Knowledge Base Enrichment by Relation Learning from Social Tagging Data

Knowledge Base Enrichment by Relation Learning from Social Tagging Data

Hang Dong[a,b,d], Wei Wang[b], Frans Coenen[a], Kaizhu Huang[c]

[a]*Department of Computer Science, University of Liverpool, Liverpool, United Kingdom*
[b]*Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China*
[c]*Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China*
[d]*Centre for Medical Informatics, Usher Institute of Population Health Sciences and Informatics, University of Edinburgh, Edinburgh, United Kingdom*

Abstract

There has been considerable interest in transforming unstructured social tagging data into structured knowledge for semantic-based retrieval and recommendation. Research in this line mostly exploits data co-occurrence and often overlooks the complex and ambiguous meanings of tags. Furthermore, there have been few comprehensive evaluation studies regarding the quality of the discovered knowledge. We propose a supervised learning method to discover subsumption relations from tags. The key to this method is quantifying the probabilistic association among tags to better characterise their relations. We further develop an algorithm to organise tags into hierarchies based on the learned relations. Experiments were conducted using a large, publicly available dataset, Bibsonomy, and three popular, human-engineered or data-driven knowledge bases: DBpedia, Microsoft Concept Graph, and ACM Computing Classification System. We performed a comprehensive evaluation using different strategies: relation-level, ontology-level, and knowledge base enrichment based evaluation. The results clearly show that the proposed method can extract knowledge of better quality than the existing methods against the gold standard knowledge bases. The proposed approach can also enrich knowledge bases with new subsumption relations, having the potential to significantly reduce time and human effort for knowledge base maintenance and ontology evolution.

Keywords: Knowledge discovery, Knowledge base enrichment, Ontology learning, Social tagging, Probabilistic association analysis, Classification

1. Introduction

Social tagging has been a popular functionality offered by most social media platforms, allowing users to provide "key words" or tags to describe resources of interest. Over the years, these accumulated tags form "folksonomies" (a portmanteau of "folk" and "taxonomies") [46], which are perceived as valuable metadata to supplement controlled vocabularies for resource organisation [50], information retrieval and recommendation [27, 38]. Unfortunately, many such folksonomies have gradually developed into dormant collections of unstructured, noisy and ambiguous "keywords" with little usefulness [38].

There has been a consensus in the research communities that social data, including tagging data, can be used to harvest "collective intelligence". Previous research has shown that tagging data can be used to capture emergent semantics for ontology learning and be transformed into structured knowledge, such as concept hierarchies or lightweight ontologies [21, 36]. Nevertheless, the task of discovering quality knowledge is challenging due to the complex and ambiguous meanings of tags. Tagging data is also sparse and contains little contextual information, making the task very different from mining relations from text documents.

Many existing methods infer tag relations by exploiting the co-occurrence information as reviewed in [17, 21], for example, through a heuristic based set inclusion measure [35, 36] or graph centrality [26] in a tag-tag network. However, they simply ignore the meanings of tags, and it is difficult to formally interpret the meanings of the inferred relations. Some methods make use of lexical information and define tag relations by matching to external resources [3]. An obvious limitation is that they cannot handle tags not covered by the external resources. Moreover, it is possible that tags are sometimes not matched to the right senses [11]. Another class of method employs machine learning techniques, especially supervised learning, to predict relations. Research in this line also leverages co-occurrence features [40] and usually relies on specific contents from the tagged resources [49].

This work aims to address the two major issues in the existing research: first, inference is primarily done through analysis of the tag co-occurrence and largely overlooks the complex meanings of tags, which often leads to low prediction accuracy; second, existing evaluation studies regarding the quality of the knowledge discovered from large-scale datasets are not adequate, e.g., the study in [42] did not formally evaluate the *enriched knowledge*. Our study focuses on learning from academic social tagging data, i.e., tagging data for academic publications and resources. The task is more challenging than learning in general domains as the academic domain contains sparser data [18, 26]. We extend our previous work on learning relations from social tags only at the relation-level [14], to more comprehensive domains with three knowledge bases, and propose methods for hierarchy generation and knowledge enrichment. The main difference from the existing research based on supervised learning is the quantification of probabilistic associations among tags in order to help the supervised models better predict their relations. It is assumed that a tag in a taxonomy is potentially ambiguous and might have complex meanings. A probabilistic based framework is a natural choice for representing the meanings of a tag. The features are inferred according to the cognitive process towards interpreting the meaning of tags. The contributions of the research include:

- A supervised framework for relation learning from social academic data, extending previous work in [14]. A tag is viewed as a complex entity that potentially has different meanings under different contexts or subject areas. We resort to techniques for probabilistic topic modelling to represent a tag as a distribution of latent topics. With this representation, we perform probabilistic association analysis to extract a set of domain independent features to predict subsumption relations. The features are extracted according to three assumptions (topic similarity, topic distribution, and probabilistic association) based on our understanding towards the tags.

- A hierarchy generation algorithm on top of the relation learning model to produce hierarchies with a predefined concept. Evaluation shows that it is particularly useful in enriching knowledge bases (KBs).

- A comprehensive evaluation using the large, publicly available Bibsonomy dataset, and three knowledge bases, DBpedia, Microsoft Concept Graph, and the ACM Computing Classification System; and three evaluation strategies: relation-level evaluation, ontology-level evaluation, and knowledge base enrichment based evaluation. To our best knowledge, this is one of the largest and most systematic evaluation studies for relation learning from academic social data (*cf.* [42]); this is also the first study focusing on enriching large-scale KBs. The proposed method outperforms the state of the art in terms of F_1 score and taxonomic similarity measures when evaluated against gold standard KBs, and is further validated through human evaluation of the KB enrichment.

The rest of the paper is organised as follows. We first discuss the related work on learning subsumption relations and KB enrichment from social tagging data in Section 2. Then, we provide an overview of the supervised learning framework in Section 3 and present the Data Representation module in Section 3.2 and the Feature Generation module in Section 4. In Section 5, the hierarchy generation algorithm in the Knowledge Enrichment module is explained. In Section 6, we describe the experimental setting and results according to the three adopted evaluation strategies. Finally, Section 7 concludes the paper and discusses the future studies.

2. Related work

Tags are used by online users to annotate resources based on their own understanding [46]. The resulting folksonomies contain many emerging terms that can potentially complement the controlled vocabularies [31, 33] and can thus support resource classification [50] and retrieval [27, 38]. However, tags have a flat structure without relations among them, which limits their usefulness in effective searching, navigation and recommendation. There have been many studies considering folksonomies as important sources for mining "collective intelligence" and deriving structured knowledge [17, 21].

2.1. Knowledge discovery from social tagging data

Existing methods for extracting knowledge from social tagging data can be broadly categorised into four classes: heuristic-based, semantic grounding to external resources, unsupervised learning and supervised learning.

Heuristic based methods mostly make use of heuristics to infer relations with respect to pre-defined rules. A common heuristic is the generality measure based on set inclusion. The work in [36] detected subsumption relations between tags using the inclusion of user sets, within a dataset crawled from the general domain social tagging system Delicious[1]. The study in [35] further defined a metric called *inclusion degree* and *generalisation degree* and automatically generates hierarchies using graph-pruning algorithms. Graph centrality is another well-known heuristic in the literature [6, 26]. The research in [26] induced a taxonomy using a greedy search algorithm with the degree centrality of tag nodes in a tag similarity graph. The study in [6] extended this approach with sense disambiguation and applied betweenness centrality on a tag-tag co-occurrence network. The work in [42] evaluated both methods proposed in [6, 26] and validated the usefulness of graph centrality in creating taxonomies from tags. This class of methods heavily relies on co-occurrence information and may not derive accurate subsumption relations [21]. The co-occurrence based heuristics are sensitive to data sparsity; with the graph-centrality measure, it is more difficult to generate a hierarchy from academic social tagging data such as CiteULike[2]

[1] `https://del.icio.us`

[2] `http://www.citeulike.org/`

than from the general domains like Delicious, as the data in the former has lower *density* and *overlap* [26]. This problem has also been statistically analysed in [18]. Thus for the sparse academic social tagging data, which is the focus of this study, the co-occurrence-based heuristics are unsuitable.

Semantic grounding to external resources based methods attempt to match tags to entities in external KBs in order to find semantic relations. The work in [13] mapped social tags to concepts in WordNet to extract relations. However, WordNet is a relatively static resource and only less than half (48.7%) of the tags could be directly matched according to the study in [3]. The work in [22] used DBpedia and its interconnected datasets in the Linked Open Data Cloud to ground tags and populate an ontology. In general, it is however difficult to choose the concept with the right sense matched to a tag due to the lack of tagging context. This is because that users' collective tagging process is very different from that of lexicographers or domain experts. This tag sense disambiguation problem has been discussed in [3, 13, 22]. Even if a tag can be lexically matched to a concept in external resources, it is uncertain that their intended meanings coincide with each other [11]. A potential solution for tag sense disambiguation is to use intelligent tools and contextual sources for semantic grounding, for example, the work in [1] utilises Google search and Wikipedia articles to disambiguate and establish tag-tag relations.

Unsupervised learning based methods mostly use various clustering or dimensionality reduction techniques. The research in [48] proposed a hierarchical clustering model based on *Deterministic Annealing* to generate subsumption structures from tagging data using Delicious and Flickr[3]. However, the model could not clearly discriminate subsumption, related and equivalent relation-s. Another clustering based method using k-means [42] showed that it did not perform better than the graph-based methods [6, 26]. Other unsupervised methods attempt to find low dimensional representations of data items to discover semantic patterns. A *Probabilistic Topic Model* [7], such as *Latent Dirichlet Al-*

[3] `https://www.flickr.com/`

location (LDA) [8], is a type of generative model used to discover themes from a large collection of documents. The study reported in [30] proposed a hybrid approach utilising graph-based heuristics with contextual information inferred, using LDA, from web corpus to learn domain ontologies from tags. The study in [47] applied LDA to a collection of abstracts of scientific publications and represented concepts through a "fold-in" process. It proposed a metric, *Information Theory Principle for Concept Relationship*, to determine subsumption relations based on the asymmetric difference of the *Kullback-Leibler Divergence* of topic distributions. The work in [45] also defined similar metrics using a *Tag-Topic model*. A common issue of these methods is whether using the divergence measure is precise enough to determine relations for tagging data.

