Discovering and Exploiting Semantics in Folksonomies

Dissertation von Rabeeh Ayaz Abbasi

Zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften

(Dr. rer. nat.)

von der Universität Koblenz-Landau

im Fachbereich Informatik

October 2010

UNIVERSITÄT
KOBLENZ · LANDAU

Rabeeh Ayaz Abbasi

Discovering and Exploiting Semantics in Folksonomies

Vom Promotionsausschuss des Fachbereichs 4: Informatik der Universität Koblenz-Landau zur Verleihung des akademischen Grades Doktor der Naturwissenschaften (Dr. rer. nat.) genehmigte Dissertation.

Vorsitzender des Promotionsausschusses:	Prof. Dr. Dieter Zöbel
Vorsitzender der Promotionskommission:	Prof. Dr. Thomas Burkhardt
Berichterstatter:	Prof. Dr. Andreas Hotho
	Prof. Dr. Steffen Staab

Die wissenschaftliche Aussprache fand am 16. Dezember 2010 statt.

Erklärung

Hiermit erkläre ich gemäß §8 der Promotionsordnung des Fachbereichs 4: Informatik der Universität Koblenz-Landau,

- dass ich die vorliegende Dissertation mit dem Titel *Discovering and Exploiting Semantics in Folksonomies* selbst angefertigt und alle benutzten Hilfsmittel in der Arbeit angegeben habe,
- dass ich die Dissertation oder Teile der Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe, und
- dass ich weder diese noch eine andere Abhandlung bei einer anderen Hochschule als Dissertation eingereicht habe.

Koblenz, den Oktober 29, 2010

Rabeeh Ayaz Abbasi

Acknowledgements

First of all I thank the Almighty Allah for giving me the strength and health to conduct my PhD. Then I would like to thank my parents who spent a large part of their lives and resources for grooming and educating me. I would like to acknowledge my advisor Prof. Dr. Steffen Staab for guiding and supporting me. Throughout the course of my studies, I have always found him very kind and helpful.

I would also like to acknowledge Prof. Dr. Andreas Hotho for reviewing my thesis and providing useful feedback. The EU project Tagora has helped me in streamlining my research and the project partners in Tagora have provided very useful feedback on my research. My PhD would not have been possible without the financial support of Higher Education Commission (HEC) of Pakistan and the organizational support of Deutsche Akademische Austausch Dienst (DAAD). The last nine months of my PhD were funded by the EU project PetaMedia, I am really thankful to them.

All of my colleagues at the WeST institute made the PhD experience a lot wonderful. Their feedback has always helped in improving my skills and research. I really enjoyed every moment that I had spent with them. I would also like to acknowledge Sergey Chernov, Raluca Paiu, and Marcin Grzegorzek who partially contributed to this research work.

My friends Imran Wajid, Sulaiman Faisal, Tasneem Akther, Tasneem Zafar, Tazeen Ather, Imtiaz Rabbani, Akmal Khattak, Nasir Naveed, Zahid Sadal, Usman Bakhsh, Anjum Naseer, and Ammar Mohammad were like a family to me. They were with me and my family at all the occasions of happiness and sorrow. Throughout the duration of PhD, the staff members Sabine Hülstrunk, Silke Werger, Ruth Ehrenstein, and Ute Lenz-Perscheid had always been there to help me and my family.

I was not alone in bearing the difficulties of my PhD. My wife Asima had always helped me to look forward and proceed. Her efforts equally contribute to my PhD. My daughter Ayesha brought lots of happiness in my life. She had always made me smile no matter how worried I was.

Abstract

Folksonomies are Web 2.0 platforms where users share resources with each other. Furthermore, they can assign keywords (called tags) to the resources for categorizing and organizing the resources. Numerous types of resources like websites (Delicious), images (Flickr), and videos (YouTube) are supported by different folksonomies. The folksonomies are easy to use and thus attract the attention of millions of users. Together with the ease they offer, there are also some problems. This thesis addresses different problems of folksonomies and proposes solutions for these problems.

The first problem occurs when users search for relevant resources in folksonomies. Often, the users are not able to find all relevant resources because they don't know which tags are relevant.

The second problem is assigning tags to resources. Although many folksonomies (like Delicious) recommend tags for the resources, other folksonomies (like Flickr) do not recommend any tags. Tag recommendation helps the users to easily tag their resources.

The third problem is that tags and resources are lacking semantics. This leads for example to ambiguous tags. The tags are lacking semantics because they are freely chosen keywords. The automatic identification of the semantics of tags and resources helps in reducing problems that arise from this freedom of the users in choosing the tags.

This thesis proposes methods which exploit semantics to address the problems of search, tag recommendation, and the identification of tag semantics. The semantics are discovered from a variety of sources. In this thesis, we exploit web search engines, online social communities

and the co-occurrences of tags as sources of semantics. Using different sources for discovering semantics reduces the efforts to build systems which solve the problems mentioned earlier.

This thesis evaluates the proposed methods on a large scale data set. The evaluation results suggest that it is possible to exploit the semantics for improving search, recommendation of tags, and automatic identification of the semantics of tags and resources.

Zusammenfassung

Folksonomien sind Web 2.0 Plattformen, in denen Benutzer verschiedene Inhalte miteinander teilen können. Die Inhalte können mit Hilfe von Stichwörtern, den sogenannten Tags, kategorisiert und organisiert werden. Die verschiedenen Folksonomien unterstützen unterschiedliche Inhaltstypen wie zum Beispiel Webseiten (Delicious), Bilder (Flickr) oder Videos (YouTube). Aufgrund ihrer einfachen Benutzungsweise haben Folksonomien viele Millionen Benutzer. Die einfache Benutzungsweise führt aber auch zu einigen Problemen. Diese Doktorarbeit beschäftigt sich mit drei der wichtigsten Probleme und beschreibt Methoden, wie sie gelöst werden können.

Das erste dieser Probleme tritt auf, wenn Benutzer die Folksonomien nach bestimmten Inhalten durchsuchen wollen. Häufig können dabei nicht alle relevanten Inhalte gefunden werden, da diesen relevante Stichwörter fehlen.

Dementsprechend tritt das zweite Problem während der Vergabe von Stichwörtern auf. Manche Folksonomien, wie zum Beispiel Delicious, unterstützen ihre Benutzer dabei, indem sie ihnen mögliche Stichwörter empfehlen. Andere Folksonomien, wie zum Beispiel Flickr, bieten keine solche Unterstützung. Die Empfehlung von Stichwörtern hilft dem Benutzer dabei, Inhalte auf einfache Art und Weise mit den jeweils relevanten Stichwörtern zu versehen.

Das dritte Problem besteht darin, dass weder Stichwörter noch Inhalte mit einer festen Semantik versehen sind und mehrdeutig sein können. Das Problem entsteht dadurch, dass die Benutzer die Stichwörter vollkommen frei verwenden können. Die automatische Identifizierung

der Semantik von Stichwörtern und Inhalten hilft dabei, die dadurch entstehenden Probleme zu reduzieren.

Diese Doktorarbeit stellt mehrere Methoden vor, wie verschiedene Quellen für semantische Informationen benutzt werden können, um die vorher genannten drei Probleme zu lösen. In dieser Doktorarbeit benutzen wir als Quellen Internetsuchmaschinen, soziale Netzwerke im Internet und die gemeinsamen Vorkommen von Stichwörtern in Folksonomien. Die Verwendung der verschiedenen Quellen reduziert den Aufwand bei der Erstellung von Systemen, die die vorher genannten Probleme lösen.

Die vorgestellten Methoden wurden auf einem großen Datensatz evaluiert. Die erzielten Ergebnisse legen nahe, dass semantische Informationen bei der Lösung der Probleme helfen, die während der Suche von Inhalten, der Empfehlung von Stichwörtern als auch der automatischen Identifizierung der Semantik von Stichwörtern und Inhalten auftreten.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The term *folksonomy* was coined by Thomas Vander Wal in 2004. It is a portmanteau of *folk* and *taxonomy*, meaning taxonomy by folks (people). A folksonomy is also often called a *collaborative tagging system* or simply a *tagging system*. It mainly consists of three elements, *users*, *resources*, and *tags*.

With the popularity of Web 2.0, many folksonomy platforms have emerged in the past few years. Folksonomies are built for almost any type of media. For example, there are folksonomies for sharing bookmarks[1], books[2], citations[3], music[4], photos[5], and videos[6]. Tagging has also been used in other domains than sharing resources. E-commerce websites like Amazon[7] allow their buyers to tag products. Users can share and tag their goals at 43things[8], and social networking websites like facebook[9] allow their users to tag photos and videos. Details of different types of folksonomies with their features are available in the book (Peters, 2009, chap. 1).

With an increasing interest of users, folksonomies are expanding tremendously.

---

[1] `http://www.delicious.com/`, `http://www.bibsonomy.org/`
[2] `http://www.librarything.com/`
[3] `http://www.citeulike.org/`, `http://www.bibsonomy.org/`
[4] `http://www.last.fm/`
[5] `http://www.flickr.com/`
[6] `http://www.youtube.com/`
[7] `http://www.amazon.com/`
[8] `http://www.43things.com/`
[9] `http://www.facebook.com/`

In September 2010, there were five billion photos on Flickr[1] and in November 2008 there were 180 million unique URLs on delicious uploaded by 5.3 million users[2]. With the passage of time, more and more users use folksonomies to share their resources with other users.

Folksonomies are easy to use since users do not require specialized skills for using them. Tags are freely chosen keywords and users are free to create or select tags of their choice. The tagging process in folksonomies is informal. For tagging resources, no fixed set of words is used in folksonomies, this allows the users to create and use new words such as *toread* (representing the verb "to read") or *day50* (representing the "50th day" of a year). Tags in folksonomies are used for navigation, searching, and browsing (Peters, 2009).

In addition to facilitating the sharing of resources, some folksonomies provide additional tools to their users. For example, Flickr allows the creation of online communities using its groups. Each Flickr group has its own topic. In a group, users post images related to the topic of the group. These groups assist users in searching and browsing resources related to a particular topic.

Folksonomies provide many benefits to its users, but problems also occur while using them. One of the problems is searching resources in folksonomies. Often there are insufficiently many relevant tags associated to the resources. Fewer relevant tags make it difficult to search for resources in folksonomies. The next problem is recommending tags for the resources. As people share a large number of resources, it becomes difficult for them to assign tags to their resources. Although some folksonomies like Delicious provide tag recommendations, but other folksonomies like Flickr do not provide any tag recommendation. Another problem is the lack of semantics in folksonomies.

For example, it is not obvious by looking at the tag *Paris*, whether it represents a person or a city. In this thesis we address these problems by discovering and exploiting semantics in folksonomies. The following sections describe each of these problems in detail and the solutions presented in this thesis.

[1] `http://blog.flickr.net/en/2010/09/19/5000000000/`
[2] `http://blog.delicious.com/blog/2008/11/delicious-is-5.html`

1.1 Search

Despite the enormous size of folksonomies, resource retrieval in folksonomies is limited (Hotho et al., 2006). Users are not obliged to assign tags to their resources. Often they do not add many relevant tags to their resources. Lack of a sufficient number of tags results in sparseness of data and it becomes difficult to search resources related to a query. For example, a user searching for funny pictures of seventies using the tags *funny* and *seventies* will get only the images which are tagged with the tags *funny* and *seventies*. The user will not be able to retrieve the resources which are tagged with the tags *1970s* and *humorous* (instead of the tags *seventies* and *funny*), although the resources tagged with the tags *1970s* and *humorous* are also relevant to the query. While searching resource in a folksonomy, many resources are not retrieved because they are not associated with many of the relevant tags.

To improve search in folksonomies, we propose methods which discover and exploit semantics. The proposed methods discover tags which are semantically related and use these semantically related tags to enrich the data in folksonomies. By enriching the data, many relevant resources are retrieved, that are otherwise not retrieved.

We discover the semantically related tags based on the context and the type of the similarity between the tags. The contexts of the tags give different perspectives to them. We consider two types of tag contexts for improving search in folksonomies: the resource context and the user context. The resource context of tags helps in finding tags which are mostly used for similar kind of resources, whereas the user context finds broad relationships between tags based on the users' interests (represented by the tags they use).

We evaluate the methods proposed for improving search in folksonomies on a large scale dataset (having around 27 million resources). Experimental results show that the enrichment of existing data helps in improving resource retrieval, particularly for the queries for which only few relevant resources exist.

1.2 Tagging

While sharing resources on a folksonomy users often add tags to their resources. Adding tags could be a tedious job, particularly when sharing many resources at once, for example, sharing pictures of a trip. It is important to automate the process of tagging by developing the *Tag Recommender Systems*. They assist users in the tagging phase.

We analyze different types of resource features like geographical coordinates, tags, and low-level image features for tag recommendation. The analysis helps in selecting the appropriate type of resource feature. We develop a framework for tag recommendation that does not require manual training. The framework automates the process of tagging in folksonomies. It relies on clustering techniques for recommending tags. The resources are first clustered based on a feature. Then tags in each cluster are aggregated to get a list of the most representative tags for the clusters. Resources in each cluster are used to train a classifier which classifies a new resource to one of the clusters. The most representative tags of the cluster are then recommended for the resource. Results of experiments performed to evaluate the tag recommendation framework on a large scale dataset show that the geographical coordinates of the resources give the best results when compared to other features.

1.3 Semantics

Tags are freely chosen or generated by the users in contrast to the formal annotations. The users do not require any formal background knowledge for tagging their resources. It is possible that the users are not an expert on the subject of the resource being tagged. The tags can therefore be of varying quality and lack formal semantics. Lack of semantics makes it difficult for machines to understand the meanings of the tags and the type of resources. Considering current searching and browsing facilities provided by these systems, it seems difficult to identify the semantics of tags and resources. We propose methods for identifying the semantics of tags and resources by classifying them into categories. Classification helps in identifying resources which might be interesting for some users, for example, in identification of images which show worth visiting sights or landmarks of a city. Classification might also help in faceted browsing by categorizing the resources and tags into categories like persons, organizations, or places etc.

We present a system *T-ORG (Tag-ORGanizer)* to discover the semantics of tags. T-ORG organizes resources by classifying the tags attached to them into predefined categories. To avoid the efforts required to train the classifier used in T-ORG, we develop a classification algorithm *T-KNOW* (Tags classification through KNOwledge on the Web). This algorithm classifies the tags into predefined categories by using lexico-syntactic patterns and a web search engine. It does not require manually labeled training data to learn a classification model. Given a list of tags and categories, *T-KNOW* classifies these tags into categories.

In addition to the classification of tags using web resources, we also exploit the information available in online social communities for classifying resources. We propose a system *TG-SVM* (Tag Group Support Vector Machine) which exploits the information available in online social communities like Flickr groups for classifying resources representing the landmarks of a location. The method involves minimum human efforts as it only requires the links to the relevant Flickr groups. The system automatically trains a classifier based on the data retrieved from these groups. Evaluation results show that TG-SVM outperforms the state of the art methods.

1.4 Structure of the Thesis

The thesis starts with an introduction to folksonomies. Its first chapter identifies the following three problems related to different aspects of folksonomies:

1. Searching
2. Tagging
3. Semantics

Chapter 2 discusses the limitation of folksonomies and gives an overview of the research work related to the solutions of these limitations.

Chapter 3 describes the formal representation of folksonomies. Formal representation is important, because it enables to describe and understand the problems and solutions in a systematic way. The chapter also covers the different elements of folksonomies and their respective formal representations by using two alternative methods, graphs and vector spaces.

Chapter 4 introduces the methods for discovering semantics in folksonomies. The chapter discusses the methods to discover semantically related tags by exploiting external and internal data sources. The external data sources include WordNet (Miller, 1995) and *Application Programming Interfaces* (APIs). The internal data sources include different co-occurrence and probabilistic methods. The methods for discovering semantically related tags presented in the chapter are published in (Abbasi, 2010, to appear).

The semantically related tags discovered in Chapter 4 are exploited in Chapter 5 to reduce the sparseness and to improve search in folksonomies. Chapter 5 exploits the semantic relationships between tags to enrich the data in folksonomies. It proposes enriched vector space models for improving search. The proposed methods in the chapter are evaluated on a large scale using a dataset of around 27 million resources, 92,000 tags and 94 million tag assignments. Experimental results show that the enriched vector space models help in improving search, especially for the queries with few relevant resources. The methods used in the chapter are also published in (Abbasi and Staab, 2009).

Chapter 6 exploits different kinds of resource descriptions to recommend tags for new resources uploaded to a folksonomy. It compares the tags, the geographical coordinates, and the low-level image features. Tags are suggested to the new resources which are uploaded to a folksonomy. The chapter presents a framework which relies on clustering for tag recommendation. The resources are first clustered based on a feature. Tags in each cluster are aggregated to get a list of the most representative tags for that cluster. Then each cluster is used as a training model and a new resource is classified to one of the clusters. The most representative tags of the cluster are then recommended for the resource. Results of experiments performed on a large scale dataset show that the geographical coordinates of resources give the best results when compared to other available features. We published the outcomes of the chapter in (Abbasi et al., 2009b).

