

Wu, L., Hoi, S. C., Jin, R., Zhu, J., & Yu, N. (2009). Distance metric learning from uncertain side information with application to automated photo tagging. In *Proc. of ACM conference on multimedia*. ACM.

Wu, L., Hua, X.-S., Yu, N., Ma, W.-Y., & Li, S. (2008). Flickr distance. In *Proceedings of ACM conference on multimedia*. ACM.

Xia, Z., Feng, X., Peng, J., Wu, J., & Fan, J. (2015). A regularized optimization framework for tag completion and image retrieval. *Neurocomputing, 147*, 500–508.

Yin, Z., Li, R., Mei, Q., & Han, J. (2009). Exploring social tagging graph for web object classification. In *Proceedings of SIGKDD conference on knowledge discovery and data mining*. ACM.

YouTube (2015). www.youtube.com

Zanetti, S., Zelnik-Manor, L., & Perona, P. (2008). A walk through the web's video clips. In *Proceedings of IEEE conference on computer vision and pattern recognition workshops*. IEEE.

Zheng, N., & Li, Q. (2011). A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications, 38*, 4575–4587.