Supervised learning based methods have also been proposed. The study reported in [49] used a binary classifier to generate a taxonomy from Stack Overflow[4] tags. Both co-occurrence features and topic-based features were considered; it also made use of textual information of resources, which is often unavailable in other types of social tagging data. However, the features may not be fine-grained enough to represent the topic information in social tags. Work reported in [40] combined several popular co-occurrence based feature extraction mechanisms to develop a binary classifier. The mechanisms considered included support and confidence [41], cosine similarity, set inclusion and generalisation degree [35], mutual overlapping [9] and graph-based taxonomy search adapted from [26]. It is reported that combining these heuristics in a classifier significantly increased the F_1 Score in relation-level evaluation. However, the method has the same drawbacks as other co-occurrence based methods in that it does not take into consideration the complex meanings of tags and suffers from the data sparsity problem.

[4] `https://stackoverflow.com/`

2.2. Knowledge base enrichment from folksonomies

While many studies used KBs or ontologies to enrich folksonomies [4, 22], less research has explored the opposite case, using folksonomies to enrich KBs. However, it is generally agreed that folksonomies represent users' terminologies and can be extracted to enrich KBs. This has been validated through comparison studies between folksonomies and controlled vocabularies. The work in [31] compared the academic tags in CiteULike with Medical Subject Headings and shows they have a highly distinct lexicon and viewpoints. The study from [33] compared the Librarything tags with the Library of Congress Subject Headings and shows little overlap between ordinary users' and experts' vocabularies.

The work in [2] proposed the idea of "Folksonomised Ontology", which is a fused terminological ontology based on folksonomies and existing KBs. It suggests the so-called "3E" techniques (Extraction, Enrichment, Evolution): (1) preprocessing the social tagging data to obtain a cleaned tag set (Extraction); (2) matching the tag concepts to KBs (Enrichment); and (3) using tag-tag relations to enrich relations in existing KBs (Evolution). Co-occurrence information was primarily used to discover the relations between tags. The enrichment and evolution processes require much human intervention with visualisation techniques. A similar work presented in [20] focused on designing a visual interface for manual editing and used a similarity measure to suggest new concepts and their relations for KB enrichment in an e-learning environment. Our study explores tagging data in the academic domain, and aims at designing a more effective method to predict new, direct and precise subsumption relations to enrich widely used KBs, with minimum human intervention.

3. Supervised relation learning from social tagging data

Learning relations from tag pairs is formulated as a supervised learning problem. Before presenting the learning framework, we first introduce some formal definitions and notations used in this study.

Formally, folksonomies can be described as a collection of tuples, $\mathbb{F} := \langle U, T, R, Y \rangle$, where U, T and R are finite sets representing *users*, *tags* and *resources*, respectively; Y is a ternary relation among them, $Y \subseteq U \times T \times R$ [27]. As folksonomies are noisy, they need to be cleaned and variants of tags need to be identified. A cleaned folksonomy is denoted as $\mathbb{F}^{clean} := \langle U, C, R, Y \rangle$, where the orignial T is transformed to a new finite set C representing *tag concepts*. Each element in C is a group of tags considered to be equivalent. The task is to learn subsumption relations from the cleaned folksonomies and finally transform these to structured knowledge, $\mathbb{F}^{str} := \langle U, C, R, Y, \prec \rangle$, where $\prec$ represents the set of learned subsumption relations, $\prec \subseteq C \times C$.

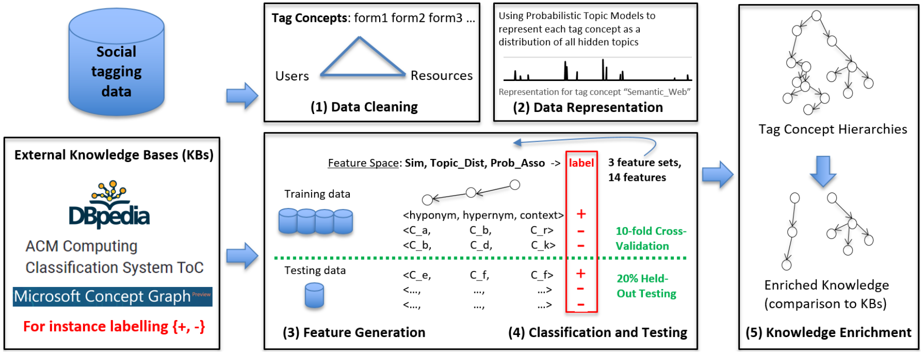
As a simple example, suppose that the raw folksonomy $\mathbb{F}$ contains four tuples regarding two users (u1 and u2) and two resources (r1 and r2), $\mathbb{F} =$ {<u1,semanticweb,r1>, <u1,socialsoftware,r1>, <u2,ontologies,r2>, <u2,semantic-web,r2>}. To create $\mathbb{F}^{clean}$, the tag variants 'semanticweb' and 'semantic-web' will be unified to a standard form of 'Semantic_Web', and 'socialsoftware' to 'Social_Software'. To form $\mathbb{F}^{str}$, the subsumption relation <ontology $\rightarrow$ Semantic_Web> should be specified.

The subsumption relation learning process can be formalised as a binary classification problem. Let $\mathcal{X}$ be the set of instances or triples in the input space and $\mathcal{Y} = \{0, 1\}$ be the set of positive and negative labels for the instances. Each instance is represented as a vector, $\vec{x_i} = (f_1(C_a, C_b, C_r), ..., f_m(C_a, C_b, C_r))$, $(C_a \neq C_b)$, where C_a and C_b are two concepts whose relation is to be determined. C_r denotes the context of the instance. C_r can be either the direct or indirect parent concept of C_b. The identifiers f_1 to f_m represent a set of different feature extraction functions based on probabilistic topic analysis. The objective is to learn a function $h : \mathcal{X} \rightarrow \mathcal{Y}$ to predict the subsumption relations between tags.

3.1. Overview of the method

Figure 1 provides an overview of the proposed learning framework, which consists of five blocks: (1) Data Cleaning: transforming $\mathbb{F}$ to $\mathbb{F}^{clean}$ by unifying tag variants and removing infrequent tags; (2) Data Representation: using

Figure 1: Architecture of the system to learn relations from social tagging data


Social tagging data
Tag Concepts: form1 form2 form3 ...
Users
Resources
(1) Data Cleaning
Using Probabilistic Topic Models to represent each tag concept as a distribution of all hidden topics
Representation for tag concept "Semantic_Web"
(2) Data Representation
Tag Concept Hierarchies
External Knowledge Bases (KBs)
DBpedia
ACM Computing Classification System ToC
Microsoft Concept Graph
For instance labelling {+, -}
Feature Space: Sim, Topic_Dist, Prob_Asso -> label
3 feature sets, 14 features
Training data
<hyponym, hypernym, context>
<C_a, C_b, C_r>
<C_b, C_d, C_k>
+
-
-
10-fold Cross-Validation
Testing data
<C_e, C_f, C_f>
<..., ..., ...>
<..., ..., ...>
+
-
-
20% Held-Out Testing
(3) Feature Generation
(4) Classification and Testing
Enriched Knowledge (comparison to KBs)
(5) Knowledge Enrichment

Probabilistic Topic Models to represent each tag concept as a distribution of latent topics in a lower dimensional semantic space; (3) Feature Generation: generating features based on the probabilistic representation for tags and different functions for calculating topic similarity, topic distribution and probabilistic association; (4) Classification and Testing: automatic creation of training and testing data through semantic grounding to external KBs, followed by training and testing of the classification models; and (5) Knowledge Enrichment: using a hierarchy generation algorithm to transform $\mathbb{F}^{clean}$ to $\mathbb{F}^{str}$. At the end, the results are presented to human domain experts for verification.

It should be noted, that input to the Feature Generation block contains triples of tag concepts. Given the context tag concept C_r, the tag concept C_a, is a direct hyponym (narrower concept) of C_b if a subsumption relation can be established between them; in other words, C_b is a direct hypernym (broader concept) of C_a. Further notation used throughout the rest of the paper is listed in Table 1.

3.2. Data representation

Social tagging data, if projected along the resource and tag dimensions, have very high dimensionality and are extremely sparse. To address the sparsity problem as well as the ambiguity of meaning in tags, it is necessary to reduce the dimensionality of the tagging data. Each resource, $r \in R$ in $\mathbb{F}^{clean}$, is initially represented as "bag of tags", analogous to the "bag of words" model in

Table 1: Notations

Notation	Description
C	Vocabulary of tag concepts
C_a	A tag concept a
$\mathbf{z}$	The set of all hidden topics
$\lvert\mathbf{z}\rvert$	Number of hidden topics
$\mathbf{z}_a^{sig}$	The set of all significant topics for the concept C_a
$\lvert\mathbf{z}_a^{sig}\rvert$	Number of significant topics for the concept C_a
z	A hidden topic
N	Number of occurrence of all tag concepts
N_z	Number of occurrence of all tag concepts assigned to topic z
$v(C_a)$	The topic distribution vector of tag concept a
$R_{a,b}$	Common parent tag concept of C_a and C_b

Information Retrieval. We wish to infer a low dimensional topic structure from the large collection of resources and tags. With Latent Dirichlet Allocation [8], approximated by Gibbs sampling [24], we can obtain the topic assignment for each tag in all the resources, and consequently, two probabilistic distributions; the tag-topic distribution $p(C|\mathbf{z})$ which represents a latent topic in terms of distributions of tags; and the topic-resource distribution $p(\mathbf{z}|R)$ which represents a resource as a distribution of latent topics.