Chapter 7 proposes methods for classifying tags and resources. The chapter presents a system called T-ORG (*Tag ORGanizer*) which classifies the tags into predefined categories. The evaluation results show that T-ORG can be used to discover semantics of tags and resources in an effective manner. The chapter also proposes a method called *TG-SVM* (Tag Group Support Vector Machine) for classifying resources by exploiting information available in online social communities. The user study presented in the chapter shows that the proposed method outperforms state-of-the-art systems that address the same kind of problems. The work related to this chapter has been published in (Abbasi et al., 2007, 2009a)

Finally Chapter 8 summarizes the main contributions of the thesis.

1.5 Publications Related to the Thesis

The research work presented in this thesis has been published at various conferences. Following is the list of most relevant publications.

- Chapter 4: Rabeeh Abbasi. Query Expansion in Folksonomies: In *Proceedings of 5th International Conference on Semantic and Digital Media Technologies, Semantic Multimedia*, Lecture Notes in Computer Science, Berlin, Heidelberg, 2010, to appear. Springer Verlag. (Abbasi, 2010, to appear).

- Chapters 4 and 5: Rabeeh Abbasi and Steffen Staab. RichVSM: enRiched vector space models for folksonomies: In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 219–228, New York, NY, USA, 6 2009. ACM. (Abbasi and Staab, 2009).

- Chapter 6: Rabeeh Abbasi, Marcin Grzegorzek, and Steffen Staab. Large Scale Tag Recommendation Using Different Image Representations. In *Proceedings of 4th International Conference on Semantic and Digital Media Technologies, Semantic Multimedia*, volume 5887 of *Lecture Notes in Computer Science*, pages 65–76, Berlin, Heidelberg. Springer Berlin / Heidelberg. (Abbasi et al., 2009b).

- Chapter 7: Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, and Steffen Staab. Exploiting Flickr Tags and Groups for Finding Landmark Photos. In *Proceedings of European Conference on Information Retrieval 2009, Advances in Information Retrieval*, volume 5478 of *Lecture Notes in Computer Science*, pages 654–661, Berlin, Heidelberg, 4 2009. Springer Berlin / Heidelberg. (Abbasi et al., 2009a).

- Chapter 7: Rabeeh Abbasi, Steffen Staab, and Philipp Cimiano. Organizing Resources on Tagging Systems using T-ORG: In *Proceedings of Workshop on Bridging the Gap between Semantic Web and Web 2.0 at European Semantic Web Conference 2007*, pages 97–110, Innsbruck, Austria, June 2007. (Abbasi et al., 2007).

Chapter 2

Related Work

This chapter discusses the research work related to the problems discussed in the previous chapter. The related work is divided into three categories, the first category is search and information retrieval, the second is tag recommendation, and the third is semantics and classification. The research work related to each of these categories in perspective of folksonomies is listed in the following sections.

2.1 Search and Information Retrieval

Tags are one of the common elements in folksonomies. They are important for searching and retrieving resources. Assigning tags to the resources is neither formal nor mandatory; therefore the motivation behind tagging varies among different users. Tagging motivations range from self-organization to social communication (Ames and Naaman, 2007; Nov et al., 2008, 2010). Nov et al. (2010) investigate four motivations (Self-Development, Reputation, Enjoyment, and Commitment) for participation of users in Flickr. They find that newer folksonomy users are more motivated for self-development than the old community members. The users who are motivated to improve their reputation mostly add meta-information (e.g. tagging) to their resources to draw attention of other users. They also discover that the motivation for enjoyment is not correlated with the number of tags or photos shared. They argue that this could be due to the reason that users might enjoy in taking photos as a first step, but find sharing and tagging photos less enjoying. Nov et al. (2010) also discover that the users who are highly committed to the community add more information to their resources as opposed to the users who are less committed.

Many researchers have proposed methods to improve search in folksonomies. Hotho et al. (2006) present an algorithm called *FolkRank* based on the *PageRank* algorithm for ranking tags, resources, and users in folksonomies. FolkRank gives higher rank to the resources which are tagged by important users and important tags. Similarly tags and users can also be ranked using FolkRank. The tags, users, and resources which are ranked higher by FolkRank are considered more important or relevant than the others. (Hotho et al., 2006) describes the FolkRank algorithm in detail.

The research work presented in this thesis differs from the research work done by Hotho et al. (2006) in different aspects. For example, our focus is on improving search by enriching the data in folksonomies, whereas the FolkRank algorithm is used for ranking the tags, resources, or users. Ranked elements can be used for generating recommendations. The evaluation performed by Hotho et al. (2006) does not show the explicit significance of the FolkRank algorithm for improving search in folksonomies.

Yahia et al. (2008) present methods based on network-aware search in folksonomies. Their proposed methods construct clusters of users based on users' activity similarity. The top-k querying methods they propose can be used for ranking resources. In their experiments, they show that clustering users leads to improvements in both space and execution time. In comparison to their work, our focus for improving search in folksonomies is at system level without going into preferences of individual users.

This thesis mainly focuses on improving search in folksonomies by exploiting semantically related tags. Other researchers have developed methods based on cross language information retrieval to improve search in folksonomies. For example, Noh et al. (2009) propose methods to improve search by translating the tags into different languages. Their proposed method translates the tags by analyzing the global tag co-occurrence methods.

Users can create or choose any set of tags to annotate their resources. A data analysis shows that there are only *3.1* tags associated to each resource on an average in a large folksonomy dataset of 54 million resources (Bolettieri et al., 2009). The fewer number of tags associated to resources makes the available data in folksonomies very sparse. The sparseness of data leads to difficulty in searching resources. Often relevant resources are not retrieved for a search query because the resources are associated with different but semantically related tags. One way of reducing the sparseness is to exploit semantically related tags.

Markines et al. (2009) analyze different types of similarity measures to discover semantically related tags. The simplest approach they considered is based on tag co-occurrence. Tag co-occurrence between two tags counts the number of resources in which the two tags appear together. Other than using co-occurrence information, Markines et al. (2009) also propose similarity measures like *Overlap*, *Jaccard*, *Dice*, *Cosine*, and *Mutual Information* to discover semantically related tags. They evaluate the tag relatedness by computing the relative placement of tags in WordNet hierarchy. Our work relates to the work done by Markines et al. (2009) with respect to discovering semantically related tags. They found that the semantic relationships between tags are best discovered using *Mutual Information*, which is also computationally the most expensive method. The methods presented in this thesis for discovering semantically related tags and exploiting them for enriching the vector space models were published (Abbasi and Staab, 2009) around the same time as the research by Markines et al. (2009). Some of the methods used in this thesis (see Chapters 4 and 5) for discovering semantically related tags like *Cosine* similarity have also been used by Markines et al. (2009). Additionally we have proposed an asymmetric co-efficient (*Modified Overlap Coefficient*) for discovering semantically related tags. It could be an interesting future work to incorporate the best results (e.g. *Mutual Information*) found by Markines et al. (2009) into the enrichment methods presented in this thesis. In comparison to the work presented by Markines et al. (2009), our work further extends the semantically discovered tags to improve search in folksonomies, whereas Markines et al. (2009) focus on the semantic grounding of semantically related tags.

Semantically related tags have also been used to enrich ontologies. Mika (2007) presents methods to extract ontologies from folksonomies. The proposed methods induce ontologies from large folksonomy data using information available through tag co-occurrence. The methods also build hierarchical relationships between tags by discovering semantically related tags. Although the main focus of the research work presented by Mika (2007) revolves around extracting ontologies from folksonomies instead of improving search, but there are some similarities between the work presented in this thesis and Mika (2007)'s work. The methods proposed by Mika (2007) implicitly exploit the contexts of resources and users. We also exploit the contextual information to discover semantically related tags. In addition, our proposed methods use different similarity methods than the simple co-occurrence, because simple co-occurrence has a bias towards relating very frequent tags to all the tags.

In addition to improving search by enriching the folksonomies, some researchers have proposed methods to improve search by query expansion. Probabilistic methods for query expansion have been proposed by Collins-Thompson and Callan (2005); Lafferty and Zhai (2001). Billerbeck et al. (2003); Cui et al. (2002) exploit query logs for expanding queries. Arguello et al. (2008) proposes different representations of blogs for expanding queries. Bertier et al. (2009) expands queries using representative tags of a user in folksonomies. Their query expansion method is personalized for the user who gives the query. Pan et al. (2009) propose to expand folksonomy search using ontologies.

The methods proposed in this thesis to discover semantically related tags exploit the type of similarity between the tags and the context of the tags simultaneously. In comparison to the presented related work, we exploit the discovered relationships among tags for improving search in folksonomies, particularly by reducing the sparseness in folksonomies.

2.2 Tag Recommendation

A tag recommendation system is used to assist users in tagging resources. These systems have been discussed in various research works over the last few years. Researchers have come up with frameworks which allow the comparison of different tag recommendation methods. Jäschke et al. (2009) present a tag recommendation framework for their system Bibsonomy. The framework allows the evaluation of different tag recommendation algorithms. The framework is though limited to the tag recommender systems which only use the tagging information.

Jäschke et al. (2007) compared two algorithms, FolkRank and Collaborative Filtering (Goldberg et al., 1992) for tag recommendation. FolkRank is based on PageRank (Brin and Page, 1998). It uses random walk techniques on the graph of users, tags, and resources and assumes that popular users, tags, and resources can reinforce each other. In collaborative filtering, similarity between users and tags and between users and resources is used to recommend tags. Their experiments based on the datasets from delicious[1], last.fm[2], and Bibsonomy[3] show that the FolkRank algorithm outperforms other methods. The tag recommendation methods as proposed by Jäschke et al. (2007) depend mainly on the tagging information and do not consider the features (like geographical coordinates or low-level image features) available in rich media (like photos or videos). The tag recommendation framework presented in this thesis considers and compares different features like geographical coordinates and low-level image features available in rich media. The tag recommendation methods proposed by Jäschke et al. (2007) suggest tags for already partially tagged resources, whereas the tag recommendation framework proposed in this thesis can suggest tags for newly uploaded resources which are not associated with any tags.

Sigurbjörnsson and van Zwol (2008) present a tag recommendation system which evaluates different similarity measures, tag aggregation methods and ranking strategies. Given a photo and some initial tags, candidate tags are derived for each of the given tag. The candidate tags are retrieved based on the tag co-occurrence information. All of the candidate tags are then merged and ranked. A final list of tags is then presented to the user. As in the work presented by Jäschke et al. (2007), the methods proposed by Sigurbjörnsson and van Zwol (2008) lack a tag recommendation strategy for newly uploaded resources. Although the experiments were performed on Flickr (photos) dataset, the methods do not consider the available rich media features.

Nowadays, the state-of-the-art imaging devices provide photos together with the geographical coordinates (*geo-tags*) stating precisely where they have been

[1]http://www.delicious.com/
[2]http://www.last.fm/
[3]http://www.bibsonomy.org/

acquired. Therefore, more and more researchers make use of this additional information. Cristani et al. (2008) exploit geographical coordinates for improving visualization of images on a map. Kennedy et al. (2007) and Kennedy and Naaman (2008) use low-level image features and geographical coordinates to identify the landmarks of a city. Moëllic et al. (2008) present a system which combines tags and low-level image features for clustering images. They suggest that clustering images can enhance the browsing and visualization of the images. But none of these approaches exploit the features available in rich media for recommending tags. In comparison to the above mentioned methods, the contribution of this thesis regarding tag recommendation is twofold, first, we develop a tag recommendation system, which recommends tags for newly uploaded resources, and second we compare the performance of different rich media features in the process of tag recommendation.

In addition to the features available in rich media, some researchers have used external data sources for recommending tags. Heymann et al. (2008) predict tags by using information available in the resource content, anchor text, and already available tags. Given a set of objects, and a set of tags applied to those objects by users, their approach predicts whether a given tag could/should be applied to a particular object. Heymann et al. (2008) formulate the problem of tag recommendation into a supervised learning problem. For each tag to be recommended, they train a binary (SVM) classifier which predicts the association of a resource with the tag. Their approach is limited to a set of tags that can be recommended and may not be applied in a generic large scale scenario. Resource features like titles of webpages have also been exploited by other researchers (Lipczak, 2008). Lipczak (2008) suggests a tag recommendation system which extracts the tags from the resource title. The tag co-occurrence information available within the resource's posts in form of a personomy is used for recommending tags. In addition to exploiting external data sources and resource contents, some researchers have also used formal ontologies in the process of tag recommendation. Adrian et al. (2007) present a system called ConTag. It generates semantic tag recommendations for documents based on Semantic Web ontologies and Web 2.0 services.

Other recent research work related to tag recommendation includes (Illig et al., 2009; Rae et al., 2010; Krestel et al., 2009; Jin et al., 2010). Illig et al. (2009) present an algorithm for tag recommendation based on the contents of the resources. They train different classifiers like *SVM* and *Multinomial Naïve Bayes* etc. on the training data for recommending tags for newly upload resources. Rae et al. (2010) propose a method of tag recommendation for partially annotated media. Their method exploits different contexts of the users. It achieves best results using the *Social Group* context. Krestel et al. (2009) and Jin et al. (2010) have used probabilistic methods like *Latent Dirichlet Allocation* (LDA) for tag recommendation. The method proposed by Krestel et al. (2009) discovers latent topics using tagging information and these topics are then used to recommend tags for the new resources belonging to the same topic. Jin et al. (2010) combine *Language Model* and *Latent Dirichlet Allocation* to recommend tags. The focus of these recent research works is basically on recommending tags using the tagging or tag co-occurrence information, and these research works do not utilize the information available in the rich media.

Finally some of the tag recommendation methods exploit rich media features like geographical coordinates and low-level image features. Moxley et al. (2008) present a tag recommendation tool called *SpiritTagger* which uses the geographical coordinates of the images available at Flickr. Their approach weights geographically relevant annotations for tag recommendation for an image database. Experimental results on two cities show that their approach outperforms the geographical and visual baselines for smaller cities, but the geographical coordinates give the best results for larger cities. The tag recommendation framework presented in this thesis also focuses on tag recommendation based on different rich media features and compare the performance of each of these features. It would be an interesting future work to compare the performance of our proposed framework and the SpiritTagger.

2.3 Semantics and Classification

Tags and resources in folksonomies lack formal semantics. The tags and resources are not classified using any taxonomy or ontology. The lack of formal semantics in folksonomies makes it difficult for an algorithm or system to understand it.

The problem of the lack of semantics in tags and resources has been considered by many researchers. Some researchers have focused on mapping folksonomy tags on a formal semantic vocabulary. For example, Schmitz (2006) presents a method to induce taxonomies from tags using a probabilistic method. Schmitz (2006) proposes to use a probabilistic subsumptions based model to produce hierarchical relationships between tags. As compared to the work presented in this thesis, the method proposed by Schmitz (2006) does not classify tags into predefine categories, which could help in focused or faceted browsing of resources.

The methods to identify semantics of tags proposed in this thesis exploit external data sources like web search engines along with linguistic patterns. The idea of utilizing linguistic patterns for identifying semantics of words and terms is not new. Hearst (1992) used lexico-syntactic patterns to extract hyponyms from large text corpora. These linguistic patterns have been used by other researchers for identifying semantics. The classification algorithm T-KNOW presented in this thesis for identifying semantics of tags is based on the C-PANKOW system (see (Cimiano et al., 2004) and (Cimiano et al., 2005)), which uses lexical patterns along with the search results from a web search engine for semantic annotation of web pages. With respect to folksonomies, we extended the research work done by Cimiano et al. (2005) by exploiting four different types of tag contexts.

In addition to ontologies and linguistic patterns, researchers have also used classification methods for identifying the semantics of tags and resources. Overell et al. (2009) present a method for classifying tags using *Wikipedia*[1] and the *Open Directory Project*[2]. They developed a classifier which utilizes information available in Wikipedia and WordNet and classifies the tags to the anchor text in Wikipedia articles. The Wikipedia articles are themselves categorized into WordNet categories. Similarly, Angeletou et al. (2009) propose a framework for annotating folksonomy tags using formal knowledge. Their framework exploits WordNet and online available ontologies for identifying semantics of tags. Their experiments show that the folksonomy enrichment using ontologies based methods outperform the enrichment based on WordNet. Although, the methods proposed in this thesis also identify the semantics of tags and resources, the method of iden-

[1] `http://www.wikipedia.org/`
[2] `http://www.dmoz.org/`

tifying semantics differs from the methods presented by Angeletou et al. (2009). We discover the semantics by exploiting online resources like web search engines and lexico-syntactic patterns. It would be an interesting future task to develop an evaluation framework where different approaches for identifying semantics in folksonomies can be compared.