However, the entities of interest in our work are tags and we need to represent a tag concept in terms of the distribution of latent topics. This can be calculated by using the Bayes' rule with $p(C|z)$ and $p(z)$ as shown in Equation 1. The prior probability $p(z)$ has been always assumed as a uniform distribution in the literature [23, 25]. However, this often does not hold in real-world data. Therefore, we propose to use a non-uniform prior probability $p(z)$, computed as the ratio of the number of times that a particular topic z is assigned to any tokens in the Gibbs Sampling process, N_z, to the number of tokens in the whole resource collection, N, as shown in Equation 2. Finally, each tag concept can be represented as a $|\mathbf{z}|$-dimensional vector; the sum of the entries (probabilities) in the vector equal to 1 (see Equation 3, where $v(C)$ is the representation of a tag concept in terms of probabilistic distributions of latent topics).

$$p(z|C_a) \propto p(C_a|z) * p(z) \tag{1}$$

$$p(z) = \frac{N_z}{N} \tag{2}$$

$$v(C_a) = \{p(\mathbf{z}_i|C_a)\}_{i=1}^{|\mathbf{z}|} \tag{3}$$

It was noted earlier that a tag concept is assumed to be potentially ambiguous and might have complex meanings. The proposed representation intuitively captures the different meanings of a tag concept implied by the latent topics. Since a tag concept is usually only related to several topics, we introduce the notion of a *significant topic set*, $\mathbf{z}_a^{sig}$, which includes the latent topics whose value is above p, for tag concept C_a (see Equation 4). We set $|\mathbf{z}|$ as 1000 based on model perplexity (see Section 6.1.2) and p as 0.1 in this study[5].

$$\mathbf{z}_a^{sig} = \{z \mid z \in \mathbf{z} \text{ and } p(z|C_a) \geqslant p\} \tag{4}$$

4. Feature Generation with probabilistic association analysis

This section presents the feature generation process used to quantify the degree that a concept is a hyponym of another given a context concept; and is based on the approach presented in [14]. The generated features form the input to the Classification and Testing module in the experiments presented in Section 6.3. We tested the usefulness of single rules for identifying subsumption relations in the literature [15] and found that the results were not satisfactory.

[5] The value of p $(= 0.1)$ is set empirically according to the distribution of $p(z|C_a)$ and the number of topics $|\mathbf{z}|$. For $|\mathbf{z}| = 1000$, the average $p(z|C_a)$ is 0.001, a very high p might produce no significant topics, while a very low p might include many irrelevant topics.

We believe that subsumption relations can be better established if we model the way humans understand and interpret the meaning of tags. Three assumptions are proposed based on how humans determine subsumption relations. For two tag concepts C_a and C_b to have a subsumption relation:

Assumption 1. *Topic similarity - they must be similar to each other to some extent.*

The topic similarity (or dissimilarity) is calculated in the low dimensional semantic space.

Assumption 2. *Topic distribution - they should have topic distributions satisfying conditions on both topic coverage and focus.*

Intuitively, a hypernym and its hyponym should have overlapping topics. In terms of topic coverage, a hypernym should have a distribution spanning over more significant topics or dimensions than the hyponym. In terms of focus, the hyponym tends to have a high probability on one, or a few, of the significant topics covered by the hypernym.

Assumption 3. *Probabilistic association - they should have a strong association to each other.*

Probabilistic association has its root in cognitive science and psychology [23, 25, 37]. It measures the degree of association between two concepts within a given context (e.g., parent of both concepts). In other words, it measures how likely that one is able to associate one concept given another and some background information. In our work, we quantify this likelihood using the conditional and joint probabilities of two tag concepts.

Based on the above assumptions, we generate three corresponding categories of features that together characterise the degree of subsumption between pairs of tag concepts, as shown in Table 2 below.

4.1. Topic similarity based features

Assumption 1 is translated into several topic based similarity and dissimilarity features. We use the Cosine similarity, Kullback-Leibler Divergence and

Table 2: Feature sets corresponding to the three assumptions

Features	Description
	Topic Similarity Based Features
Cos_sim	Cosine similarity of two topic distribution vectors
KL_Div1	Kullback-Leibler Divergence from C_a to C_b
KL_Div2	Kullback-Leibler Divergence from C_b to C_a
Gen_Jaccard	Generalised Jaccard Index of two topic distribution vectors
	Topic Distribution Based Features
overlapping	Number of overlapping significant topics
diff_num_sig	Difference of the number of significant topics
diff_max	Difference of the maximum elements in two tag vectors
diff_aver_sig	Difference of the average probability of significant topics
	Probabilistic Association Features
$p(C_a \mid C_b)$	Probabilistic association of C_a given C_b
$p(C_b \mid C_a)$	Probabilistic association of C_b given C_a
$p(C_a \mid C_b, R_{a,b})$	Local probabilistic association of C_a given C_b and $R_{a,b}$
$p(C_b \mid C_a, R_{a,b})$	Local probabilistic association of C_b given C_a and $R_{a,b}$
$p(C_a, C_b)$	Joint probabilistic association of C_a and C_b
$p(C_a, C_b \mid R_{a,b})$	Local joint probabilistic association of C_a and C_b given $R_{a,b}$

Generalised Jaccard Index.

Cosine similarity, denoted as *Cos_sim*, is one of the most common similarity measures used in Information Retrieval and Natural Language Processing. It is computed using the topic distribution vectors of two tag concepts. We use the Kullback-Leibler (KL) Divergence as defined in [47] to measure how far two tag concepts (or probability distributions) diverge. It can be interpreted as the amount of "surprise" arising from the difference between the true distribution and its approximation [47]. As it is an asymmetric measure, we generate two features, *KL_Div1* and *KL_Div2*. The generalised Jaccard index, *Gen_Jaccard*, is based on the intersection and union of topic sets, taking into consideration the magnitude of probability distributions.

4.2. Topic distribution based features

Assumption 2 is translated into the following features as shown in the second part of Table 2.

4.2.1. Number of overlapping significant topics

Having overlapping significant topics is a simple while important indication of a subsumption relation. It is denoted as *overlapping* in the equation below, where $\mathbf{z}_a^{\mathrm{sig}}$ and $\mathbf{z}_b^{\mathrm{sig}}$ can be obtained from Equation 4.

$$\text{overlapping}(C_a, C_b) = |\mathbf{z}_a^{\mathrm{sig}} \cap \mathbf{z}_b^{\mathrm{sig}}| \tag{5}$$

4.2.2. Difference of the number of significant topics

The number of significant topics is an indicator of how broad a tag concept is in terms of meanings. It is natural that general concepts tend to have more significant topics than specific ones. The difference of the number of significant topics between C_a and C_b is also used as a feature and is denoted as *diff_num_sig*.

$$\text{diff\_num\_sig}(C_a, C_b) = |\mathbf{z}_a^{sig}| - |\mathbf{z}_b^{sig}| \tag{6}$$

4.2.3. Difference of maximum probability in topic distributions

The difference of the maximum probabilities given the two topic distributions is defined in Equation 7. This feature works jointly with *overlapping* and the topic similarity based features: If C_a and C_b are similar and share some overlapping topics, a positive value of this feature, diff_max(C_a, C_b), may imply that C_a is more specific than C_b. The intuition is that the maximum probability of a hyponym on a topic should be higher than that of the hypernym. We denote this feature as *diff_max* in the equation below, where $\max(v(C))$ returns the maximum entry in the probability distribution.

$$\text{diff\_max}(C_a, C_b) = \max(v(C_a)) - \max(v(C_b)) \tag{7}$$

4.2.4. Difference of the average probability of significant topics

The feature *diff_max* only captures the difference of maximum probabilities and is not enough for concepts which have multiple significant topics. We add another feature, the difference of the average probability of significant topics between C_a and C_b. It is calculated using Equation 8 and denoted as *diff_aver_sig*.

$$\begin{aligned} \text{diff\_aver\_sig}(C_a, C_b) &= \text{Aver}(\mathbf{z}_a^{sig}) - \text{Aver}(\mathbf{z}_b^{sig}) \\ &= \frac{\sum(\mathbf{z}_a^{sig})}{|\mathbf{z}_a^{sig}|} - \frac{\sum(\mathbf{z}_b^{sig})}{|\mathbf{z}_b^{sig}|} \end{aligned} \tag{8}$$

If $|\mathbf{z}_a^{sig}|$ or $|\mathbf{z}_b^{sig}|$ is zero, we set its corresponding average probability $\text{Aver}(\mathbf{z}_a^{sig})$ or $\text{Aver}(\mathbf{z}_b^{sig})$ as zero.

4.3. Probabilistic association based features

The idea of probabilistic association among words is firstly proposed in [23, 25] and has its root in cognitive psychology [37]. It is believed that, in human memory, words have pre-existing associative structures constantly created from experiences [37]. With a probabilistic generative model, we can extract the gist of words and predict other associated ones based on bayesian inference [25]. We extend this idea and define new methods to quantify probabilistic associations among social tags under a given context.

The associative relations between words can be computed as a conditional probability of a response word given a cue word, marginalising over the hidden topics. While the *conditional probability* measures how likely one tag concept can be generated given another, the *joint probability* measures how likely two tag concepts can be generated together. In addition, we introduce a third tag as the context for the computation, which can be the root concept of the domain or sub-domain under consideration, or the direct parent concept of the “hypernym” of the tag pair. This allows us to learn a concept hierarchy from top to bottom (see Section 5). As these features are extracted with a local context, they are referred to as *local* probabilistic associations. The six features in this category are summarised in the third part of Table 2 and described below.

4.3.1. Probabilistic association

The probabilistic association between two tag concepts is defined as a conditional probability of one tag concept given another. The association is asymmetric and analogous to how we cognitively associate words [25]. The conditional probability $p(C_a|C_b)$ and $p(C_b|C_a)$ are computed by marginalising the inferred topics as shown in Equation 9.

$$\begin{aligned} p(C_a|C_b) &= \sum_{z \in \mathbf{z}} p(C_a|z, C_b) p(z|C_b) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) p(z|C_b) \end{aligned} \tag{9}$$

The $p(C_a|z)$ can be obtained from the LDA analysis, and $p(z|C_b)$ can be obtained using Equation 1. We adopt the assumption made in [25] that C_a and C_b are conditionally independent given the latent topic z. Similarly, we can compute $p(C_b|C_a)$.

4.3.2. Local probabilistic association

When constructing a hierarchy using a top down approach, a potential subsumption relation between two tag concepts should be considered with respect to their common parent. The parent tag concept represents the local context under consideration, which would facilitate disambiguating the meanings of the two tag concepts. To capture this idea, we propose the concept of local probabilistic association, which is computed conditioned on a context tag $R_{a,b}$. It is asymmetric and we define two feature extraction functions, $p(C_a|C_b, R_{a,b})$ and

$p(C_b|C_a, R_{a,b})$, as shown in Equation 10.

$$\begin{aligned} p(C_a|C_b, R_{a,b}) &= \sum_{z \in \mathbf{z}} p(C_a|z, C_b, R_{a,b}) p(z|C_b, R_{a,b}) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) p(z|C_b, R_{a,b}) \\ &= \sum_{z \in \mathbf{z}} p(C_a|z) \cdot \frac{p(C_b, R_{a,b}|z)p(z)}{p(C_b, R_{a,b})} \\ &= \sum_{z \in \mathbf{z}} \frac{p(C_a|z)p(C_b|z)p(R_{a,b}|z)p(z)}{p(C_b, R_{a,b})} \end{aligned} \tag{10}$$

Here we extend the assumption in [25] and assume that C_a, C_b and $R_{a,b}$ are conditionally independent given the latent topic z. The $p(C_a|z)$, $p(C_b|z)$, and $p(R_{a,b}|z)$ can be obtained from the LDA analysis; $p(z)$ is computed by using Equation 2 and $p(C_b, R_{a,b})$ is computed by using Equation 11 (see Section 4.3.3).

4.3.3. Joint probabilistic association

Tag concepts that have a direct subsumption relation fall into similar areas and should have a high likelihood of being jointly generated. Therefore, we define the joint probabilistic association, $p(C_a, C_b)$. It is symmetric and computed by using Equation 11, where $p(C_a|C_b)$ can be obtained using Equation 9.

$$\begin{aligned} p(C_a, C_b) &= p(C_a|C_b)p(C_b) \\ &= p(C_a|C_b) \sum_{z \in \mathbf{z}} p(C_b|z)p(z) \end{aligned} \tag{11}$$

4.3.4. Local joint probabilistic association

Similar to local probabilistic association, the local joint probabilistic association is further conditioned using a context tag $R_{a,b}$. It measures the likelihood of two tags being jointly generated with a particular context. It is also symmetric, denoted as $p(C_a, C_b|R_{a,b})$, where the $p(C_a|C_b, R_{a,b})$ and $p(C_b|R_{a,b})$ can be computed using Equations 9 and 10, respectively.

$$
\begin{aligned}
p(C_a, C_b | R_{a,b}) &= p(C_a | C_b, R_{a,b}) p(C_b | R_{a,b}) \\
&= \sum_{z \in \mathbf{z}} p(C_a | z, C_b, R_{a,b}) p(z | C_b, R_{a,b}) p(C_b | R_{a,b}) \\
&= \sum_{z \in \mathbf{z}} p(C_a | z) \cdot \frac{p(C_b, R_{a,b} | z) p(z)}{p(C_b, R_{a,b})} \cdot p(C_b | R_{a,b})
\end{aligned}
\tag{12}
$$

Once the three groups of features (14 features in total) are defined (see Table 2 for a summary), in the Classification and Testing module, we generate positive and negative instances, through tag grounding and instance labelling as described in Sections 6.2.1 and 6.2.2. Each instance is represented as a 14-dimensional feature vector. We create training, validation and testing datasets and feed the data into a classifier, which aims at learning a decision boundary in the feature space for binary prediction, i.e. whether the subsumption relation holds between a new ordered pair of tag concepts given a context tag concept. The selection of a classifier is independent from our approach. We will test and evaluate several mainstream of-the-shelf classifiers in Section 6.3.

5. Hierarchy generation algorithm

A hierarchy can be generated with an algorithm that organises tag concepts with valid subsumption relations from top to bottom, in an iterative manner. The algorithm starts with a specified “root” concept (a specific concept in a KB, which is designated by the users) and learns the layer below it. Then it learns the next layer from the current layer, and so on. The learned hierarchy is a Direct Acyclic Graph (DAG), where the nodes are tag concepts and edges represent subsumption relations among them.

A key step in this algorithm is to select candidate hyponyms for a concept under consideration and then pass them to the trained classifiers for prediction. To enhance the consistency of the hierarchy generation, during the candidate hyponym selection, the algorithm makes use of the context of a concept, which is defined as the direct hypernym of that concept if available, otherwise, it is defined as the specified root concept. The candidate hyponyms of a concept should be associated to the concept, the root, as well as the context. The candidate selection condition is therefore calculated by using the global and local probabilistic association, according to Equations 9 and 10. Let *cand* be a candidate hyponym, *root* be the user-specified root concept, *concept* be the concept under consideration for which the candidate hyponyms are to be selected, *context* be the direct hypernym of *concept*, and TH be a pre-defined threshold. If the following two conditions are met then *cand* is chosen as a candidate hyponym of *concept*: (1) $p(cand|root) > TH$, this means that all candidates should be associated to the specified root; and (2) $p(cand|concept, context) > TH$, this means that all candidates should be associated to the concept under consideration given the context[6]. The two probabilities can be calculated based on the Equations 9 and 10, respectively.

The notations used in Algorithm 1 are explained as follows.

- G_{layer} represents a layer in the learned hierarchy; it is initialised as the root layer.
- H is the hierarchy to be generated; it is initialised as $\emptyset$.
- $h(x_i, \Theta)$ is the classification function to predict if a subsumption relation holds between two tag concepts (see Section 6.3). Θ represents the learned weights in training the classifier; $x_i = f(I_i)$ is an instance which is represented as a vector of the extracted features; and f represents the feature extraction function defined in Section 4.
- L is the list of associated tag concepts to the user specified root, i.e., $L \leftarrow \{cand \mid p(cand|root) > TH\}$. All the candidate hyponyms will be selected from this list.

[6] TH is empirically set within $[\frac{1}{|C|}, \frac{10}{|C|}]$ for both conditions, where $|C|$ is the number of tag concepts. This is to ensure that TH is higher than the average probability while retaining a considerable number of candidates.

When selecting the candidate hyponyms for the *root*, as *context* is not available, only the condition (1) is used (see line 2 in Algorithm 1). From line 4 to line 13, the algorithm learns the layer below the root. If the layer is not the root layer, then there are possibly multiple concepts on that layer. From line 15 to line 26, for each of the concepts, the algorithm selects a number of candidates from the list L. Then the pairs of each of the candidates and the concept under consideration are passed to the classification function h for prediction. If a subsumption relation can be established, then the pair is added into the temporary layer G'_{next}. The layer may need to be pruned and then added into the hierarchy H (lines 27-29, detail of the pruning process is presented in Algorithm 2). Then the algorithm learns the next layers with recursive calls (lines 30-31).

To create a hierarchy as a Direct Acyclic Graph, it is necessary to prune edges to ensure that each node (except the root) has only one hypernym. Algorithm 2 prunes a weighted directed graph with possible cycles. The input is an intermediate layer, G'_{next}, in Algorithm 1 and the output is G_{next}. The idea is to select the hypernym with the highest confidence score from the classification. In line 2, the algorithm first sorts the edges by their weights (i.e., classification scores) in descending order. In lines 3-8, for each edge E_i, it retrieves the hyponym *hypo*, which is then inserted if there is no parent for *hypo* in the G_{next} layer (function hasParent$(hypo, G_{next})$ returns a boolean value).

The time-complexity of Algorithm 1 is $\mathcal{O}(d \cdot (l \cdot m \cdot c + m' \log m' + m'))$, where l is the number of possible candidate hyponyms; m and m' are the number of possible edges at the G_{layer} and G'_{next} respectively; d is the depth of the hierarchy H; and c is the time-complexity of the classifier function $h(x_i, \Theta)$. The graph pruning algorithm (Algorithm 2), which is a part of Algorithm 1, has time complexity $\mathcal{O}(m' \log m' + m')$. For most academic domains, the values of l, m, m', and d are limited; the time-complexity of the algorithm is dependent on the time-complexity c of the underlying classifier. Therefore, the algorithm is reasonably efficient.

Algorithm 1: generateHierarchy(G_{layer})

```
Require: $G_{layer}$, $H$, $L$, and $h$.
Ensure: $H$, hierarchy to be learned.
Initialise $G_{next} \leftarrow \emptyset$;
if $G_{layer}$ is the root layer then
    Add root to $H$;
    for each cand in $L$ do
        $context \leftarrow root$;
        $I_i \leftarrow <cand,\ root,\ context>$;
        $x_i \leftarrow f(I_i) = [f_1(I_i), f_2(I_i), ..., f_{14}(I_i)]$;
        Predict subsumption relation using $h(x_i, \Theta)$;
        if subsumption relation holds then
            $G_{next} \leftarrow G_{next} \cup\ < cand, root >$;
            Remove cand from $L$;
        end
    end
else
    for each edge $< concept, context >$ in $G_{layer}$ do
        $L_{sub} \leftarrow \{cand \mid p(cand|concept, context) > TH, cand \in L\}$;
        for each cand in $L_{sub}$ do
            $I_i \leftarrow <cand,\ concept,\ context>$;
            $x_i \leftarrow f(I_i) = [f_1(I_i), f_2(I_i), ..., f_{14}(I_i)]$;
            Predict subsumption relation using $h(x_i, \Theta)$;
            if subsumption relation holds then
                $G_{next} \leftarrow G_{next} \cup\ < cand, concept >$;
                Remove cand from $L$;
            end
        end
    end
    $G_{next} \leftarrow$ prune($G'_{next}$);
end
```

```
Add $G_{next}$ to H;
while not finished do
    generateHierarchy($G_{next}$)
```

Algorithm 2: prune(G'_{next})

```
Require: $G'_{next}$
Ensure: $G_{next}$, a pruned graph as a DAG.
Initialise $G_{next}$;
Sort all edges ($E < hypo, hyper >$) in $G'_{next}$ in descendant order by
  classification score;
for $i \leftarrow 1$ to $|E|$ do
    Retrieve the hypo from $E_i$;
    if NOT hasParent($hypo, G_{next}$) then
        $G_{next} \leftarrow G_{next} \cup E_i < hypo, hyper >$;
    end
end
```

6. Experiments and evaluation

We conducted experiments using three large-scale, publicly available KBs, DBpedia, Microsoft Concept Graph (MCG), and ACM Computing Classification System (CCS). The training and testing data were automatically created by grounding the tag concepts in these KBs. The results were compared to those produced by the state-of-the-art mechanisms and evaluated using three strategies: relation-level, ontology-level and knowledge base enrichment based evaluation. The implementation of the system and experiments are available on GitHub[7].