In addition to identifying the semantics of tags, we also identify the semantics of resources by exploiting information available in online communities like Flickr groups[1]. In particular we focus on identifying the photos in Flickr which represent the landmarks of a city. Previous algorithms for identifying landmark photos have employed both purely content-based techniques, as well as methods combining content and contextual information of the pictures. Popescu et al. (2008) use external data sources (Geonames, Wikipedia, Panoramio, and Search engines snippets) to extract geographical entities of a place. Jaffe et al. (2006) propose an approach for generating photo summaries relying on hierarchical clusters; each of these clusters is scored and finally a flat ordering of all photos in the dataset is generated. In a later work, Kennedy et al. (2007) replace the original clustering algorithm with K-Means and add the analysis of image visual features. A similar approach combining context- and content-based tools is presented in (Kennedy and Naaman, 2008). Landmarks are detected by analyzing the distribution patterns of the tags in the dataset. The representative pictures for a landmark are identified based on canonical views, using various image processing methods. Also using content-based techniques, Berg and Forsyth (2007) present an algorithm which ranks iconic images labeled with a particular theme, according to how well they represent a visual category.

Closely related to the work presented in this thesis, Ahern et al. (2007) propose a system called *World Explorer* for identifying landmarks of a city. *World Explorer* clusters the photos according to their geographical location. Then the tags in each cluster are ranked according their representativeness. A TF-IDF based method is used for ranking the tags. Ahern et al. (2007) suggest that the identified clusters represent an interesting location and the ranked tags suggest the labels for the clusters. The method proposed in this thesis called *TG-SVM* utilizes information available in online communities to learn a classifier which

[1] `http://www.flickr.com/groups/`

can identify landmark photos. *TG-SVM* does not require the geographical coordinates for identifying landmark photos as in the case of *World Explorer*. This could be particularly useful for classifying resources which do not contain geographical information. The experiments comparing *TG-SVM* to *World Explorer* show that *TG-SVM* mostly outperforms *World Explorer*.

Once the semantics of tags and resources are discovered, they can be used in a variety of applications. One such application is faceted browsing, where facets from the dataset are displayed to the user and the user can narrow down his browsing experience using these facets. An example of such a system which supports facets and semantics is *SemaPlorer*. *SemaPlorer* allows users to browse and visualize resources in real time. The data (e.g. images from Flickr) is visualized on a map and users can narrow down the results by selecting different facets like *persons*, *locations*, or *tags*. Systems like *SemaPlorer* can be further enhanced by utilizing the methods proposed in this thesis for identifying semantics of tags and resources.

2.4 Conclusions

This chapter described the research work related to the problems of searching, tagging, and semantics as discussed in Chapter 1. The related work mentioned in this chapter discussed a variety of methods to address these problems. The discussed solutions varied from methods which analyze the semantics in folksonomies to the frameworks which allow comparison of different methods.

Chapter 3

Representation of Folksonomies

When millions of users share their resources, they generate huge amounts of data. To analyze folksonomy data, we need to identify and formalize different elements of folksonomies. In this chapter, we discuss different elements of folksonomies and their formal representation. We describe two alternative methods, graphs and vector spaces, to formally represent a folksonomy. Formal representation of folksonomies let us analyze folksonomies in a systematic way.

3.1 Elements of Folksonomies

Mainly, a folksonomy consists of three elements *Users*, *Tags*, and *Resources* (see Figure 3.1), where users annotate resources with keywords called tags. Users are free to create tags of their choice. Details about tags, users, and resources and their interaction in a folksonomy are described in the following sections:

3.1.1 Users

A user interacts with a folksonomy in multiple ways like browsing, searching, and sharing. While sharing a resource, a user can associate tags to the resource. Researchers have suggested different motivations like enjoyment, commitment, self-development, and reputation for sharing resources on folksonomies. Motivations of users behind sharing their resources are discussed in detail by Ames and Naaman (2007), Nov et al. (2008) and Nov et al. (2010).

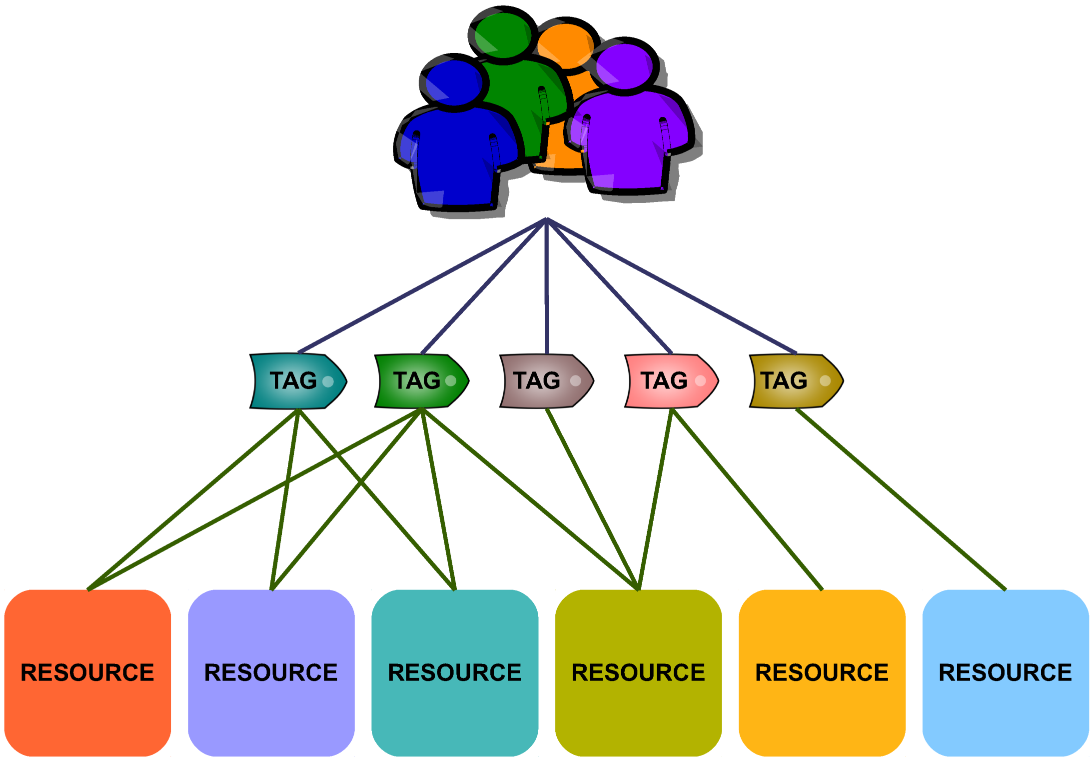

Figure 3.1: Relationships between elements of folksonomies.

3.1.2 Resources

Folksonomies support different types of resources. For example, a user of Flickr[1] can share photos, one can use YouTube[2] for sharing videos, Delicious[3] for bookmarks, Citeulike[4] for publications, Last.fm[5] for music etc. Some folksonomies also support multiple types of resources e.g. in Bibsonomy[6] users can share their bookmarks as well as publications.

In some folksonomies like Flickr or YouTube, users own their resources, e.g. pictures or videos they upload. Mostly they themselves assign tags to their resources. Rarely other users might also add tags to the resources of other users. Tagging resources of other users is sometimes not allowed in these folksonomies (e.g. when owner of the resource does not allow doing so). Such folksonomies in which mostly the owners assign tags their resources are called narrow folksonomies (Vander Wal, 2005). On the other hand, some folksonomies allow the sharing of common resources, such as websites[7], songs[8], or books[9] etc. In these folksonomies, different users can assign keywords to the same resources; such folksonomies are called broad folksonomies (Vander Wal, 2005).

3.1.3 Tags

Tags are assigned to the resources. They are selected or created by the users themselves. For example, if a user shares a website `http://last.fm/`, he might add the tags *university, Koblenz, Germany, to_visit* to it. Later on, he or other users can search the folksonomy using one or more of these tags. Tags are not only used for searching resources, but they can also be used for browsing the resources.

Folksonomies usually provide a *tag cloud* which displays the most frequent

[1] `http://www.flickr.com/`
[2] `http://www.youtube.com/`
[3] `http://www.delicious.com/`
[4] `http://www.citeulike.org/`
[5] `http://www.last.fm/`
[6] `http://www.bibsonomy.org/`
[7] `http://www.delicious.com/`
[8] `http://www.last.fm/`
[9] `http://www.librarything.com/`

(or recent) tags in the folksonomy. Figure 3.2 shows a tag cloud of the delicious website. Most frequently used tags like *design*, *blog*, and *video* etc. are shown in a bigger font size, whereas relatively less frequent tags like *car*, *flickr*, and *artist* are displayed in a smaller font size. When a user clicks on a tag, he gets a list of resources associated with the clicked tag. The tag clouds can be personalized by displaying tags of a particular entity (e.g. a user).

Figure 3.2: A tag cloud displaying most popular tags in delicious.

There are a few tags which are used by many of the users. Figure 3.3 shows the frequency of users using a particular tag. We observe that only few tags are used by many users and most of the tags are used by only a few users.

3.1.4 Other Elements of Folksonomies

The common elements of folksonomies are users, tags, and resources. But some folksonomies also provide additional elements like tag bundles[1], networks of users[2],

[1] `http://blog.delicious.com/blog/2005/10/bundle_up.html`, last accessed in October 2010

[2] `http://blog.delicious.com/blog/2006/04/its_made_out_of.html`, last accessed in October 2010

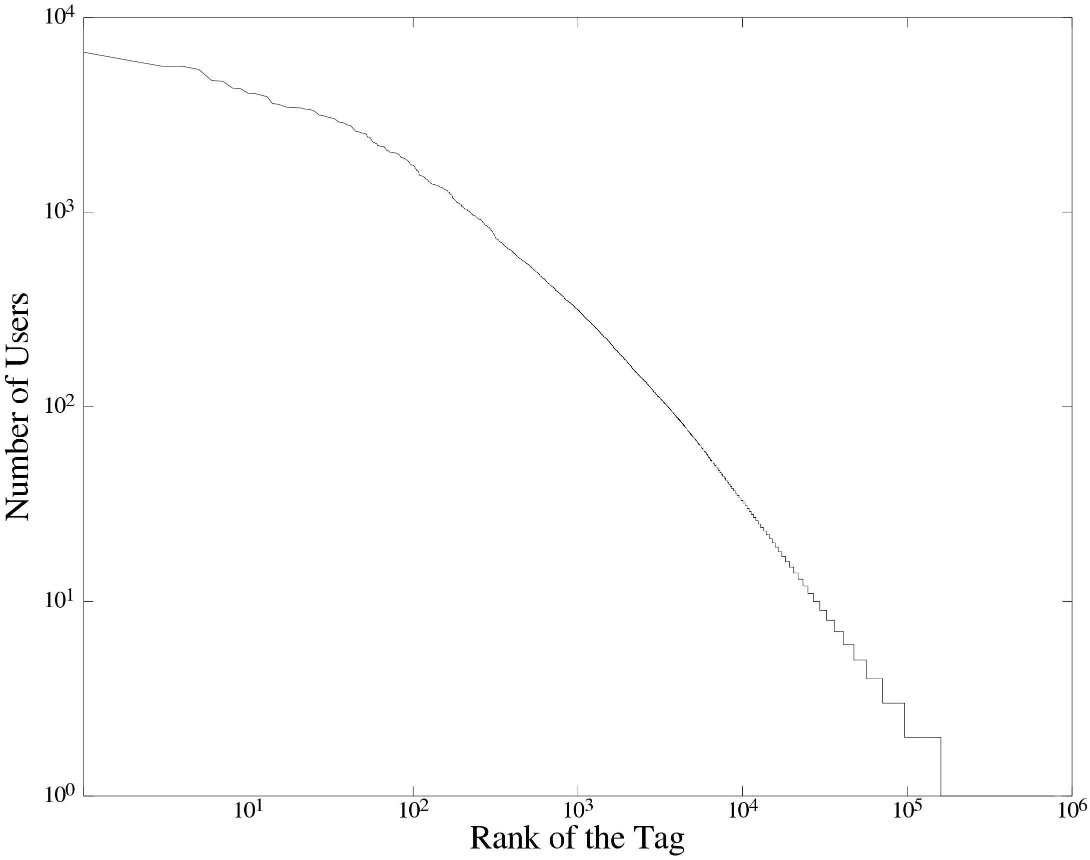

Figure 3.3: Number of users per tag.

and groups[1] etc. These elements can also be used for discovering semantics from folksonomies. A major part of this thesis presents techniques for discovering and exploiting semantics from users, tags and resources, but Chapter 7 exploits other elements like online social communities for discovering and exploiting semantics. In the following sections, we formally represent the main elements a folksonomy and their relationships using graph and vector based representations.

3.2 Graph Based Representation

A folksonomy can be represented by a tri-partite hyper-graph. Where users, tags, and resources are the nodes of the graph and they are connected by hyper-edges, showing relationships between a user, a tag, and a resource. We use the same formal model of folksonomies as defined in (Schmitz et al., 2006). A folksonomy is a tuple:

$$F := (U, T, R, Y) \tag{3.1}$$

Where U, T, and R are finite sets representing users, tags, and resources respectively. Y represents taggings by users U, using tags T of resources R, and $Y \subseteq \{U \times T \times R\}$. The relationships between different entities of a folksonomy can be represented using bipartite graphs which are projections of the original tri-partite hyper-graph. Relationships between users and tags can be represented by the bipartite graph over $\{U \times T\}$. Similarly, relations between tags and resources, and resources and users can be represented by the graphs $\{T \times R\}$ and $\{R \times U\}$ respectively.

3.3 Vector Based Representation

In addition to the graph representation of folksonomies, vector space models can be used for representing the relationships between users, tags, and resources in a folksonomy. The relationships between users, tags, and resources can be

[1] `http://www.flickr.com/groups/`, last accessed in October 2010

projected on a two dimensional vector space in form of matrices. Let us define the relationships between tags and users using the matrix U as follows:

$$U = [u_{i,j}] \tag{3.2}$$

Where $u_{i,j}$ is equal to 1, if user j has used the tag i, otherwise $u_{i,j}$ is equal to 0. Each row vector $\vec{u}_{i,*}$ of the matrix U represents a tag vector, whose non-zero elements represent the users that have used this tag. Each column vector $\vec{u}_{*,j}$ of the matrix U represents the users. In this thesis, we denote row vectors with the subscript $i, *$ and the column vectors with the subscript $*, j$ throughout the text.

Similar to the matrix U, we represent the relationships between tags and resources using the matrix R as follows:

$$R = [r_{i,j}] \tag{3.3}$$

Where $r_{i,j}$ denotes how many times the tag i appeared with the resource j. Each row $\vec{r}_{i,*}$ of the matrix R is a tag vector, whose non-zero elements represent how many times these elements (resources) have been annotated with this tag (i). Each column vector $\vec{r}_{*,j}$ of the matrix R represents a resource, which has non-zero values for the associated tags, and zero for the tags it does not use. As there are millions of tags and resources, but each resource is assigned only few tags, therefore the matrix R is a very sparse matrix.

The matrix U is a sparse matrix, but denser than the R matrix, because the column vector $\vec{u}_{*,j}$ has non-zero value for all the tags which are used by the user j and it is more likely that the set of tags used by a user is bigger than the tags used in a resource.

In some folksonomies (called *Narrow Folksonomies* (Vander Wal, 2005) like Flickr), a resource cannot be tagged with a tag more than once, while in other folksonomies (called *Broad Folksonomies* (Vander Wal, 2005)) a single resource can be tagged with a tag multiple times from different users. In case of *Narrow Folksonomies*, the value of $r_{i,j}$ is always equal to zero or one.

3.4 Conclusions

This chapter discusses the three main elements of folksonomies: users, tags, and resources. It furthermore discusses how they interact within a folksonomy. It also discusses graph and vector based representations of the main elements of folksonomies. It also lists non-standard elements which are found in some folksonomies. These elements include tag bundles, user networks, and user groups.

Chapter 4

Discovering Semantic Relationships among Tags

Only few tags are on average associated with each resource in a folksonomy. Due to the lack of a sufficient number of relevant tags, data in folksonomies is very sparse. The sparseness of data makes search and retrieval in folksonomies difficult. This sparseness can be overcome by discovering semantically related tags. This chapter discusses the methods to discover semantically related tags. The methods use external data sources like WordNet and Data Application Programming Interfaces (APIs) as well as the internal data of folksonomies. We propose two dimensions for discovering semantically related tags, one is based on the type of similarity between the tags, and the second is based on the context of the tags. The application of the methods proposed in this chapter is shown in the next chapter, where semantically related tags are used to reduce the sparseness of folksonomy data for improving search.