[7] `https://github.com/acadTags/tag-relation-learning/`

6.1. Social tagging data processing

We extracted a social tagging dataset from Bibsonomy, a well-known social bookmarking system for academic publications and Web links, maintained by the Knowledge and Data Engineering Group at the University of Kassel [5]. We used the whole dump of the Bibsonomy dataset (version "2015-07-01"), which can be downloaded after request[8]. The whole dataset contains 3,794,882 annotations, 868,015 distinct resources and 283,858 distinct tags contributed by 11,103 users, accumulated from 2005 to July 2015.

6.1.1. Data Cleaning

To create a cleaned folksonomy $\mathbb{F}^{clean}$, we performed pre-processing including: (1) special character handling, for example, tags having colons (:) and underscores (_); (2) multi-word and single-word tag extraction, we paid extra attention to multi-word tags such as "Natural_Language_Processing" and "Social_Semantic_Web". We grouped different forms of multi-word and single-word tags and chose a standard form for them (referred to as a tag concept). In this way, we created tag groups within which tags refer to the same concept; (3) tag filtering by metrics and languages; for example, we filtered out insignificant tags and only kept multi-word and single-word tag groups which have been used by no less than four distinct users. Also we only kept English tags based on the automatic detection results obtained using the Google Translation API[9]. For a more detailed description of the Data Cleaning steps, see [16]. We also removed resources that are not academic papers and have less than three tag concepts. Finally, we obtained a cleaned folksonomy of higher quality, $\mathbb{F}^{clean}$, with 7,458 tag concepts and 128,782 publications.

6.1.2. Probabilistic topic analysis from tagging data

Each resource was treated as a "bag of tags". Probabilistic topic analysis was performed with LDA and Gibbs Sampling by using the MALLET Machine

[8] `https://www.kde.cs.uni-kassel.de/bibsonomy/dumps/`

[9] `https://cloud.google.com/translate/`

Table 3: Example latent topics related to the tag concept "web"

Topic ID	Most probable 5 tag concepts
14	web accessibility centre mobility human
17	web mining web_mining data_mining data_web
126	web social social_web science web_science
247	semantic_web web semantic ontology rdf
333	application web web_application ajax web_interfaces
466	service web_service web composition service_composition
576	search web web_search social_search social_web
577	web archive crawl alexandria l3s

Learning Library[10]. The two concentration parameters for the dirichlet distribution were set empirically: topic-word hyperparameter $\alpha = 50/|\mathbf{z}|$; and the document-topic hyperparameter $\beta = 0.01$. We held out 10% of the data to optimise the number of topics $|\mathbf{z}|$ with minimum perplexity and set $|\mathbf{z}|$ as 1000. We then used this probabilistic representation to extract features for learning.

Table 3 provides an example on the learned topics, each of which is represented as a probabilistic distribution of tags. Only the five tag concepts with the highest probabilities in the distribution $p(C|z)$ are shown. It can be seen that collectively the tag concepts provide an intuitive definition on the meanings of the hidden topics. From a different perspective, probabilistic topic modelling is also an effective dimensionality reduction technique which transforms the original resource representation from a "bag of tags" to a vector of latent topics in a lower semantic space. A tag concept may relate to multiple topics, for example, the tag "web" is related to topics 14 (human accessibility), 17 (data mining), 126 (social Web) and 247 (semantic Web), 333 (Web applications), 466 (Web service), 576 (Web search), 577 (Web archiving and crawling). Tag concepts such as "web" contribute to multiple topics and are potentially general concepts. Then, we represent each tag as a distribution of the topics from $p(C|z)$ and $p(z)$, according to the Equations 1-3.

[10] http://mallet.cs.umass.edu/

6.2. Labelled dataset creation

To learn subsumption relations, we need to generate labelled training and testing data. Selected tag pairs from the Bibsonomy dataset were automatically grounded to those in KBs and then labelled as either positive (subsumption) or negative.

6.2.1. Tag grounding

Three external KBs were leveraged: (1) **DBpedia** contains structured information of Wikipedia, described in RDF (Resource Description Framework). We used the DBpedia "2015-10" version[11], to be consistent with the Bibsonomy dataset (2015 version). According to the ontological structure of DBpedia[12], we extracted concepts with subsumption relations using the *skos:broader* predicate and we used the *dbo:wikiPageRedirects* predicate to extract equivalent concepts to increase the recall of string matching; (2) **Microsoft Concept Graph (MCG)**[13] is a data-driven KB mined from billions of Web pages, released in September 2016, consisting of 85 million "is-a" relations and 18 million concepts. Each "is-a" relation is associated with a strength value. We selected the strength no less than 5, which resulted in 2.8 million relations; and (3) **ACM Computing Classification System (CCS) version 2012**[14] is an academic classification system that has been used to organise and retrieve publications by subjects in the ACM Digital Library. From the RDF version of CCS, we treated *skos:broader* relations as subsumption relations and *skos:altLabel* as equivalent relations.

Table 4 provides some statistics concerning the overlapping between external KBs and Bibsonomy. DBpedia had 2,191 common concepts with Bibsonomy and CCS had 691. The number is not excessive, suggesting that social tags can be

[11] `http://downloads.dbpedia.org/2015-10/`

[12] For an example see the DBpedia Category, Machine Learning, `http://dbpedia.org/page/Category:Machine_learning`.

[13] `https://concept.research.microsoft.com/Home/Download`

[14] `https://www.acm.org/publications/class-2012`

Table 4: Statistics of the external knowledge bases and the Bibsonomy folksonomy

	Concepts	Subsumption relations	Concept overlap with Bibsonomy	Release Date
DBpedia	1,316,674	2,706,685	2,191	2015-10
MCG	1,483,135	2,844,951	6,030	2016-09
CCS	9,060	2,390	691	2012 (latest version)
Bibsonomy	7,458	-	-	2015-07

potentially used to enrich human-engineered KBs. The number of overlapped concepts between MCG and Bibsonomy is 6,030, suggesting that there is still room to enrich the KB even though MCG is created from billions of Web pages.

6.2.2. Instance labelling with knowledge bases

We extend the instance labelling method in [14] to generate training and testing data in full domains from all three KBs. For each KB, we created directed pairs of the overlapped tags concepts $< C_a, C_b >$, and labelled them. We used simple string matching, based on Levenshtein distance, to map a cleaned tag to a concept in the external KB. Then, a tag pair instance can be labelled as positive if there is an asserted, direct subsumption relation between the two tags in the external KB, and the probabilistic association between them, $p(C_a|C_b) > TH$, computed using Equation 9. This is to ensure the labelled instances are consistent with both the external KBs and Bibsonomy dataset. We created the negative instances by using the following methods: (i) reversed negative, for each positive pair $< C_a, C_b >$, we created a negative pair $< C_b, C_a >$; and (ii) random negative, if both randomly generated tag concepts appear in one of the KBs, but a subsumption relation between them cannot be found in any of the three KBs, then we label the instance as negative. We also extracted the context tags for these instances to facilitate probabilistic association analysis. Finally, we obtained 4,965 positive instances and 9,570 negative instances (including 4,785 reversed negative instances and 4,785 random negative instances). In total there are 14,535 instances and the ratio of positive to negative instances is around 1 : 1.93.

It should be noted that the instance labelling process is based on the assumption that all relations in KBs are correct. In reality, the positive instances may suffer the quality issues of the KBs, due to the nature of the collaboratively generated data. Similarly, the random negative instances, according to the open-world assumption, may not necessarily be negative if they are not contained in any of the KBs. Nevertheless, the quality of these KBs is improving over time with the efforts of millions of individuals.

6.3. Classification settings

Using the data created above, we generated features for each instance with the method proposed in Section 4 and fed them into different classifiers. We held out 20% of all instances for testing and used the remaining 80% for training. 10-fold cross-validation was used to tune the parameters and validate the generalisation of the trained models. We used the standard precision, recall and F-measure to evaluate the performance of the classifiers.

To test the effectiveness of the methods, we adopted four popular classification algorithms, namely, Support Vector Machine (SVM), AdaBoost, Logistic Regression and the CART algorithm (Classification And Regression Trees). Support Vector Machine (SVM) searches for a hyperplane which separates two classes with the maximum margin. We used the radial basis function (RBF) kernel which outperformed others kernels in our experiments. AdaBoost is a typical boosting algorithm for ensemble learning, which provides a structure to improve performance by aggregating the prediction of multiple weak classifiers. We used decision trees as weak classifiers to train Adaboost. Logistic Regression is a generalised regression model for categorical values adapted from linear regression. CART is a decision tree learning algorithm that searches for a hierarchical structure to classify data. As each of the classification algorithms has its own characteristics and constraints [44], the evaluation was based on results from a group of classifiers, instead of any single classifier.

We used the LibSVM 3.22[15] [10] Matlab version for SVM training. The RBF kernel with grid-search was adopted to tune the two parameters c and γ to optimise the F_1 score, as suggested in [28]. The remaining three algorithms (CART, Logistic Regression and AdaBoost) were implemented in the Classification Learner App[16] in Matlab. We set the number of weaker learners as 30 and each of them used the same settings as the CART algorithm, and a shrinkage learning rate was set to 0.1 to prevent overfitting. All algorithms were validated using 10-fold cross-validation.

6.4. Evaluation

Three strategies were used for the evaluation: (i) relation-level evaluation using the testing set; (ii) ontology-level evaluation using external KBs as the gold standard; and (iii) knowledge base enrichment based evaluation through human assessment. The results allowed us to see to what extent social media data can be exploited to enrich existing KBs.

6.4.1. Relation-level evaluation

We compared the performance of the proposed method to several representative studies as explained in the following. The feature set proposed in this work is denoted as FS_{all}, which consists of features related to topic similarity ($FS_{topicSim}$), topic distribution ($FS_{topicDist}$) and probabilistic association ($FS_{probAsso}$) (see the whole three feature sets in Table 2).

1. Binary classification using co-occurrence related features [40]: Combining several heuristics as features in previous studies, i.e., support and confidence [41], cosine similarity of tag-tag vector [29, p. 56-p. 59], set inclusion and generalisation degrees [35], mutual overlapping [9] and graph-based taxonomy search [26]. In total there are 8 features and the feature set is denoted as FS_{co}.

[15] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[16] `https://cn.mathworks.com/help/stats/classification-learner-app.html`

2. The method in [47] based on *Information Theory Principle for Concept Relationship*: This proposed two conditions to measure the degree of subsumption between two concepts. The first condition is the *similarity condition*, measuring the similarity between two concepts; the second condition is the *divergence difference condition*, which calculates the difference between the Kullback-Leibler divergence of two tag concepts. This is generally equivalent to the topic similarity based feature set in our method. It contains 4 features, denoted as $FS_{topicSim}$.
3. The topic distribution related features, $FS_{topicDist}$: To allow performance comparison with using only the topic distribution.
4. The probabilistic association features, $FS_{probAsso}$: To allow performance comparison with using only the probabilistic association.
5. Combining both the co-occurrence related features [40] and the feature sets proposed in this study: To determine if the performance of the proposed method can be further improved by combining the co-occurrence based features. In total there are 22 features, denoted as FS_{all}+FS_{co}.

The results are presented in Table 5. In general, using the feature sets FS_{all} achieved higher F_1 scores with a large margin than using any others, and the best ranking (ranked first with SVM and Adaboost and second with LR and CART). The performance was stable and consistent with different classification techniques, showing the robustness of the proposed feature set in characterising subsumption relations. Co-occurrence based features (FS_{co}), which have achieved impressive results for supervised learning, as reported in the study presented in [40], did not perform well for the relation learning problem with our large labelled dataset in the academic domain. F_1 scores obtained using the co-occurrence based features (FS_{co}) [40] were much lower compared to FS_{all} (absolutely lower by 6.86% with SVM and by 18.13% with AdaBoost). Adding them to the proposed features sets (FS_{all}+FS_{co}) did not improve performance.

We also compared the proposed method to our previous work in [47], which applied probabilistic topic analysis on a collection of scientific publication ab-

Table 5: Classification testing results with comparison among feature sets

Feature set	Classifier	Recall	Precision	F_1 Score
Full features in our approach, FS_{all}	SVM RBF ($2^{10.5}$,$2^{4.5}$)	51.56% (1)	52.95% (3)	52.25% (1)
	AdaBoost	50.15% (1)	63.52% (3)	**56.05%** (1)
	LR	34.04% (2)	65.00% (2)	44.68% (2)
	CART	45.02% (3)	62.87% (2)	52.46% (2)
Rêgo *et al.* [40] (co-occurrence related features FS_{co}, including [41, 29, 35, 9, 26])	SVM RBF (2^{10},2^{7})	36.96% (5)	58.81% (2)	**45.39%** (4)
	AdaBoost	27.49% (4)	61.07% (4)	37.92% (4)
	LR	19.64% (3)	56.20% (4)	29.10% (3)
	CART	27.19% (4)	58.95% (4)	37.22% (4)
Wang *et al.* [47] (based on $FS_{topicSim}$)	SVM RBF ($2^{10.5}$,2^{9})	46.02% (3)	47.02% (5)	**46.51%** (3)
	AdaBoost	17.52% (5)	59.59% (5)	27.08% (5)
	LR	15.01% (4)	54.78% (6)	23.56% (4)
	CART	11.78% (5)	66.10% (1)	20.00% (5)
Topic distribution, $FS_{topicDist}$	SVM RBF (2^{10},2^{11})	40.28% (4)	46.14% (6)	**43.01%** (5)
	AdaBoost	11.48% (6)	59.07% (6)	19.22% (6)
	LR	10.27% (6)	55.14% (5)	17.32% (6)
	CART	3.02% (6)	47.62% (6)	5.68% (6)
Probabilistic association, $FS_{probAsso}$	SVM RBF (2^{12},$2^{8.5}$)	27.80% (6)	60.53% (1)	38.10% (6)
	AdaBoost	44.51% (3)	63.60% (2)	52.37% (3)
	LR	14.20% (5)	68.12% (1)	23.50% (5)
	CART	53.07% (1)	60.09% (3)	**56.36%** (1)
Combining full features with co-occurrence features in [40], FS_{all}+FS_{co}	SVM RBF ($2^{9.5}$,2^{4})	49.25% (2)	52.41% (4)	50.78% (2)
	AdaBoost	46.32% (2)	65.25% (1)	**54.18%** (2)
	LR	36.56% (1)	62.69% (3)	46.18% (1)
	CART	46.73% (2)	57.35% (5)	51.50% (3)

The values (2^a,2^b) after SVM RBF are the parameters c and γ tuned to optimise F_1 score. The highest F_1 Score for each feature set is bolded. The number in brackets shows ranking of the feature set under the same classifier.

stracts and then detected subsumption relations with the *Information Theory Principle for Concept Relationship*. It is comparable to the supervised learning method only using the topic similarity features ($FS_{topicDist}$). The proposed feature set (FS_{all}) performed generally better in terms of all metrics (in terms of F_1, an absolute increase by 5.74% with SVM and by 28.97% with AdaBoost). One of the main reasons is that the dataset used in [47] contains texts and rich contextual information, which is not the case for social tagging data.

When using single feature sets we found that the proposed probabilistic association features ($FS_{probAsso}$) generated higher precision (overall best ranking), while the recall was lower than others. In most classifier settings, the best F_1

score was achieved by using the full feature sets FS_{all}. This confirms the hypothesis that we can better characterise subsumption relations through all the feature sets founded on the three assumptions. We noticed that the classification with $FS_{probAsso}$ and CART obtained a slightly higher F_1 score (+0.3%) than FS_{all} and Adaboost (56.34% vs. 56.05%), with the former having higher recall (+2.92%) but lower precision (-3.43%). The performance with CART was, however, not consistent with other classifiers and the overall ranking of the $FS_{probAsso}$ was worse than FS_{all}. This is probably because the individual features in $FS_{probAsso}$ can better satisfy the impurity criteria and are suitable for the rectilinear decision boundaries of the CART algorithm [44, p. 143-p. 147], while the other features which have strong interactions among them, especially those in $FS_{topicDist}$ (only 5.68% F_1 with CART but 43.01% with SVM), are more suitable for models with nonlinear boundaries and better generalisation capabilities. SVM and AdaBoost performed generally better than Logistic Regression (LR) and CART within each feature set. It is also noticed that, compared to the other 3 classifiers, training the SVM models with grid search to find the best parameters is computationally expensive, e.g., with best c values varying from $2^{9.5}$ to 2^{12} and γ values from 2^4 to 2^{11} as shown in Table 5.

6.4.2. Ontology-level evaluation

The ontology-level evaluation was designed to measure the quality of the hierarchies or ontologies derived using the hierarchy generation algorithm. We used a reference-based strategy adopted from the study in [42]. The prerequisite of this strategy is the existence of a “gold-standard” ontology to be compared against. The quality of the learned hierarchies is thus measured as the similarity to the “gold standard”. To ensure the reproducibility of the evaluation, we chose the popular KBs, DBpedia and CCS as the “gold standard” and aimed to test the capabilities of classifiers and the algorithm for generating hierarchies, although we are aware of the fact that both KBs are not perfect and the CCS has been relatively static (last updated 7 years ago at the time of reporting this work). The data-driven knowledge base MCG is not chosen as a “gold standard", because the transitivity of subsumption relations in MCG (which is an acyclic graph and suffers from semantic drift) is low [32].

We adopted the standard metrics for reference-based evaluation, *taxonomic precision* (TP), *taxonomic recall* (TR), *taxonomic F-measure* (TF) [12] and *taxonomic overlapping* (TO) [34], also applied in [42]. The idea is to find a common concept C_c between a learned hierarchy L and a referenced hierarchy G, and to generate a characteristic extract from each of them, $\mathrm{ce}(C_c, L)$ and $\mathrm{ce}(C_c, G)$. The partial similarity of the two extracts regarding the common concept C_c is then calculated. The local taxonomic precision and recall regarding the common concept C_c can be calculated using Equations 13 and 14.

$$tp(C_c, L, G) = \frac{|\mathrm{ce}(C_c, L) \cap \mathrm{ce}(C_c, G)|}{|\mathrm{ce}(C_c, L)|} \tag{13}$$

$$tr(C_c, L, G) = \frac{|\mathrm{ce}(C_c, L) \cap \mathrm{ce}(C_c, G)|}{|\mathrm{ce}(C_c, G)|} \tag{14}$$

The global taxonomic precision $TP(L, G)$ and recall $TR(L, G)$ are computed by averaging all local tp and tr with respect to all common concepts. The taxonomic F-measure is the harmonic mean of both taxonomic precision and recall.