4.1 Discovering Semantically Related Tags

Data in folksonomies is very sparse as compared to ordinary text documents. The amount of tags associated to a resource is much less than the words associated to a document. Figure 4.1 shows the number of tags associated with around 27 million resources uploaded to the Flickr website between the years 2004 and 2005.

A user searching against a query might not find relevant resources because many relevant resources might not be tagged with the searched terms. For example, if a user wants to search resources related to the terms "*forties, coin*", he might not be able to retrieve resources tagged with "*1940s, penny*" or "*1944, cent*".

Resources not tagged with sufficient number of relevant tags make the data in a folksonomy very sparse, which makes search in folksonomies difficult. We can reduce the sparseness in folksonomies by discovering semantically related tags and enriching the data in folksonomies. The common methods for finding semantically related tags use either co-occurrence information among tags or use the external data sources like Wikipedia or WordNet.

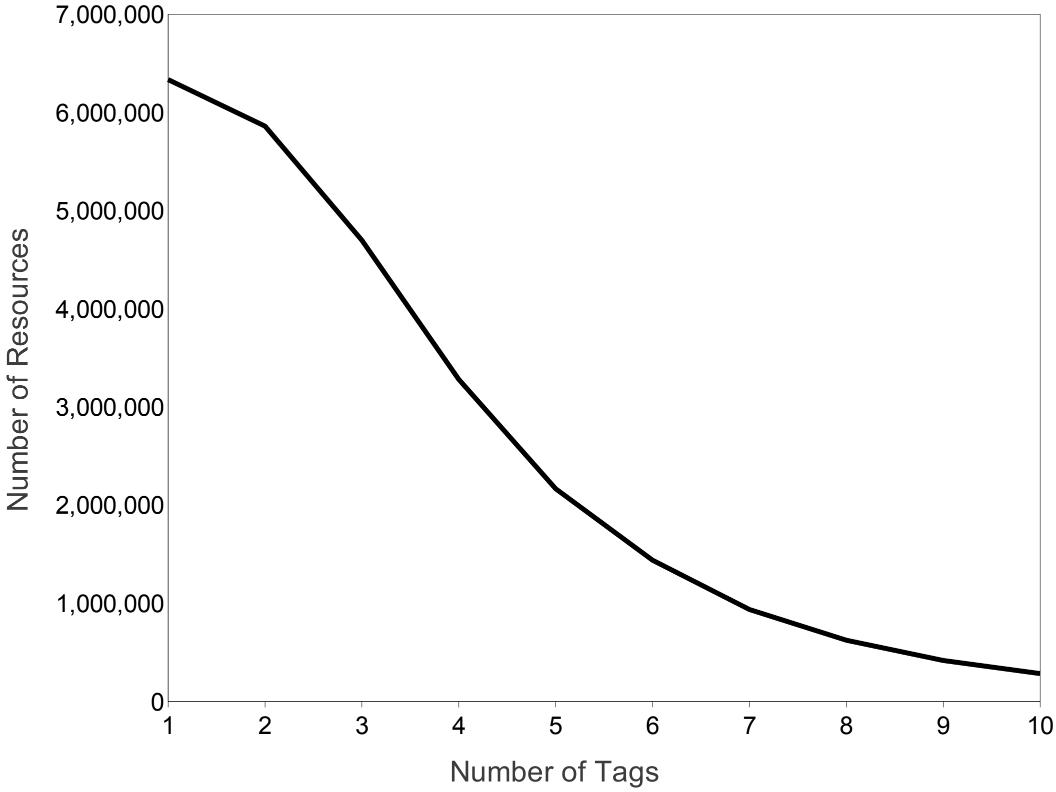

Figure 4.1: Number of tags associated with around 27 million resources uploaded to Flickr between the years 2004 and 2005.

In this chapter, we discuss the methods for discovering relationships between tags based on two dimensions, first the context of the tags and second the type of similarity between the tags. We consider two types of tag contexts, the resource context (which resources are assigned to a particular tag), and the user context (which users have used a particular tag). The resource context of tags helps in finding tags which are mostly used in similar kind of resources, whereas the user context finds broad relationships between tags based on the users' interests (represented by the tags they use). We also discuss different kinds of tag similarities like co-occurrence based, probabilistic, and heuristic based similarity as well as similarity among tags based on external data sources like WordNet and Data APIs.

4.2 Exploiting External Data Sources

Semantic relationships between tags or words can be retrieved using external data sources like thesauri or Data APIs.

4.2.1 WordNet

WordNet (Miller, 1995) is a lexical reference system which represents relationships between different lexical entities in English (nouns, verbs, adjectives, adverbs). In WordNet, words are grouped into different sets called "synsets". These synsets provide a method to identify semantically related tags. Given a tag, semantically similar tags can be identified by searching words in the synset of the given tag. For example, for the word *Jaguar*, a synset in WordNet includes the related words *panther*, *Panthera onca*, and *Felis onca*. The limitation of exploiting WordNet for identifying semantically related tags is its limited vocabulary. WordNet does not include domain specific words or non-standard tags created particularly in folksonomies.

4.2.2 Data APIs

Folksonomies like Flickr provide an extensive set of services using APIs. As an example, the Flickr API[1] can be used for finding semantically related tags.

[1] `http://www.flickr.com/services/api/flickr.tags.getRelated.html`, last accessed in October 2010

According to Flickr API website, the function *flickr.tags.getRelated* "returns a list of tags related to the given tag, based on clustered usage analysis". For any given tag, relevant tags can be retrieved by the API. In general, the main disadvantage of using APIs is the lack of knowledge about the methodology for finding semantically related tags. Changes in the methods used for finding related tags could result in different sets of related tags over a period of time.

4.3 Exploiting Internal Data of a Folksonomy

Co-occurrence based information can also be used for discovering semantically related tags. We explore different co-occurrence methods based on similarity between tags. The relevant tags are identified by measuring similarity between the tags. We classify the similarities between tags[1] into the following three categories:

Co-occurrence: The simplest form of finding semantically similar tags is co-occurrence; two tags are relevant, if they co-occur in resources. If we represent the tag t_i with the tag vector $\vec{t_i}$, then simple co-occurrence between two tag vectors $\vec{t_1}$ and $\vec{t_2}$ can be computed using Equation 4.1, i.e. by counting, in how many resources a tag t_1 appears together with another tag t_2. The main disadvantage of using this simple co-occurrence measure is the lack of normalization. Very frequent tags like *sky*, *family* or *travel* co-occur with many of the tags in Flickr, and it leads to very obvious or unwanted relationships between tags. Normalizing co-occurrence with the frequencies of the tags helps in finding relevant tags which might not be very frequent globally. One way of normalization is using *cosine* similarity (see Equation 4.2), in which co-occurrence of tags is normalized by the *Euclidean norm* of the tag vectors. Other co-occurrence based similarity measures include *Dice* (see Equation 4.3) and *Jaccard* (see Equation 4.4) coefficients. The *Dice* coefficient gives higher value to co-occurring tags than the *Jaccard* coefficient. Jaccard coefficient penalizes tags which do not co-occur very frequently (Manning and Schütze, 1999).

[1]We represent the tag vector of the tag t_i without any contextual information (see Section 4.4.2) as $\vec{t_i}$.

$$simple(\vec{t_1}, \vec{t_2}) = \vec{t_1} \cdot \vec{t_2} \tag{4.1}$$

$$cosine(\vec{t_1}, \vec{t_2}) = \frac{\vec{t_1} \cdot \vec{t_2}}{\|\vec{t_1}\| \times \|\vec{t_2}\|} \tag{4.2}$$

$$dice(\vec{t_1}, \vec{t_2}) = 2\frac{\vec{t_1} \cdot \vec{t_2}}{|\vec{t_1}| + |\vec{t_2}|} \tag{4.3}$$

$$jaccard(\vec{t_1}, \vec{t_2}) = \frac{\vec{t_1} \cdot \vec{t_2}}{|\vec{t_1}| \times |\vec{t_2}| - \vec{t_1} \cdot \vec{t_2}} \tag{4.4}$$

The above mentioned measures provide semantically similar tags or synonyms based on co-occurrence information. If two tags appear together in many of the resources, then they are considered to be similar. For example, the tags *Brazil* and *Brasil* appear often together, therefore they are considered to be semantically similar tags. Similarly *Notre* and *Dame* can also be considered semantically similar tags.

Probabilistic: We also use a probabilistic model (Mutual Information) for discovering semantically related tags. *Mutual Information* (MI, see Equation 4.5) measures the association between two tag vectors. The measure of association depends upon the probability of two tags appearing together. It is more likely that the tags t_1 and t_2 appear together, the value of mutual information between their tag vectors will be high. Mutual Information among two tag vectors can be computed as follows:

$$MI(\vec{t_1}, \vec{t_2}) = \sum_{\vec{t_1}} \sum_{\vec{t_2}} p(\vec{t_1}, \vec{t_2}) \log\left(\frac{p(\vec{t_1}, \vec{t_2})}{p(\vec{t_1}) \cdot p(\vec{t_2})}\right) \tag{4.5}$$

Where $p(\vec{t_1})$ and $p(\vec{t_2})$ are the marginal probabilities of the tag vectors $\vec{t_1}$ and $\vec{t_2}$ respectively. $p(\vec{t_1}, \vec{t_2})$ is the joint probability between the tag vectors $\vec{t_1}$ and $\vec{t_2}$.

Heuristic: In addition to standard similarity measures (see Equations 4.1–4.4), we also explore two heuristics (*overlap* and *modified overlap coefficients*) for finding relevant tags. The value of the overlap coefficient (see Equation 4.6) between two tag vectors $\vec{t_1}$ and $\vec{t_2}$ is high, if one of the tags mostly appears with the other tag. For example, if the tag *sky* always appears with the tag *blue*, then the value of their overlap coefficient is equal to one.

We also propose a non-symmetric version of the overlap coefficient called *overlap_mod* (see Equation 4.7). The modified overlap coefficient identifies the relationship between a less common tag and a common tag. A less common tag is associated to the common tag, but not vice versa. For example, if the tag *Koblenz* appears in 50,000 resources and the tag *Germany* appears in 5 million resources, and both of these tags appear in 10,000 resources together, then the similarity value between the tag vectors of the tags *Koblenz* and *Germany*[1] using modified coefficient is $overlap\_mod(\vec{t}_{Koblenz}, \vec{t}_{Germany}) = 0.2$ $(10,000 \div 50,000)$. However, the similarity between the tag vectors of the tags *Germany* and *Koblenz* using modified overlap coefficient is $overlap\_mod(\vec{t}_{Germany}, \vec{t}_{Koblenz}) = 0$ $(|\vec{t}_{Germany}| \nleq |\vec{t}_{Koblenz}|)$.

$$overlap(\vec{t}_1, \vec{t}_2) = \frac{\vec{t}_1 \cdot \vec{t}_2}{min(|\vec{t}_1|, |\vec{t}_2|)} \tag{4.6}$$

$$overlap\_mod(\vec{t}_1, \vec{t}_2) = \begin{cases} \dfrac{\vec{t}_1 \cdot \vec{t}_2}{|\vec{t}_1|} & \text{if } |\vec{t}_1| \leq |\vec{t}_2|; \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

Examples of semantically related tags using the methods presented are shown in Table 4.1 for three different sets of tags. The dataset for discovering these semantically related tags was consisted of approximately 27 million resources and 92 thousand tags from Flickr (see Section 5.3.2). We have listed the top 5 relevant tags for each set of tags. We can observe that different types of similarities result in different semantically related tags. The example of WordNet shows its limitation of limited vocabulary. There are no relevant words for the tag *chevy* in WordNet, but based on global or local analysis, or on an external data source like Flickr API, we can minimize this limitation. We can see that all other methods than WordNet have the relevant word *Chevrolet* for the tag *chevy*.

[1] The tag vector $\vec{t}_{Koblenz}$ represents the tag *Koblenz* and $\vec{t}_{Germany}$ represents the tag vector of the tag *Germany*.

Table 4.1: Sample semantically related tags using different methods.

Method	**1929, chevy**	**lake, huntington**	**purple, flowers**
Co-Occ.	flood, iowa, cedarrapids, chevrolet, car, cars, automobile	water, 2005, nature, sunset, beach, garden, california	flower, green, nature, garden, macro
Cosine	cedarrapids, christman, memorium, royalyork, chevrolet, chrystler, supercharged, impala	water, steelhead, frankfort, frenzy, huntingtongardens, hage, skulptur, jardin	flower, pink, violet, garden, plants, nature
Dice	cedarrapids, christman, memorium, modela, chevrolet, corvette, impala, carshow	water, michigan, mountains, tahoe, huntingtongardens, hage, skulptur, jardin	flower, pink, green, garden, nature, plants
Jaccard	cedarrapids, christman, memorium, modela, chevrolet, corvette, impala, viper	water, michigan, fishing, mountains, huntingtongardens, hage, skulptur, jardin	violet, catchycolors, iris, orchid, garden, flower, plants, macro
MI	flood, iowa, ford, memorium, chevrolet, car, cars, automobile	water, tahoe, district, michigan, garden, gardens, beach, giardino	flower, green, pink, garden, nature, plants
Overlap	christman, memorium, royalyork, car, chrystler, hhr, supercharged, chevrolet	peyto, jarvi, natsuki, huntingtongardens, hage, dieu, myths	roxo, paars, purplerain, beautyberry, daises, genitalia, loreak, gerberas
Flickr	chavorlet, car, classic, truck	water, trees, nature, reflection, beach, california, pier, surf	flower, pink, blue, nature, spring
WordNet			flower, blossom, bloom, peak, prime

4.4 Dimensions of Similarity among Tags

We define two dimensions of semantic relationships between tags and propose enriched vector space models based on these dimensions. Following sections describe each dimension in detail:

4.4.1 Type of Similarity

Semantic relatedness between tags based on different similarity measures / heuristics provide different types of associations among tags. For example, similarity between two tags based on WordNet could be different from similarity based on cosine measure. Even different co-occurrence based similarity measures / heuristics provide different types of similarities. The strength or weight of similarity between two tags based on two different measures could be different. Therefore we consider the methods of finding semantically similar tags as one dimension, called "Tag Similarity". Different types of similarity measures based on external and internal data sources are described in the previous Sections 4.2 and 4.3 respectively.

4.4.2 Context of the Tag

In addition to the type of tag similarity, the context of the tag is another dimension for finding semantically related tags. We describe the two different kinds of contexts of the tags as follows:

Resource Context: The resource context of a tag t_i consists of all the resources that are annotated with the tag t_i. We formally represent the resource context of the tag t_i as a resource vector $\vec{r}_{i,*}$ (see Equation 3.3) . To discover the semantically similar tags t_i and t_j based on the resource context, we can compute the similarity between them using their resource contexts. Similarity between two tag vectors $\vec{r}_{i,*}$ and $\vec{r}_{j,*}$ is computed using the Equations 4.1–4.7.

User Context: We hypothesize that tags represent the interests of the users. Users would usually add tags to the resources in which they are interested in. A set of users sharing a particular set of tags reflects their common interest in some resources or subjects. The user context of a tag consists of all the users that share the same tag. We can exploit the user context for discovering semantically related tags. For example, if many users annotate different resources with the tags *coin* and *cent*, and they do not use these two tags together in any of the resources they annotate, it would still be possible to find relationships between these tags by considering the number of users that shared both of these tags. We formally represent the user context of the tag t_i as a vector $\vec{u}_{i,*}$ (see Equation 3.2), which represents all the users who have used the tag t_i. Similarity between two tag vectors based on their user context can be computed using the Equations 4.1–4.7.

Table 4.2 shows some examples of semantically related tags using the methods described in this section with respect to different contexts. The examples are based on a dataset of around 27 million resources uploaded to the Flickr website during the years 2004 and 2005.

Table 4.2: Semantically related tags based on resource and user tag contexts.