Taxonomic overlapping is symmetric and can be used independently. The local version is defined as follows and the global version $TO(L, G)$ is computed by averaging all the local ones.

$$to_{\mathrm{ce}}(c, L, G) = \frac{|\mathrm{ce}(c, L, G) \cap \mathrm{ce}(c, G, L)|}{|\mathrm{ce}(c, L, G)| \cup \mathrm{ce}(c, G, L)|} \tag{15}$$

We used several domains for ontology-level evaluation. For DBpedia, concepts matched to those within the top 5 layers under the categories "Areas_of_computer_science" and "Information_science" were selected (the domain is denoted as "CS/IS"). For the domains of "Education" and "Economics", concepts within the top 3 layers were selected. For CCS, all tag concepts matched to the uppermost 2, 3 or 4 layers were selected. We finally obtained 217 tag concepts in CS/IS, 226 in Education and 152 in Economics in DBpedia, and 43, 113, 133 tag concepts

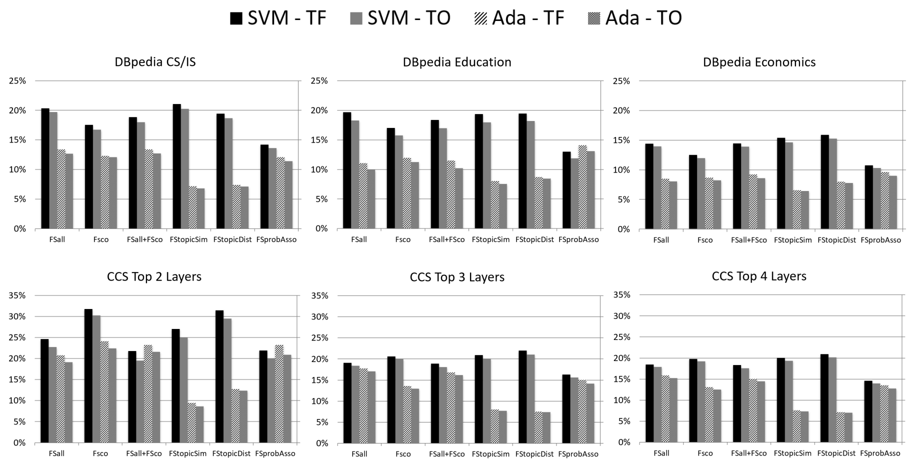


Figure 2: Results of ontology-level evaluation. The figures show the TF and TO values computed with the learned hierarchies from Bibsonomy and the "gold standard" (DBpedia and CCS). Three domains were selected for DBpedia, Computer Science/Information Science, Education and Economics; and three sub-hierarchies uppermost 2, 3 and 4 layers were tested for CCS. SVM or AdaBoost (denoted as "Ada") were used for classification. The x-axis represents methods with different feature sets and the y-axis represents the similarity in percentage. Higher TF and TO values indicate greater similarity to the gold-standard.

matched to the uppermost 2, 3, 4 layers of CCS, respectively. For each tag concept in the selected domain, we generated a sub-hierarchy using the hierarchy generation algorithm and calculated TP, TR, TF and TO (averaged results over the sub-hierarchies for each domain are reported in Figure 2). We believe this novel evaluation process on multiple hierarchies is more rational than on only one global hierarchy against the KBs. The latter approach may be biased as it does not test the similarity of the branches between two hierarchies [42].

Figure 2 shows the results obtained with different combinations of KBs, features sets and classifiers. The results demonstrate satisfying description ability of the proposed feature sets with the hierarchy generation algorithm, with generally better and more consistent results compare to other feature sets. The TF and TO values are also consistent with those reported in the previous study [6]. In all three domains of DBpedia, the TF and TO scores generated with the proposed features FS_{all} were generally higher than those generated with other features sets based on co-occurrence, topic similarity, topic distribution and probabilistic association. There were few exceptions, however, their performance was highly inconsistent between classifiers, e.g. the topic similarity features $FS_{topicSim}$ had higher TF than FS_{all} for CS/IS using SVM, but much lower TF using Adaboost. For CCS with only 2 uppermost layers, the highest TF and TO scores were obtained with the co-occurrence based features, but the proposed feature set performed generally better with concepts matched to 3 and 4 uppermost layers. This shows the advantage of the proposed feature set on generating hierarchies with more *specific* concepts than the co-occurrence based features. Furthermore, results of the proposed feature set with CCS were also consistent between classifiers. Similar to the results in the relation-level evaluation, the performance of using only the topic similarity or topic distribution based features varied significantly with different classification techniques in all settings.

6.4.3. Knowledge base enrichment based evaluation

One particularly interesting part of this research is to discover previously unseen knowledge or emerging semantics from social tagging data. The enrichment-based evaluation is to assess to what extent the method can enrich external KBs with new and meaningful concepts and relations. For this purpose domain experts were used for manual assessment.

We selected a number of concepts from DBpedia and CCS and used the trained classification models to predict their direct hyponyms. Then we identified new hyponyms which do not appear in the "gold-standard" KBs and let the human experts make judgement about their validity. A large number of direct subsumption relations was generated and around 99% of them were not present in the KBs. In total, there were 3,846 distinct new relations for DBpedia, and 1,302 for CCS.

As the number of enriched relations is large, we only selected a subset (298 out of 5,148) for manual assessment. Thirteen domain experts, including four academic staff members and nine senior PhD candidates, from universities in the UK and the US, participated in the evaluation. They work in different areas of computer or information sciences. In the evaluation sheet, we asked them to mark the predicted relations with one of the four options: (1) subsumption: C_a is a narrower concept of C_b given C_r; (2) Semantically related: C_a is not a narrower concept of C_b, but they are highly related; (3) Unrelated; and (4) Not sure.

Using the proposed method with SVM and AdaBoost, we generated two sets of subsumption relations for DBpedia and CCS respectively. We merged the results in the evaluation sheet and ended up with 298 distinct relations after filtering out those with low confidence scores. The multi-rater Fless Kappa [19] was 0.15 and free-marginal kappa [39] was 0.22 among the domain experts, showing a "slight" agreement. This is also consistent with the results reported in previous studies, e.g., Fless Kappa 0.137 in [22] and free-marginal kappa 0.139 in [43]. This "slight" agreement is because that the learned relations and concepts concern very specific sub-areas and rare topics, thus some of them (especially abbreviations) may not be familiar to all participants.

Among the 3,874 ratings (298 × 13) presented to the judges, 1,489 of them (38.44%) were marked as "subsumption", and 1,131 (29.20%) were "related". We further compared the enrichment accuracy in terms of KBs. The ground truth was determined by assuming no less than a certain number of votes were for "subsumption" and the accuracy was computed with respect to the ground truth. As shown in Figure 3, the x-axis represents settings for the classifiers and KBs, and the y-axis represents the accuracy of the enriched relations. If we define a predicted relation as a true subsumption when at least five domain experts have the agreement, then the overall accuracy of the enriched relations was 53.36%. The accuracy increased to 66.44% and 74.50% if we only need agreement from four and three domain experts respectively; the accuracy decreased to 28.52% when we need agreement from seven of them. Higher accuracy was seen in most cases when enriching CCS than DBpedia. The reason might be that the selected concepts in CCS are more general and the hierarchy is more shallow than those of DBpedia. Therefore, there is much room for new relations

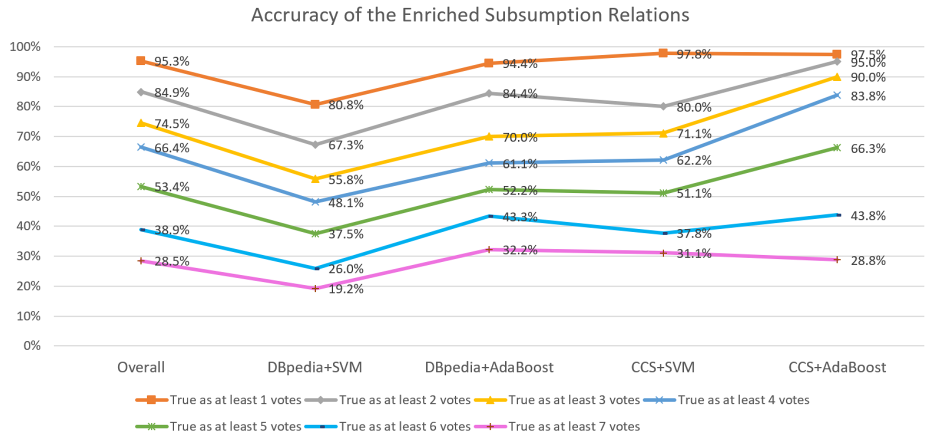


Figure 3: Results on knowledge base enrichment based evaluation

and concepts in CCS. The results clearly show that the proposed method can help discover meaningful knowledge from noisy tagging data[17].

7. Conclusion and future work

Harvesting the "collective intelligence" from social media data has been a promising direction for knowledge discovery. Along this line we show a method for enriching KBs with academic tagging data. The novelty of the method lies in the supervised learning framework with training data automatically extracted through probabilistic association analysis. We also carried out a comprehensive evaluation towards the quality of the discovery knowledge using three different strategies: relation-level evaluation, ontology-level evaluation and enrichment-based manual evaluation. To our best knowledge, this is one of the most comprehensive evaluation studies using large, publicly available, datasets and knowledge bases, especially for knowledge base enrichment. We recognised the fact that social tagging data is extremely noisy and of low quality and did not expect that all the learned knowledge would be meaningful and useful. This is

[17] The evaluation sheet and the ratings from the domain experts are available on `https://github.com/acadTags/tag-relation-learning`

confirmed by the evaluation results, while the discovered new knowledge can be used to enrich KBs, it needs scrutiny of domain experts.