Tag	**Resource Context**	**User Context**
brick	wall(0.11)	wall(0.37) fence(0.36) window(0.36) rust(0.35)
bromelia	airplant(0.32) bromeliad(0.17) tillandsia(0.15)	lirio(0.18) tibouchina(0.15) soneca(0.15) strelitzia(0.15)
designs	desktops(0.29) wallpapers(0.22) backgrounds(0.21)	
madrid	spain(0.19) zarzuela(0.13) hipodromo(0.13) carreras(0.12)	spain(0.31) espaa(0.24) segovia(0.22) toledo(0.21)
pub	crawl(0.17)	beer(0.25) bar(0.24) sign(0.22) london(0.22)
seventies	70s(0.16) entertainers(0.14) sixties(0.13)	sixties(0.19) 70s(0.17) eighties(0.16) forties(0.13)
spain	espaa(0.37) barcelona(0.25) andalucia(0.20) madrid(0.19)	barcelona(0.36) espaa(0.32) madrid(0.31) gaudi(0.27)
style	crave(0.14) arian(0.13) fashion(0.11) persians(0.10)	fashion(0.16) hair(0.13) woman(0.12) man(0.12)

The different types of relationships using different contexts can be observed from the examples. The resource context discovers close relationships among tags, for example, the tag *brick* is similar to the tag *wall*. The tags *bromelia* is similar to the tags *airplant*, *bromeliad*, and *tillandsia* which belong to the same family of flowers. If we consider the user context of the tags, we observe a wide range of tag associations. For example, we find the tags *tibouchina* and *strelitzia* associated to the tag *bromelia* which are different kinds of flower plants. User context also associate the tags *seventies* to the tags *sixties*, *eighties*, and *forties*, which shows the interest of users in old pictures. In the next chapter, we exploit the semantically related tags for enriching the data in a folksonomy.

4.5 Conclusions

This chapter discusses different methods of discovering semantically related tags. Semantically related tags can be exploited to enrich the sparseness in folksonomies. Reducing sparseness in folksonomies would enable us to retrieve resources which are otherwise not retrieved due to sparseness. As detailed in this chapter, the next chapter discusses how reducing sparseness in folksonomies can enable us to improve search in folksonomies.

Chapter 5

Exploiting Semantics for Improving Search

In order to access and share resources, users add tags to the resources. The tags are freely chosen keywords and it is not possible for the users to tag their resources with all the relevant tags for obvious reasons. As a result, most resources are not annotated with the majority of tags relevant to them. The lack of relevant tags results in sparseness of folksonomy data, and this sparseness of data makes many relevant resources unsearchable. To overcome the problem of search in folksonomies, we propose the enriched vector space models which exploit the semantically related tags discussed in Chapter 4. The enriched vector space models are less sparse and help in improving search in folksonomies. We evaluate the methods proposed in this chapter on a large dataset. The dataset consists of around 27 million resources and 92 thousand tags, 150 queries are used for the evaluation done by 18 users. Experimental results based on the large scale evaluation show that the enrichment of the existing data by exploiting semantic relationships among tags helps in improving the search results, particularly for the queries for which only few relevant resources exist in the original data.

5.1 Sparseness in Folksonomies

In folksonomies users are free to choose tags for annotating their resources. A lot of resources are not tagged with many of the relevant tags. An analysis of a large folksonomy dataset of 54 million resources shows that there are on average only *3.1* tags associated with each resource (Bolettieri et al., 2009). This makes the data in a folksonomy very sparse. The sparseness of the data makes it difficult to search resources. For example, if a picture of a penny of 1972 is tagged with *penny* and *1972* instead of the tags *seventies* and *coin*, and a user searches for the resources using keywords *seventies* and *coin*, he will not discover the resource because this resource is not tagged with the tags *seventies* and *coin*. However, if we associate the tag *coin* with the tag *penny* and the tag *seventies* with the tag *1972*, then it would be possible to retrieve the resources which are tagged with *1972* and *penny* using the query *seventies* and *coin*.

We hypothesize that there are many resources in a folksonomy which are not searchable because they are not associated with most of the relevant tags. By adding relevant tags to the resources, it is possible to search resources which are otherwise not searchable. We associate the relevant tags to the resources using a linear transformation function to enrich the standard vector space model of a folksonomy. We consider several methods to enrich the vector space model depending upon the type of similarity between tags and the context of the tags. The basic idea behind all the proposed methods is to discover the semantically related tags and then enrich the standard vector space model using the discovered semantically related tags.

5.2 Exploiting Semantically Related Tags

To exploit the semantically related tags for enriching the sparse vector space model of folksonomies, we define a linear transformation function $L_T : V \rightarrow E$ based on the type of similarity (see Section 4.4.1) and context of the tags (see Section 4.4.2).

$$L_{T_C^S}(R) = T_C^S \times R \tag{5.1}$$

Where R is the simple vector space model for a folksonomy as defined in Equation 3.3 and T_C^S is a matrix representing the similarity between the tags based on the similarity type (S) and the context (C) of the tags. The matrix T_C^S is always a square matrix with both dimensions equal to the number of tags in the folksonomy. Each value at the i^{th} and j^{th} index of the matrix T_C^S is the similarity between the tags i and j. The matrices T_C^S based on similarity measures/coefficients and the resource context can be computed as follows:

$$T_R^C = [cosine(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.2}$$
$$T_R^S = [simple(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.3}$$
$$T_R^D = [dice(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.4}$$
$$T_R^J = [jaccard(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.5}$$
$$T_R^M = [MI(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.6}$$
$$T_R^O = [overlap(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.7}$$
$$T_R^P = [overlap\_mod(\vec{r}_{i,*}, \vec{r}_{j,*})] \tag{5.8}$$

To exploit the user context, we can use the user based vector space model (see Equation 3.2).

$$T_U^C = [cosine(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.9}$$
$$T_U^S = [simple(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.10}$$
$$T_U^D = [dice(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.11}$$
$$T_U^J = [jaccard(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.12}$$
$$T_U^M = [MI(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.13}$$
$$T_U^O = [overlap(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.14}$$
$$T_U^P = [overlap\_mod(\vec{u}_{i,*}, \vec{u}_{j,*})] \tag{5.15}$$

To represent the context of the tags in the similarity measures, we changed the notation of tag vectors from $\vec{t_i}$ (as used in Eqs. 4.1–4.7) to $\vec{r}_{i,*}$ or $\vec{u}_{i,*}$ (as defined in Chapter 3). The vectors denoted by $\vec{r}_{i,*}$ or $\vec{r}_{j,*}$ denote the tag vectors based on the resource context, whereas the vectors denoted by $\vec{u}_{i,*}$ or $\vec{u}_{j,*}$ represent the tag vectors based on the user context.

Any representation of the matrix T_C^S from Equation 5.2 to Equation 5.15 can be replaced in the Equation 5.1 to compute an enriched vector space model. Some examples of the enriched vector space model are given as follows:

$$L_{T_R^C}(R) = T_R^C \times R \qquad \textit{(resource context / cosine similarity)} \tag{5.16}$$

$$L_{T_R^P}(R) = T_R^P \times R \qquad \textit{(resource context / overlap-mod co-eff.)} \tag{5.17}$$

$$L_{T_U^C}(R) = T_U^C \times R \qquad \textit{(user context / cosine similarity)} \tag{5.18}$$

After the transformation of the original vector space model R into the enriched vector space model, the semantically related tags are assigned to the resources in the enriched vector space model. As an example, assume that Table 5.1 shows the original vector space model. The matrix T_R^C computed using the resource context and the cosine similarity (see Equation 5.2) is shown in Table 5.2. After transforming the original vector space model R (see Table 5.1) into an enriched vector space using the Equation 5.1, the resulting enriched vector model space is shown in Table 5.3.

Table 5.1: An example original vector space model.

	r_1	r_2	r_3	r_4
1972	1	1	0	0
coin	0	1	0	0
flower	0	0	1	0
hibiscus	0	0	1	1
nature	0	0	0	1
penny	1	0	0	0
seventies	0	1	0	0

We can observe that some of the missing relevant tags are now added to the enriched vector space model, as the tags *1972* and *seventies* are semantically related in Table 5.2 (representing the matrix T_R^C). Therefore, they are assigned to the resource r_1 in the enriched vector space model (see Table 5.3). Similarly, the tag *flower* is assigned to the resource r_4 because it is semantically related to the tag *hibiscus* in the matrix T_R^C showing semantic relationships between tags (see Table 5.2).

Table 5.2: An example matrix, showing semantically related tags. The matrix is computed using the resource context and the cosine similarity.

	1972	**coin**	**flower**	**hibiscus**	**nature**	**penny**	**seventies**
1972	1	0.71	0	0	0	0.71	0.71
coin	0.71	1	0	0	0	0	1
flower	0	0	1	0.71	0	0	0
hibiscus	0	0	0.71	1	0.71	0	0
nature	0	0	0	0.71	1	0	0
penny	0.71	0	0	0	0	1	0
seventies	0.71	1	0	0	0	0	1

Table 5.3: An example enriched vector space model.

	r_1	r_2	r_3	r_4
1972	1.71	2.41	0	0
coin	**0.71**	2.71	0	0
flower	0	0	1.71	**0.71**
hibiscus	0	0	1.71	1.71
nature	0	0	**0.71**	1.71
penny	1.71	**0.71**	0	0
seventies	**0.71**	2.71	0	0

Ranking Relevant Resources against a Query:

In order to rank the relevant resources against a query, we adopt a two-step ranking strategy. The ranking strategy is a mixture of exact match and cosine similarity. The same ranking strategy is used for all the vector space models and baselines. In the first step the resources having the maximum number of queried tags are ranked higher, followed by the resources having fewer queried tags. This is an enhancement of the simple exact match in which all the search results must be tagged with the queried tags. But using enhanced exact match, we also retrieve the resources which are not tagged with all of the queried tags and these resources are ranked below the resources which are tagged with all of the queried tags. Formally, the rank of each resource is computed as follows:

$$C_{j=1..N} = |\vec{q} \cap \vec{r}_{*,j}| \tag{5.19}$$

Where $\vec{C}$ is the common term vector representing the number of common terms between the query q and the resource j, $\vec{q}$ represents the query vector, N is the total number of resources in the vector space model, $\vec{r}_{*,j}$ represents the j^{th} resource in the vector space model and $|\vec{q} \cap \vec{r}_{*,j}|$ represents the number of common tags between the query q and the j^{th} resource[1]. The resources retrieved against the query q are ranked in descending order of the values of the vector $\vec{C}$.

Based on exact match, there can be a situation where many resources are ranked equally (having the same number of queried tags) and as we do not compute the global rank of each resource (as in case of PageRank), we need to further rank the resources which contain the same number of queried tags. For this purpose, in the second step of ranking, the resources having same number of queried tags are ranked based on their cosine similarity to the query using the Equation 4.2.

As an example of ranking resources, if a query q consists of the tags t_1, t_2, and t_3, then the resources having all these tags t_1, t_2, and t_3 are ranked the highest, followed by the resources which contain any two of these tags ("t_1 and t_2" or "t_1 and t_3" or "t_2 and t_3"). The resources having only one of the queried tags (t_1, t_2, or t_3) are ranked the lowest. If there are more than one resource which contain same number of queried tags, they are ranked based on their cosine similarity with the query. For example, if the resources r_1 and r_2 both have the queried tags t_1, t_2, and t_3 and the cosine similarity between the resource r_1 and the query q is 0.7 and the cosine similarity between the resource r_2 and the query q is 0.5,

[1]Note that the intersection symbol ($\cap$) in Equation 5.19 is used to count the number of common terms between the vectors $\vec{q}$ and $\vec{r}_{*,j}$. If q_i represents the i^{th} tag in the query and $r_{i,j}$ represents the i^{th} tag in the j^{th} resource, then the value of $q_i \cap r_{i,j}$ would be equal to 0 if the value of either q_i or $r_{i,j}$ is equal to zero and 1 otherwise.

then the resource r_1 is ranked higher than the resource r_2.

5.3 Evaluating Enriched Vector Space Models

To evaluate the use of enriched vector space models for improving search in folksonomies, we used real life queries from a query log of a web search engine. The search results for the queries were retrieved from a dataset of around 27 million resources. 18 users evaluated the results for different vector space models. Following are the details of the evaluation methodology and the dataset:

5.3.1 Evaluation Methodology

For doing experiments on real life queries, we used the AOL query log (details in (Pass et al., 2006)). This log originally contained 20 million queries from 650,000 users during three months from March to May 2006. Out of these 20 million queries, we randomly selected those queries for which a user had clicked on a link to the Flickr website. We split the queries into three sets depending upon the number of relevant resources in the original dataset. Each query set had 1 to 10, 11 to 50, or more than 50 exact matches (resources having all the queried tags) in the original dataset. We randomly selected 50 queries from each of these three sets, resulting in 150 total queries for the evaluation.

The results were evaluated by 18 users (mostly PhD students). Each user was shown a search result page similar to the screenshot shown in Figure 5.1. The query was shown at the top of each evaluation page with resources retrieved as a result. The title of each resource was shown at the top of the resource, the tags on the right side, and the evaluation options at the bottom of each resource. Every user was given a set of queries and results obtained using different vector space models as described in Section 5.2. Users were unaware of the method used for creating the search result page. They were asked to mark a resource as *very relevant* or *relevant* if the resource matched the query, mark as *don't know* if they were not sure about the resource, *irrelevant* or *very irrelevant* if the resource did not match the given query. Queries were randomly distributed among users. The resources marked as *relevant* or *very relevant* were considered as relevant and others as irrelevant in the evaluation.

Human-based evaluation on a large scale dataset limited the scope of experiments. Out of all possible enriched vector space models, we selected three of them, first based on *cosine* similarity and *resource* context (see Equation 5.16), represented by $L_{T_R^C}$, second based on *modified overlap* coefficient and *resource* context (see Equation 5.17), represented by $L_{T_R^P}$ and the third based on *cosine* similarity and the *user* context (see Equation 5.18), represented by $L_{T_U^C}$.

In addition to the enriched vector space models, we also defined two other vector space models, *Semi Random* (Semi_Rand) as the baseline and the *Best of Breed* (BB) as the best meta enriched vector space model against a particular type of query. For the *Semi_Rand* vector space model, the semantic relationship matrix was created by associating random tags to each of the tags. The random values generated are uniformly distributed on the interval (0,1). The number of random tag associations generated for each tag is equal to the average number of tag associations in the enriched vector space models. The similarity of each tag to itself was explicitly set to 1 (maximum) in the *Semi_Rand* vector space model.

The *Best of Breed* meta enriched vector space model selects an appropriate enriched vector space model based on the type of the query. Based on empirical analysis the *Best of Breed* model uses the enriched vector space model based on the *cosine* similarity and the *user* context for the queries having *1 to 10* or *11 to 50* search results (exact matches) in the original data. For the queries having more than 50 search results, the *Best of Breed* model uses the enriched vector space model based on the *resource* context and the *cosine* similarity.

5.3.2 Dataset

The dataset we crawled consists of resources uploaded to Flickr during the years 2006 and 2007 to create a large-scale dataset[1]. The target of the crawling activity was the core elements, namely users, tags, resources and tag assignments. The statistics of the crawled dataset are summarized in Table 5.4.

[1]The reference dataset used for this evaluation is available at `http://west.uni-koblenz.de/Research/DataSets/PINTSExperimentsDataSets/`, last accessed in October 2010

Figure 5.1: A screenshot of an evaluation page.

users	tags	resources	tag assignments
319,686	1,607,879	28,153,045	112,900,000

Table 5.4: Flickr dataset statistics.

We applied the following strategy to crawl the Flickr dataset. First, we started a tag centric crawl of all resources that were uploaded between January 2004 and December 2005 and that were still present in Flickr as of June 2007. For this purpose, we initialized a list of known tags with the tag assignments of a random set of resources uploaded in 2004 and 2005. After that, for every known tag we started crawling all resources uploaded between January 2004 and December 2005 and further updated the list of known tags. We stopped the process after we reached the end of the list.

We filtered our dataset by removing those tags which were used by less than 10 users. Those users and resources were also removed from the dataset who did not use any tag. The final dataset was consisted of approximately 27 million resources, 300,000 users, and 92,000 tags. The exact statistics of the dataset are shown in Table 5.5. We conducted all our experiments presented in current chapter on the filtered dataset.

users	tags	resources	tag assignments
317,260	92,460	26,801,921	94,499,112

Table 5.5: Flickr filtered dataset statistics.

5.3.3 Results and Discussion

For evaluating the enriched vector space models, we used the standard *Precision at k* method as the evaluation measure. We computed precision at 5, precision at 10, precision at 15, and precision at 20 for each of the methods and each query set. In each of the results, the original vector space in the results show the average precisions obtained without enriching the vector space model. *Res./Cosine*, *Res./Overlap-M*, and *User/Cosine* show the average precisions obtained using the enriched vector space models $L_{T_R^C}$, $L_{T_R^P}$, and $L_{T_U^C}$ respectively. *Semi_Rand* shows the baseline results obtained using *SEMI_ Rand* vector space model (described in the previous subsection). *BB* represents the *Best of Breed* model (also described in the previous subsection). Note that the results in all the figures are shown in this order *Original*, *Res./Cosine*, *Res./Overlap-M*, *User/Cosine*, *Semi_Rand*, and *BB*.

Figure 5.2 shows the average precisions achieved for the queries which had 1 to 10 exact matches (resources associated with all the queried tags) in the original vector space. The results on each of the evaluation page were ranked using the ranking method described in Section 5.2. We achieve 0.4 to 0.45 precision at 15 and 20 for the original vector space model. This is due to the reason that retrieved resources are still associated with some of the queried tags, hence making these resources relevant.

We observe a significant improvement in the precision at all levels using the enriched vector space models, especially using the vector space model based on the cosine similarity and the user context (User/Cosine), which is also used in the *Best of Breed* for queries having 1 to 10 exact matches. The reason for the improvement in precision is the retrieval of those resources which do not contain the queried tag(s) exactly, but have some relevant tag(s). If we consider arbitrary tag relationships (*Semi_Rand*), then we get even worse results than the original vector space model. That suggests that the tags must be semantically related to improve the resource retrieval.

Figure 5.3 shows the results of precision values at different levels for the queries having 11 to 50 exact matches in the original vector space model. We observe a slight decrease in the performance of the enriched vector space models when compared to the original vector space model for precision at 5. But if we consider higher precision levels (15, 20), the results of the enriched vector space models are better than the results obtained from the original vector space model. Particularly, the model based on the user context and the cosine similarity performed better than all other methods and is also used in the *Best of Breed* model for the queries having 11 to 50 exact matches.

Figure 5.4 shows the results for the queries having more than 50 exact matches in the original vector space model. The overall performance of all the methods remains almost the same. The precision at 5 measure for the enriched vector space model based on the resource context and the modified overlap coefficient (Res./Overlap-M) is slightly higher. This is because some resources displayed in the top 5 search results were more relevant than the results from the original vector space model. For example, for the query *blue, bedroom*, the 3rd and the 4th resources displayed for the original vector space model had the tags *blue, bed-*

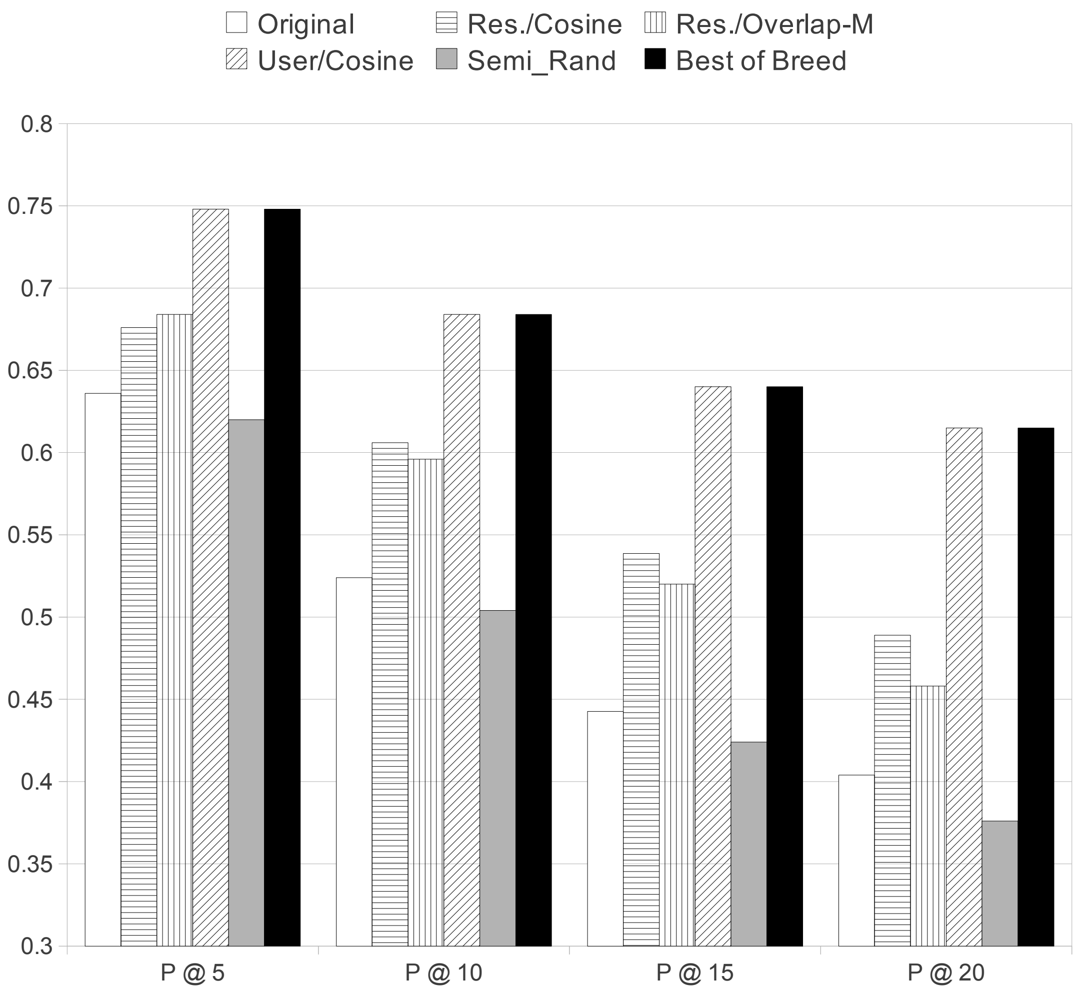

Figure 5.2: Results of precision at 5, 10, 15 and 20 for the evaluation of the queries having 1 to 10 relevant resources.

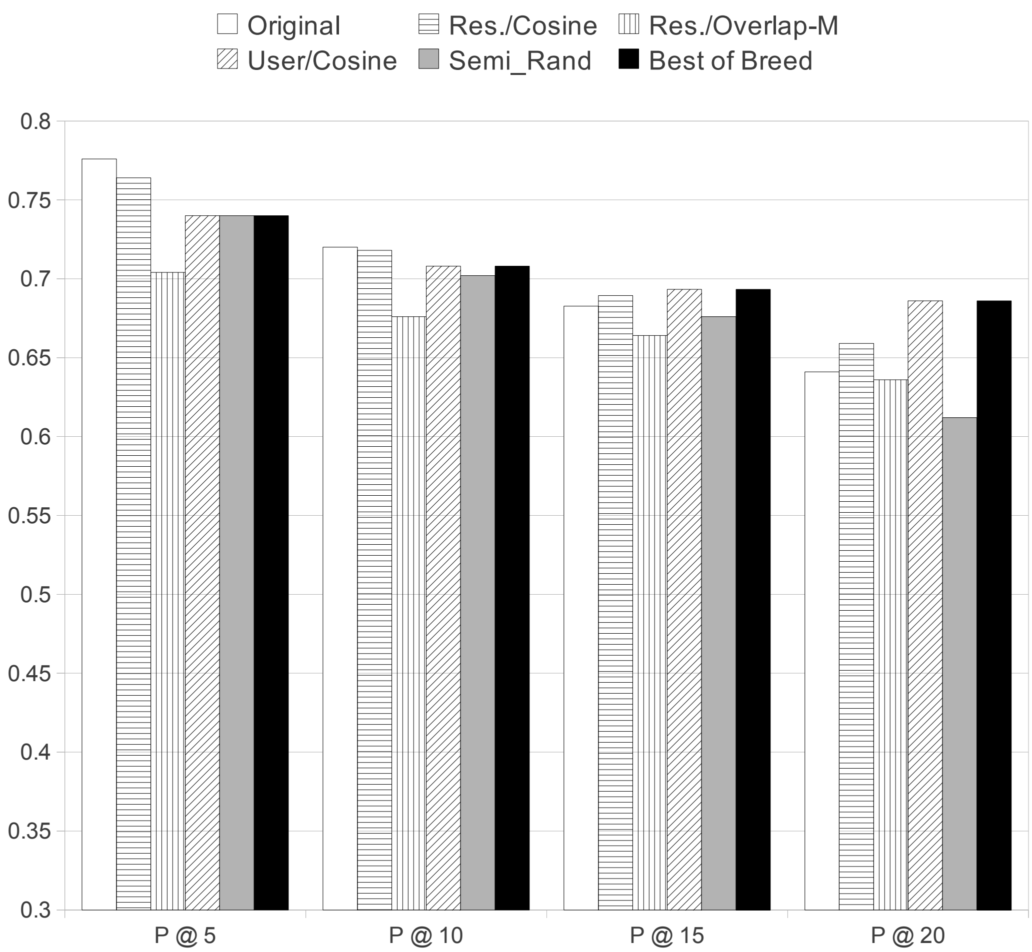

Figure 5.3: Results of precision at 5, 10, 15 and 20 for the queries with 11 to 50 relevant resources.

room, plant and *blue, bedroom, self-portrait* respectively. Whereas the resources displayed for the vector space model based on resource context and modified overlap coefficient (Res./Overlap-M) at 3rd and 4th positions had the tags *blue, bedroom, home* and *blue, bedroom, house*. The tags *home* and *house* were more relevant to the queried keyword *bedroom*, as compared to the tags *plant* and *self-portrait* associated to the resources in the original vector space model. This also suggests that using the enriched vector space models also help in ranking relevant resources higher where we already have many exact matches for the query in the original data. Compared to other methods for precision at 20, the model based on resource context and the cosine similarity performed better and is also used in the *Best of Breed* model for queries having more than 50 exact matches.

Figure 5.5 shows the results of all the queries used for evaluation. We represent the appropriate vector space model against a particular type of query as the *Best of Breed* (BB) model. This model performs better than other methods and achieves an improvement of 15% when comparing its results to the original vector space model for precision at 20.

The main goal of this research is to improve search, particularly for queries having only few relevant resources by enriching the vector space models. For the queries which had 1 to 10 exact matches in the original vector space model, we achieve great improvement in results using the enriched vector space models, which shows the significance of our proposed models. Figure 5.6 compares the performance of all methods for queries having varying number of relevant resources. The X axis represents the types of the queries.

We can observe that the enriched vector space models, particularly the best selection *Best of Breed* model, perform better than the baselines and the original vector space model. For queries having 1 to 10 relevant resources, we achieve an improvement of 35%. The improvement decreases for the queries having many relevant resources in the original vector space model (7% for 11 to 50 resources and 1.5% for more than 50 relevant resources). The reason for the decrease in improvement is that there are sufficient relevant resources in the original vector space model to be ranked in the top 20 results. If we consider the results of all the queries together, we still get an improvement of 12% using the *Best of Breed* model.

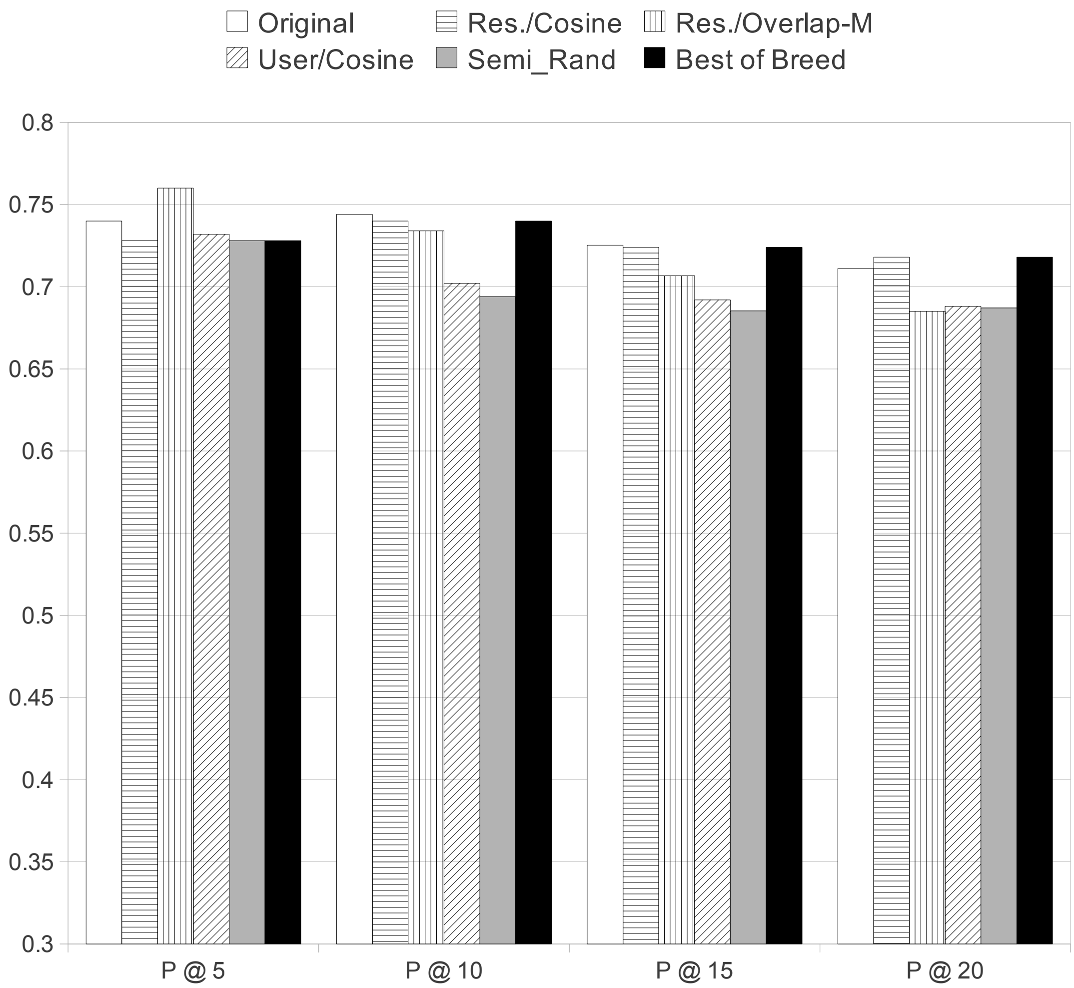

Figure 5.4: Results of precision at 5, 10, 15 and 20 for queries having more than 50 relevant resources.

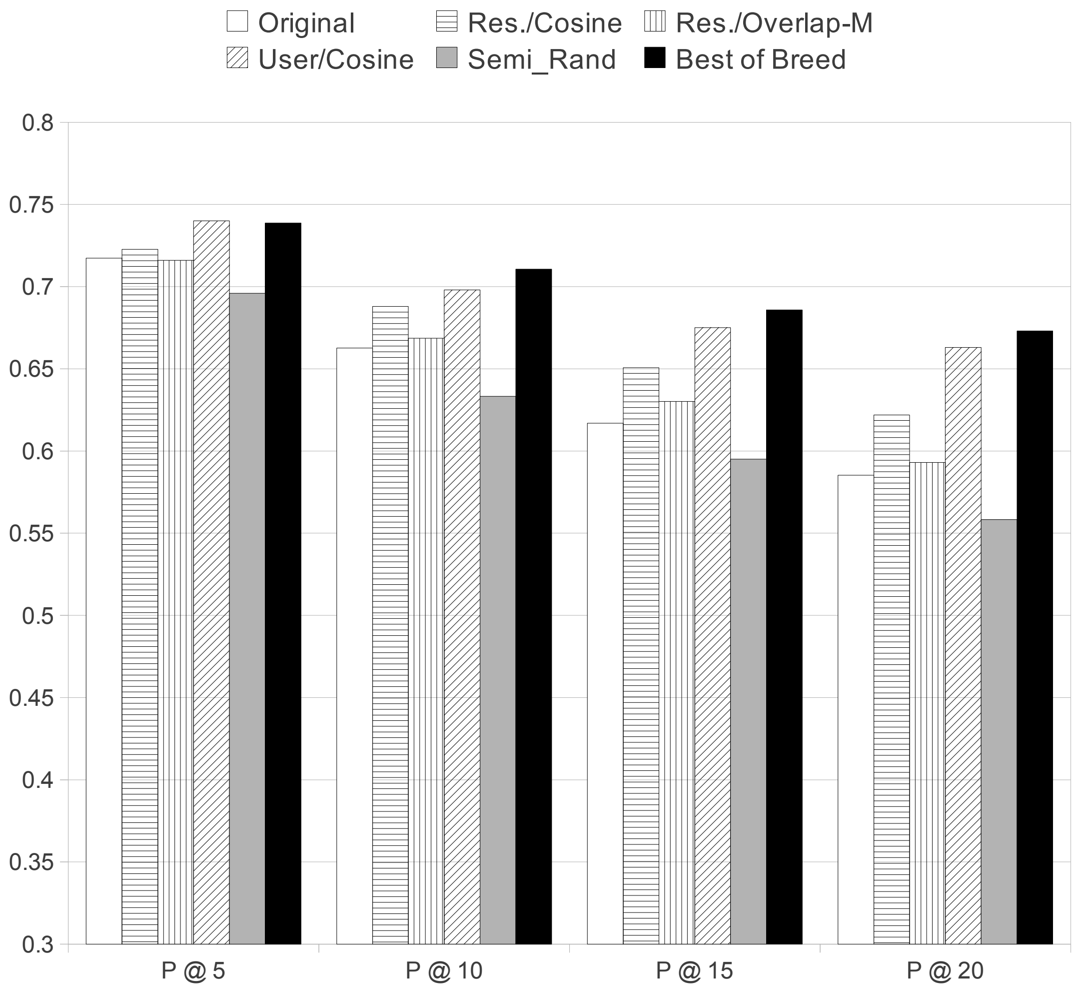

Figure 5.5: Results of precision at 5, 10, 15 and 20 for all the 150 queries. The Best-of-Breed (BB) method performs better than all other methods at all precision levels.