With the recent rise of deep learning for language processing, one of the aims of future work is to apply deep learning models to improve the quality of the discovered knowledge. For example, it is possible to combine or align probabilistic topic representations with deep distributional representations of tags. Another area for future work is to adapt the current supervised learning method to an online learning framework in order to build evolving knowledge structures. In this way, the learned hierarchy can update itself with the availability of new tagging data taking into consideration temporal factors. The design would also help capture emerging semantics more timely.

Acknowledgement

We would like to thank the domain experts for their help with the evaluation and the anonymous reviewers for their constructive feedback. This research is funded by the Research Development Fund at Xi'an Jiaotong-Liverpool University (XJTLU), contract number RDF-14-01-10 and partially supported by the following: The National Natural Science Foundation of China under no. 61876155; Suzhou Science and Technology Program under no. SZS201613; Natural Science Foundation of Jiangsu Province BK20181189; Key Program Special Fund in XJTLU under no. KSF-A-01, KSF-T-06, KSF-E-26, KSF-P-02, and KSF-A-10.

References

[1] Alruqimi, M., & Aknin, N. (2019). Bridging the gap between the social and semantic web: Extracting domain-specific ontology from folksonomy. *Journal of King Saud University - Computer and Info. Sci.*, *31*, 15 − 21.

[2] Alves, H., & Santanch, A. (2013). Folksonomized ontology and the 3e steps technique to support ontology evolvement. *Web Semantics: Science, Services and Agents on the World Wide Web*, *18*, 19 − 30.

[3] Andrews, P., Pane, J., & Zaihrayeu, I. (2011). Semantic disambiguation in folksonomy: A case study. In *Advanced Language Technologies for Digital Libraries: International Workshops on NLP4DL 2009, Viareggio, Italy, June 15, 2009 and AT4DL 2009, Trento, Italy, September 8, 2009* (pp. 114–134). Berlin, Heidelberg: Springer Berlin Heidelberg.

[4] Angeletou, S. (2008). Semantic enrichment of folksonomy tagspaces. In *The Semantic Web - ISWC 2008: 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings* (pp. 889–894). Berlin, Heidelberg: Springer Berlin Heidelberg.

[5] Benz, D., Hotho, A., Jäschke, R., Krause, B., Mitzlaff, F., Schmitz, C., & Stumme, G. (2010). The social bookmark and publication management system bibsonomy. *The VLDB Journal*, *19*, 849–875.

[6] Benz, D., Hotho, A., Stumme, G., & Sttzer, S. (2010). Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge. In *Proc. of the 2nd Web Science Conf. (WebSci10)* (pp. 1–8).

[7] Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, *55*, 77–84.

[8] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*, 993–1022.

[9] Cai, S., Sun, H., Gu, S., & Ming, Z. (2011). Learning concept hierarchy from folksonomy. In *2011 Eighth Web Information Systems and Applications Conference* (pp. 47–51).

[10] Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27.

[11] Chen, J., Feng, S., & Liu, J. (2014). Topic sense induction from social tags based on non-negative matrix factorization. *Information Sciences*, *280*, 16–25.

[12] Dellschaft, K., & Staab, S. (2006). On how to perform a gold standard based evaluation of ontology learning. In *5th Int. Semantic Web Conf.* (pp. 228–241). Springer Berlin Heidelberg.

[13] Djuana, E., Xu, Y., & Li, Y. (2011). Constructing tag ontology from folksonomy based on wordnet. In *Proceedings of the IADIS International Conference on Internet Technologies and Society 2011* (pp. 1–8). International Association for Development of the Information Society (IADIS).

[14] Dong, H., Wang, W., & Coenen, F. (2018). Learning relations from social tagging data. In X. Geng, & B.-H. Kang (Eds.), *PRICAI 2018: Trends in Artificial Intelligence* (pp. 29–41). Cham: Springer International Publishing.

[15] Dong, H., Wang, W., & Coenen, F. (2018). Rules for inducing hierarchies fromsocial tagging data. In *Transforming Digital Worlds* (pp. 345–355). Cham: Springer International Publishing.

[16] Dong, H., Wang, W., & Frans, C. (2017). Deriving dynamic knowledge from academic social tagging data: a novel research direction. In *iConference 2017 Proceedings*. iSchools.

[17] Dong, H., Wang, W., & Liang, H. N. (2015). Learning structured knowledge from social tagging data: A critical review of methods and techniques. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)* (pp. 307–314).

[18] Du, H., Chu, S. K., & Lam, F. T. (2009). Social bookmarking and tagging behavior: an empirical analysis on delicious and connotea. In *Proceedings of the 2009 International Conference on Knowledge Management* (pp. 1–11).

[19] Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, *76*, 378.

[20] Gaevi, D., Zouaq, A., Torniai, C., Jovanovi, J., & Hatala, M. (2011). An approach to folksonomy-based ontology maintenance for learning environments. *IEEE Transactions on Learning Technologies*, *4*, 301–314.

[21] García-Silva, A., Corcho, O., Alani, H., & Gómez-Pérez, A. (2012). Review of the state of the art: discovering and associating semantics to tags in folksonomies. *The Knowledge Engineering Review*, *27*, 57–85.

[22] García-Silva, A., García-Castro, L. J., García, A., & Corcho, O. (2014). Social tags and linked data for ontology development: A case study in the financial domain. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)* (pp. 32:1–32:10). ACM.

[23] Griffiths, T. L., & Steyvers, M. (2003). Prediction and semantic association. *Advances in neural information processing systems*, (pp. 11–18).

[24] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*, 5228–5235.

[25] Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological review*, *114*, 211.

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

[27] Hotho, A., Jäschke, R., Schmitz, C., & Stumme, G. (2006). Information retrieval in folksonomies: Search and ranking. In Y. Sure, & J. Domingue (Eds.), *The Semantic Web: Research and Applications* (pp. 411–426). Berlin, Heidelberg: Springer Berlin Heidelberg.

[28] Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A practical guide to support vector classification*. Technical Report Dept. of Computer Science, National Taiwan University.

[29] Keller, C. (2010). *Theoretical and Practical Perspectives on Ontology Learning from Folksonomies*. Ph.D. thesis Universität Stuttgart.

[30] Lau, R. Y. K., Zhao, J. L., Zhang, W., Cai, Y., & Ngai, E. W. T. (2015). Learning context-sensitive domain ontologies from folksonomies: A cognitively motivated method. *INFORMS Journal on Computing*, *27*, 561–578.

[31] Lee, D. H., & Schleyer, T. (2012). Social tagging is no substitute for controlled indexing: A comparison of medical subject headings and citeulike tags assigned to 231,388 papers. *Journal of the American Society for Information Science and Technology*, *63*, 1747–1757.

[32] Liang, J., Zhang, Y., Xiao, Y., Wang, H., Wang, W., & Zhu, P. (2017). On the transitivity of hypernym-hyponym relations in data-driven lexical taxonomies. In *AAAI* (pp. 1185–1191).

[33] Lu, C., ran Park, J., & Hu, X. (2010). User tags versus expert-assigned subject terms: A comparison of librarything tags and library of congress subject headings. *Journal of Information Science*, *36*, 763–779.

[34] Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. In *13th Int. Conf. EKAW 2002 Proc.* (pp. 251–263). Springer Berlin Heidelberg.

[35] Meo, P. D., Quattrone, G., & Ursino, D. (2009). Exploitation of semantic relationships and hierarchical data structures to support a user in his annotation and browsing activities in folksonomies. *Information Systems*, *34*, 511–535.

[36] Mika, P. (2007). Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*, 5–15.

[37] Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (2004). The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, *36*, 402–407.

[38] Peters, I., & Becker, P. (2009). *Folksonomies: Indexing and Retrieval in Web 2.0*. De Gruyter/Saur.

[39] Randolph, J. J. (2005). Free-marginal multirater kappa (multirater κfree): An alternative to fleiss fixed-marginal multirater kappa. In *the Joensuu Learning and Instruction Symposium*.

[40] Rêgo, A. S. C., Marinho, L. B., & Pires, C. E. S. (2015). A supervised learning approach to detect subsumption relations between tags in folksonomies. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (SAC '15)* (pp. 409–415). ACM.

[41] Schmitz, C., Hotho, A., Jäschke, R., & Stumme, G. (2006). Mining association rules in folksonomies. In *Data Science and Classification* (pp. 261–270). Springer.

[42] Strohmaier, M., Helic, D., Benz, D., Körner, C., & Kern, R. (2012). E-valuation of folksonomy induction algorithms. *ACM Trans. Intell. Syst. Technol.*, *3*, 74:1–74:22.

[43] Syn, S. Y., & Spring, M. B. (2013). Finding subject terms for classificatory metadata from user-generated social tags. *Journal of the Association for Information Science and Technology*, *64*, 964–980.

[44] Tan, P., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to Data Mining*. (2nd ed.). Pearson.

[45] Tang, J., Leung, H.-f., Luo, Q., Chen, D., & Gong, J. (2009). Towards ontology learning from folksonomies. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence* IJCAI'09 (pp. 2089–2094). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

[46] Wal, T. V. (2007). Folksonomy. `http://vanderwal.net/folksonomy.html`. [Online; accessed 10-July-2018].

[47] Wang, W., Barnaghi, P. M., & Bargiela, A. (2010). Probabilistic topic models for learning terminological ontologies. *IEEE Transactions on Knowledge and Data Engineering*, *22*, 1028–1040.

[48] Zhou, M., Bao, S., Wu, X., & Yu, Y. (2007). An unsupervised model for exploring hierarchical semantics from social annotations. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings* (pp. 680–693). Springer Berlin Heidelberg.

[49] Zhu, J., Shen, B., Cai, X., & Wang, H. (2015). Building a large-scale software programming taxonomy from stackoverflow. In *SEKE* (pp. 391–396).

[50] Zubiaga, A., Fresno, V., Martnez, R., & Garca-Plaza, A. P. (2013). Harnessing folksonomies to produce a social classification of resources. *IEEE Transactions on Knowledge and Data Engineering*, *25*, 1801–1813.