We also performed statistical significance tests (Student's t-test) of results achieved through the enriched vector space models and the original vector space model. When considering search results for all queries, the results were significantly different for precision at 10, 15, or 20 with p ranging from 0 (P at 20) to 0.003 (P at 10). However, the results were not significantly different for precision at 5 with $p = 0.11$. This was due to the fact that most relevant results are listed at the top for all the methods. We achieve significantly different results for precision levels higher than 5.

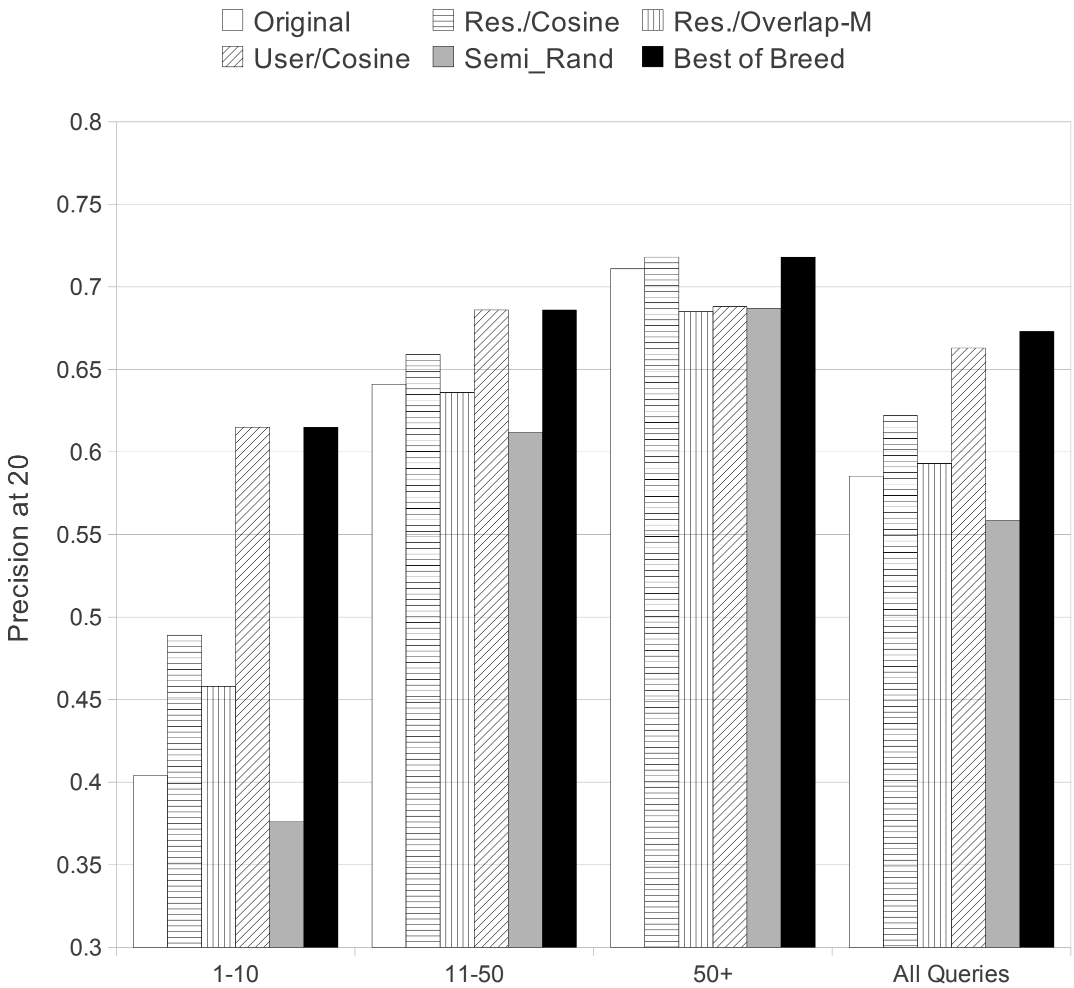

Figure 5.6: Comparison of methods for different types of queries. The X axis shows the number of relevant resources for the evaluated queries and the Y axis shows the results for precision at 20. The results are most significantly visible for the queries having 1 to 10 relevant resources.

5.4 Conclusions

In this chapter we have shown that it is possible to improve search in folksonomies by reducing the sparseness in the data. We have proposed methods to exploit the semantically related tags having different types of relationships and contexts for improving search in folksonomies. By enriching folksonomies using semantically related tags, we have shown that the resources which are currently unsearchable can be retrieved. Human-based evaluation of the enriched vector space models has shown improvement in the search results, especially for the queries where many relevant resources are not retrieved due to the lack of relevant tag annotations. We have suggested using the appropriate vector space model against a query based on the number of relevant resources for that query. Experimental results have shown that such methods give an overall improvement in the search results.

Chapter 6

Exploiting Different Features for Tag Recommendation

Tag recommendation is the process of suggesting relevant tags for a given resource and a *tag recommender* is a system which recommends the tags. In a folksonomy without a tag recommender system, users have to manually annotate their resources which could be inconvenient and time consuming. A tag recommender system is therefore important for assisting users in the tagging phase.

Many of the existing tag recommendation methods exploit only the tagging information (Jäschke et al., 2007; Marinho and Schmidt-Thieme, 2008; Sigurbjörnsson and van Zwol, 2008). However, many folksonomies support multiple resource features like the geographical coordinates. In addition to the tags, these features can also be exploited for improving the tag recommendation. In this chapter, we compare three types of resource features for tag recommendation. The features we consider are *geographical-coordinates*, *low-level image descriptors* and *tags*. We investigate the performance of each of these features independent of each other. First, the existing data collection is clustered separately for the geographical coordinates, tags, and low-level features.

The rest of the chapter is structured as follows. Section 6.1 gives an overview about the tag recommendation system we propose to compare different resource features. In Section 6.2 the content description methods (features) used in social media are shortly explained, especially those used in our framework. Section 6.3 explains how generating image annotations works in our framework. In Section 6.4, we describe the dataset used for experiments and evaluate tag recommender systems of the architecture proposed at large-scale. The tests compare different image representation methods in terms of precision and recall in the process of tag recommendation. Section 6.5 concludes our investigations and the results presented in this chapter.

6.1 System Overview

We split the overall system for tag recommendation into two parts: training and tag recommendation. The system is trained based on the image features available in social media, once the system is trained, it is used for recommending tags for new images. Figure 6.1 shows the tag recommendation process. The training phase is shown in the bottom (shaded region) of the figure. The tag recommendation process is shown in the top (non-shaded region) of the figure. Following is the brief description of the training and the tag recommendation phases:

Training: In the training phase images are clustered (see Section 6.3.1) based on their features. A cluster contains homogeneous images depending upon the types of features used for clustering. For this research work, we considered geographical coordinates, low-level image features and tags as image features. As an example, a cluster based on geographical coordinates represents the images taken in a particular location. A cluster based on low-level image features contains images sharing a particular texture or a color like the images of a sea or a forest. A cluster based on tagging data represents resources related to high level concepts like concert or river. The clustering process used in this research work is described in Section 6.3.1. Representative tags of a set of homogeneous images are used to annotate new images. The method of identifying representative tags is described in Section 6.3.2.

Classification: For recommending tags to a new image, we map the image to its closest representative group of images and assign the representative tags (see Section 6.3.2) of the mapped group to the new image. The method of classifying an image to its closest cluster and recommending tags are described in

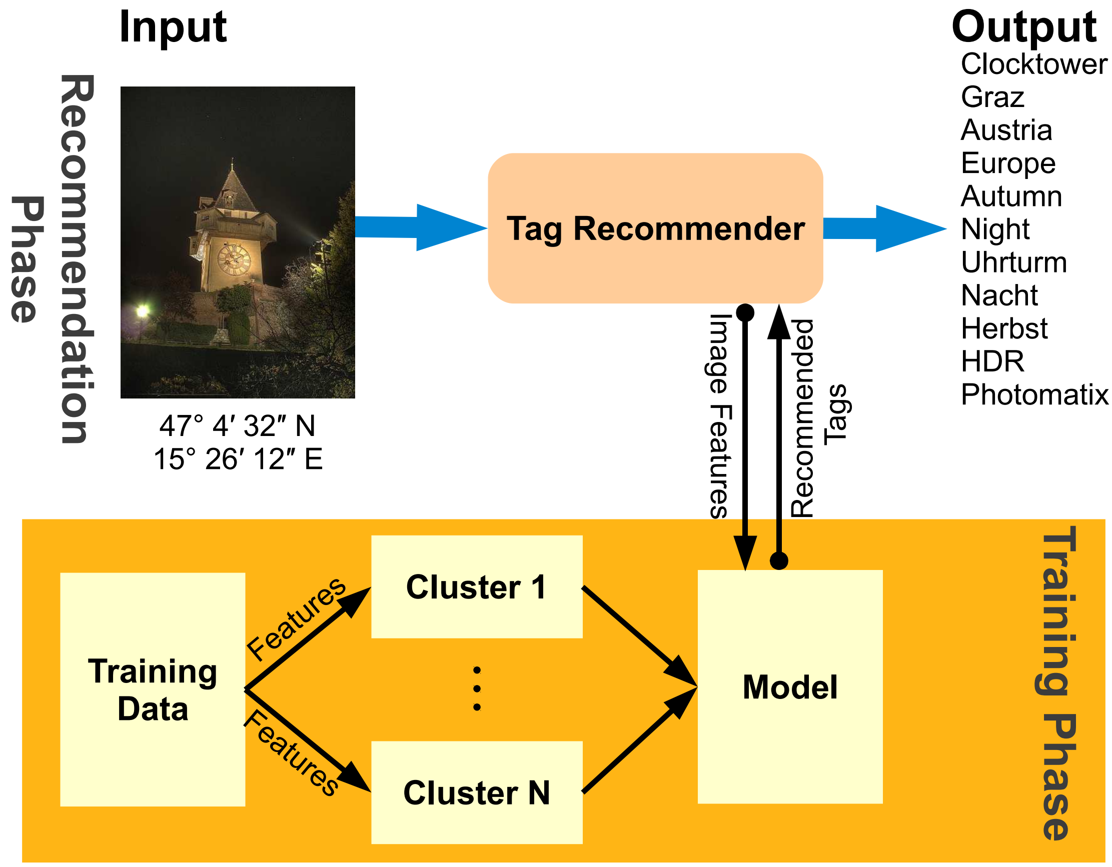

Figure 6.1: Overview of the tag recommendation system.

Section 6.3.3. In the following section, we describe the features that we have used in our experiments and are also available in folksonomies on a large scale.

6.2 Features in Social Media

To analyze the effect of different type of features on the performance of tag recommendation, we use three different image features in our experiments, namely *Geographical Coordinates* (G), *Low-level image features* (L), and *Tags* (T). Following are the details of the features used in this research work.

Geographical Coordinates: With the advancement in camera and mobile technologies, nowadays many devices are available in the market that are able to capture the location of the image using a built-in or an external GPS (global positioning system) device. In addition to the possibility of capturing the location of an image using a GPS device, some folksonomies like Flickr facilitate the users to add geographical coordinates to their images by providing a map interface where users can place their images on the map as shown in Figure 6.2. Due to this easiness, there are many images in Flickr which are enriched with geographical information. In the CoPhIR dataset (Bolettieri et al., 2009), around 4 million out of 54 million images are annotated with geographical coordinates. The number of geographically annotated images is supposed to increase in future as more devices will be able to capture the geographical coordinates. We represent the geographical coordinates of the images in a two dimensional vector space $G \in \Re^2$. Each row vector $\vec{g}_i$ of the feature space G represents the geographical coordinates of the image i.

Low-level Image Features: An image can be represented in a variety of low-level image features. Some of these features are represented using MPEG-7 multimedia content description standard (Manjunath et al., 2002). There are five different types of low-level MPEG-7 features available in the CoPhIR dataset for 54 million images. Table 6.1 shows the properties and dimensions of the low-level features available in the CoPhIR dataset. Based on initial experimental results, we consider two low-level features for evaluation, the MPEG-7 *Edge Histogram Descriptor* (EHD) and *Color Layout* (CL), which outperformed other available low-level image features. EHD represents the local edge distribution.

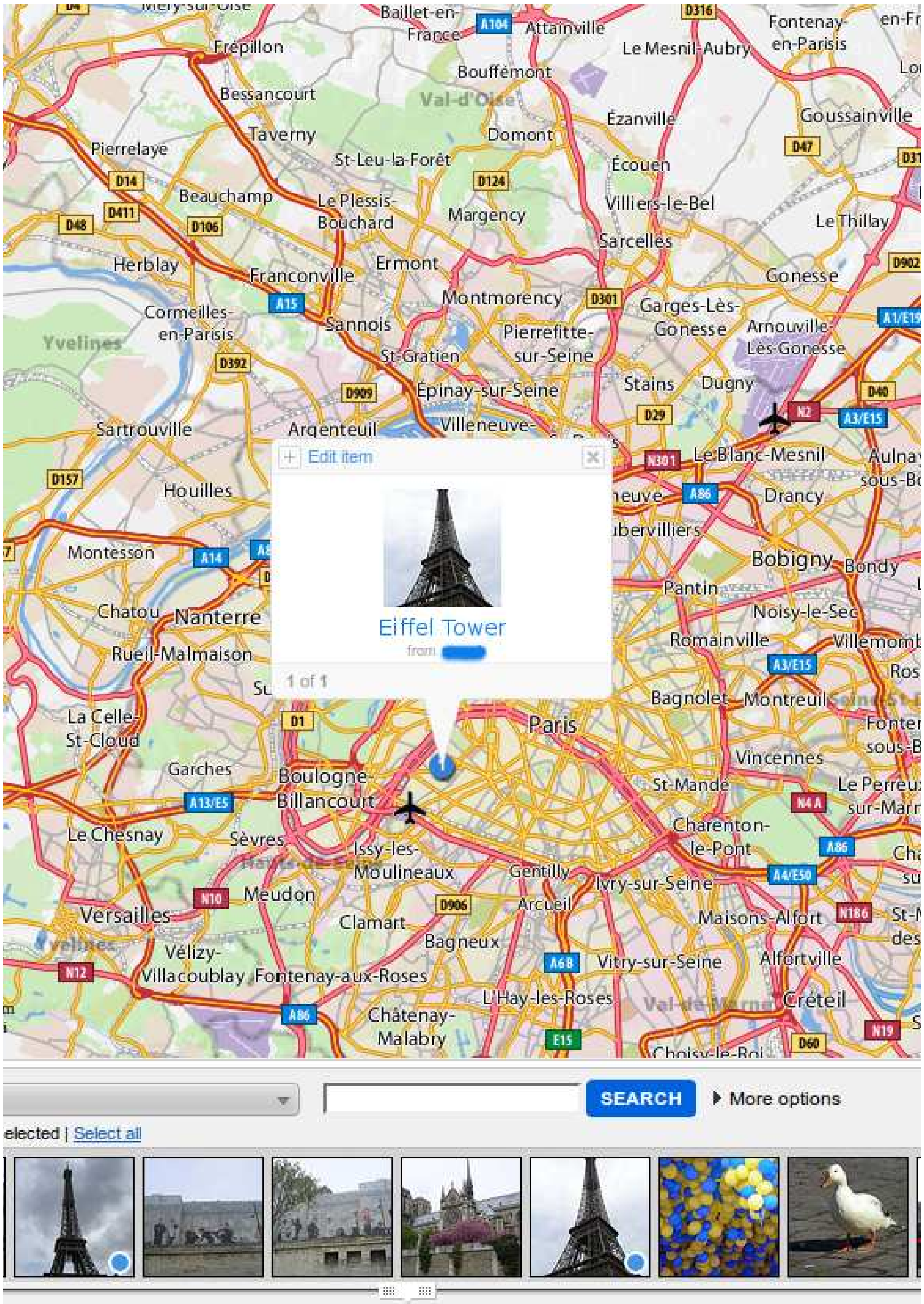

Figure 6.2: Screenshot of Flickr interface where users can add geographical coordinates to their images using a map.

Each image is divided into 4×4 non-overlapping blocks (resulting into 16 equal partitions). Edges in each block are categorized into five directions: vertical, horizontal, 45°diagonal, 135°diagonal and non-directional edges. The information about these edges is stored in a vector of 80 coefficients (Bolettieri et al., 2009). The Color Layout descriptor captures both color and spatial information. It is obtained by applying the discrete cosine transformation (DCT) on a 2-dimensional array of local representative colors in Y or Cb or Cr color space. The information is stored in 12 co-efficients (Bolettieri et al., 2009). The low-level image features based on EHD and CL are represented in 80 and 12-dimensional feature spaces $L_E \in \Re^{80}$ and $L_C \in \Re^{12}$ respectively. A row vector $\vec{\ell_i}$ of the feature space L_E or L_C represents the edge histograms or color layout of the image i respectively.

Table 6.1: Properties and dimensions of low-level features available in the CoPhIR dataset.

Low-level Feature	**Properties**	**Dims**
Scalable Color	Color histogram	64
Color Structure	Localized color distributions	64
Color Layout	Color and spatial information	12
Edge Histogram	Local-edge distribution	80
Homogeneous Texture	Texture	62

Tags: Tags are freely chosen keywords associated with the images. There is no restriction in selecting a tag for an image. A tag might represent a concept in an image, describe the image itself or it might also represent the context of the image (e.g. location, event, time etc.). On average there are only few tags associated with the images. In 54 million images of the CoPhIR dataset, each image has on average 3.1 tags. The tags are represented by a term by resource matrix T. A row vector $\vec{t}_{i,*}$ of the matrix T represents a resource. The non-zero values of the row vector $\vec{t}_{i,*}$ represent the tags associated with the resource i. A column vector $\vec{t}_{*,j}$ represents a tag vector whose non-zero values represent the resources associated with the tag j. A non-zero value of the matrix $T_{i,j}$ represents that the resource i is associated with the tag j.

To reduce the bias towards resources with many tags and very common tags, we assign Term Frequency (TF) and Inverse Document Frequency (IDF) weights to the resource by tag matrix T. TF and IDF are computed as follows:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{6.1}$$

Where $n_{i,j}$ is the number of times the tag t_i appears in the resource j, and the denominator is the sum of number of occurrences of all tags in the resource j.

$$IDF_i = \log \frac{N}{N_i} \tag{6.2}$$

Where N is the number of resources in the dataset and N_i is the number of resources in which the tag t_i appears.

6.3 Tag Recommendation

This section explains the proposed tag recommendation system in detail. In the training phase of tag recommendation, the resources are first clustered (see Section 6.3.1), then for each cluster, its representative tags are identified (see Section 6.3.2). In the tag recommendation phase, a new resource is mapped to its closest cluster and the representative tags of the closest cluster are recommended for the new image (see Section 6.3.3).

6.3.1 Clustering

We group (cluster) images to build a model which is used to assign tags to the given resources. Many possibilities exist for clustering a set of resources, which could affect the final quality of the recommended tags. For the presented tag recommender system, we adapt a well-known clustering technique called *K-Means* (MacQueen, 1967). K-Means is capable of clustering very large and high dimensional datasets. Of course, other clustering methods can also be employed in the framework, when one desires to fine tune the performances or improve the results. The K-Means algorithm we used is described in Figure 6.3.

In the following, we describe in detail how we set different parameters for using K-Means.

Input: Feature space $F \in \{G, L, T, D\}$, Number of clusters k
Output: A set of k clusters
Method:

1: Randomly select k vectors from the feature space F as initial cluster centroids
2: **while** values of cluster centroids are updated **do**
3:     **for** each resource r **do**
4:         find the cluster c whose centroid is closest to the resource r
5:         assign r to the cluster c
6:     **end for**
7:     **for** each cluster c **do**
8:         recompute the centroid of the cluster c based on the document assigned to it
9:     **end for**
10: **end while**

Figure 6.3: K-Means clustering algorithm.

Number of clusters: There is no generally accepted rule for setting the number of clusters for using K-Means. For our experiments, we use the number of clusters as suggested by Mardia et al. (1979, page 365). We define the number of clusters for n images as follows:

$$k = \sqrt{\frac{n}{2}} \tag{6.3}$$

By using k as defined in the above equation, we get the same number of clusters for each feature space.

Initial Cluster Centroids: In K-Means clustering, the quality of clustering also depends on the selection of initial cluster centroids. For our experiments, k images are randomly selected. The same sets of randomly selected images are used as cluster centroids for each feature space.

Computing distance/similarity between resources: During the clustering process, each image is assigned to its closest cluster (see Figure 6.3, step 4). We need a distance measure to compute the distance between an image and its closest centroid. The most popular distance measure used is *Euclidean Distance* (Han and Kamber, 2006, page 388). Euclidean Distance between two

m-dimensional vectors $\vec{f}$ and $\vec{c}$ is defined as follows:

$$euclidean(\vec{f}, \vec{c}) = \sqrt{\sum_{i=1}^{m} (\vec{f_i} - \vec{c_i})^2} \tag{6.4}$$

We use Euclidean distance for non-text feature spaces (i.e. geographical, and low-level feature spaces). For text (or tag) based feature spaces it is common to use *Cosine Similarity* (Han and Kamber, 2006, page 397). We use Cosine similarity to compute similarity between image tags (in feature space T) and cluster centroids. As defined in Eq. 4.2, cosine similarity between two m-dimensional vectors $\vec{f}$ and $\vec{c}$ can be computed as:

$$cosine(\vec{f}, \vec{c}) = \frac{\vec{f} \cdot \vec{c}}{||\vec{f}|| \cdot ||\vec{c}||} \tag{6.5}$$

Experimental results show that Cosine similarity for tag/text based features performs significantly better than Euclidean distance. For comparison between different distance measures, we also evaluated the results on Manhattan distance for non-text based features. Manhattan distance between two vectors $\vec{f}$ and $\vec{c}$ is defined as follows:

$$manhattan(\vec{f}, \vec{c}) = \sum_{i=1}^{m} |\vec{f_i} - \vec{c_i}| \tag{6.6}$$

For low-level image features (specifically for *Edge Histogram Descriptor-EHD*) we evaluate *Histogram Intersection* (Smith, 1997) similarity. Histogram Intersection is computed as:

$$HI(\vec{f}, \vec{c}) = \frac{\sum_{i=1}^{m} min(\vec{f_i}, \vec{c_i})}{min(|f|, |c|)} \tag{6.7}$$

6.3.2 Identifying Representative Tags

After grouping the images into k clusters, we identify the representative tags for each cluster. Based on empirical analysis, we use the user frequency as the criterion for tag representativeness. Let's assume that Figure 6.4 represents the tags in a cluster. The tags which are used by more users are ranked higher in the cluster, e.g., the tags *Clocktower* and *Graz*. The tags which are used by fewer users are ranked lower, e.g., the tags *HDR* and *Photomatix*. Ranking by user frequency instead of resource frequency avoids the situation where many resources in a cluster are tagged by a single user. The ranked tags of a cluster are recommended for the new image. Section 6.4.4 discusses about the number of tags that should be recommended.

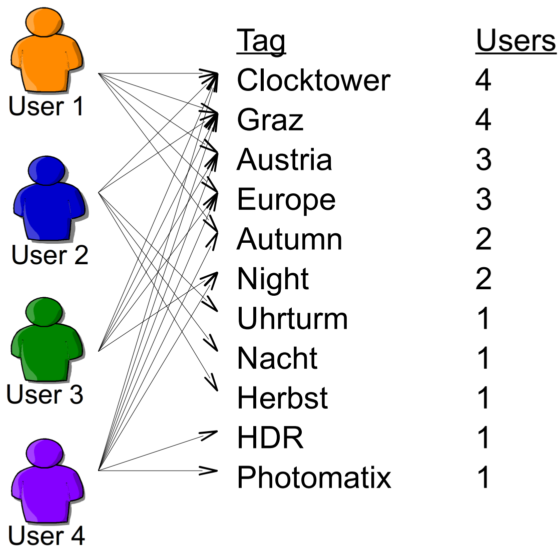

Figure 6.4: Example showing the ranking of the tags. The tags used by more users are ranked higher than other tags.

6.3.3 Classification and Tag Recommendation

The training for the proposed tag recommender system is completed after identifying the representative tags for each cluster. The tag recommendation system can be trained off-line using a large dataset. After the completion of the training phase, given a new image, it has to be mapped to the closest cluster, whose representative tags are assigned to the image. As in the case of clustering, we have many choices for training a classifier which learns its model based on the clusters identified in Section 6.3.1. One can train a one-class classifier (Manevitz and Yousef, 2002) considering images in a cluster as positive training examples. It is also possible to use a two class classifier (Baeza-Yates and Ribeiro-Neto, 1999), where images in one cluster are used as positive training examples, and images in other clusters are used as negative training examples. Instead of fine tuning the classification process, we use a method similar to Rocchio classification (Rocchio, 1971). A new image is mapped to a cluster whose centroid is at minimum distance from the image. Most representative tags associated with the mapped cluster are assigned to the new image.

For clusters based on geographical coordinates, we classify the new image to one of the clusters whose centroid is at minimum geographical distance from the new image. For low-level clusters, we classify the new image based on the distance between its low-level features and cluster centroids. For tag based clusters, as we do not have any tags for the new image, we classify the new image based on the distance between its geographical coordinates and the mean of geographical coordinates of the tag based clusters. The mismatch between feature spaces used for tag based clustering and the new image negatively affects the results of tags based clustering.

6.4 Experiments and Results

In this section the experiments and results are presented. The image dataset is briefly described in Section 6.4.1, the distinction between the training and the test data comes in Section 6.4.2, which is followed by the evaluation method in Section 6.4.3. Section 6.4.4 presents the comprehensive results achieved in our work.

6.4.1 Image Dataset

The CoPhIR dataset (Bolettieri et al., 2009) consists of images uploaded to Flickr by hundreds of thousands of different users, which makes the dataset very heterogeneous. One can find images of very different types like portraits, landscapes, people, architecture, screen shots etc. To perform an evaluation on different types of features (geographical coordinates, tags, low-level) on a reasonably large scale, we created a subset of the original CoPhIR dataset.

We selected the images taken in national capitals[1] of all the world countries. For this purpose, we considered all the images at a geographical distance of less than 0.1 Euclidean units from the center of the capital cities. We ignored the capital cities which had less than $1,000$ images; this resulted into a set of 58 cities. To keep the experiments scalable, we randomly selected $30,000$ images for cities which had more than $30,000$ images. There were only three such cities *Paris*, *London*, and *Washington DC*. In the end, we had images of 58 capital cities, ranging from $1,000$ to $30,000$ images with an average of $8,000$ images per city. Total number of images in our evaluation dataset was $413,848$. Images are trained and evaluated separately for each city.

Base Line: In order to compare the effectiveness of different image features, we created a random feature space for the images. We assign a random value between 0 and 1 to each image in the dataset as its random feature. We consider the random feature as the baseline for comparison. Same clustering methods are applied on the random features as on the other features. Random feature space is uni-dimensional and is represented as $D \in \Re$.

6.4.2 Training and Test data

It is important to carefully select the training and test datasets, because when a user uploads images to Flickr, he can perform batch operations on the set of images. For example, he can assign the same tags or geographical coordinates to all the images in a batch. It is also possible that the images have very similar low-level features. Now if we randomly split the images into the training and the test datasets, there is a chance that the images belonging to one user are used in both of the training and the test dataset. Such a random split may affect the final evaluation. Test images from one user might be mapped to a cluster which is trained on the images from the same user. It is very likely that the test image is annotated with the perfect tags, as the tags of both the test and the training images were provided by the same user.

To make the evaluation transparent, instead of randomly splitting the resources into training and test dataset, we split the users. For each city, we use

[1] `http://en.wikipedia.org/wiki/National_capitals`

resources of 75% users for training and resources of 25% users as test dataset. No image in the test dataset is annotated by a user who has also annotated images in the training dataset. After splitting the users into training and test datasets, we use $310,590$ images for training the system and $103,258$ images used as ground truth for evaluating the system.

Another aspect of fair evaluation is the quality of the tags. There are some tags which are very common in both test and training datasets. These tags mostly represent city or country names, which can be suggested by looking into a geographical database. Some common tags might not be very specific, e. g., the tags *geo-tagged*, *2007*, *travel* etc. Very common tags also affect the evaluation results, as they are abundant in both test and training datasets, and are almost always recommended for every test image. This results in higher precision and recall values.

To make the evaluation more transparent, we do not consider the ten most frequent tags for each city and we also ignore the frequent tags *geo-tagged* and *geotag*, because all the images in our dataset are geo-tagged and most of the images have these two tags. For each city, we also remove the very rare tags which might be incorrectly spelled tags or tags specific to a particular user. For this reason, for each city, we ignore those tags which are used by less than three users.

6.4.3 Evaluation

We consider the tags associated with the $103,258$ test images as ground truth. The images in the ground truth are tagged by different users and as there is no restriction on the selection of tags for a resource, therefore the tags in the ground truth are very noisy. The noise in the data leads to inferior results, but the overall results show the comparative analysis of different feature spaces. We evaluate the methods using standard evaluation methods used in information retrieval: Precision P, Recall R, and F-Measure F. The evaluation measures are defined as follows:

$$P = \frac{Number\ of\ correctly\ recommended\ tags}{Number\ of\ recommended\ tags} \tag{6.8}$$

$$R = \frac{\textit{Number of correctly recommended tags}}{\textit{Number of expected tags}} \tag{6.9}$$

$$F = \frac{2 \times P \times R}{P + R} \tag{6.10}$$

If few tags are recommended correctly for most of the cases, micro evaluation measures may have bias towards those few tags. To reduce this bias, we also compute the macro precision P_m, macro recall R_m, and macro F-Measure F_m over tags. The mean of averages in macro evaluations reduce the bias towards certain tags correctly recommended. Macro evaluations are computed as follows:

$$P_m = \frac{\displaystyle\sum_{t \in \textit{Tags recommended}} \frac{\textit{\# of times t correctly recommended}}{\textit{\# of times t recommended}}}{\textit{\# of tags recommended}} \tag{6.11}$$

$$R_m = \frac{\displaystyle\sum_{t \in \textit{Tags Expected}} \frac{\textit{\# of times t correctly recommended}}{\textit{\# of times t expected}}}{\textit{\# of tags expected}} \tag{6.12}$$

$$F_m = \frac{2 \times P_m \times R_m}{P_m + R_m} \tag{6.13}$$

6.4.4 Results

The results presented in this section give a comparative view of tag recommendation based on different types of features. The automated evaluation on one hand provides the possibility to do evaluation on a large scale, but on the other hand the ground truth (test data) might contain invalid tags, which gives inferior results. We make the evaluation transparent and more meaningful by filtering certain types of tags (see Section 6.4.2).

By removing very common tags, there is a certain decrease in the performance of recommender system, but it is an important step towards a fair evaluation. We have also evaluated the results without filtering the dataset, and in that case even random feature space gives an F-Measure value of 0.42. This is because very common tags are being recommended for the test images and there is always a major overlap between common tags of the training and the test data.

The precision, recall, and F-Measure values presented in this section might appear to be low for the reader, but one should consider that the dataset was filtered to make the evaluation more transparent.

The Figure 6.5, Figure 6.6, and Figure 6.7 depict the so called *micro average* evaluation and were generated in accordance to the evaluation criteria (see Equation 6.8), (see Equation 6.9), and (see Equation 6.10) respectively. As one can see, in all three cases the results are significantly better when using geographical coordinates for image description.

The performance of the tag recommendation using low-level features and textual tags differs only slightly from the results based on random clustering. For exactly one tag being recommended, the precision amounts to: 0.1385 for geographical coordinates, 0.0502 for low-level features, 0.0512 for textual tags, and 0.0338 for random clustering. Besides the fact that low-level image features perform worse than the geographical coordinates, one possible reason could be the sparsity of tagging data in the training dataset. The training dataset consisted of the images from Flickr where users might not added a sufficient amount of tags to their resources. When observing the results of different clustering methods, we noticed that though the low-level image features resulted into meaningful clusters, the images in the clusters were not properly tagged. For example, a cluster based on low-level image features depicting a football ground in all of its images did not contain the tag *ground*. Similarly a cluster of images having close-ups of crowd in the stadium did not contain the tag *crowd*. Such kind of sparsity in the training data could have affected the results of tag recommendation using low-level image features.

In addition to the problem of data sparsity, the worse results of tag recommendation based on the tag feature could be a result of the feature mismatch problem (see Section 6.3.3). The proposed framework recommends the tags for new images, and as the new images are not already tagged, their feature space cannot be used to map them on existing clusters. In our experiments, we have used the geographical coordinates of the new image to map it on the existing clusters (based on tag feature). This mismatch of different feature spaces of the new image and the existing clusters (in case of tag clustering) could have affected the results of tag recommendation for tag feature.

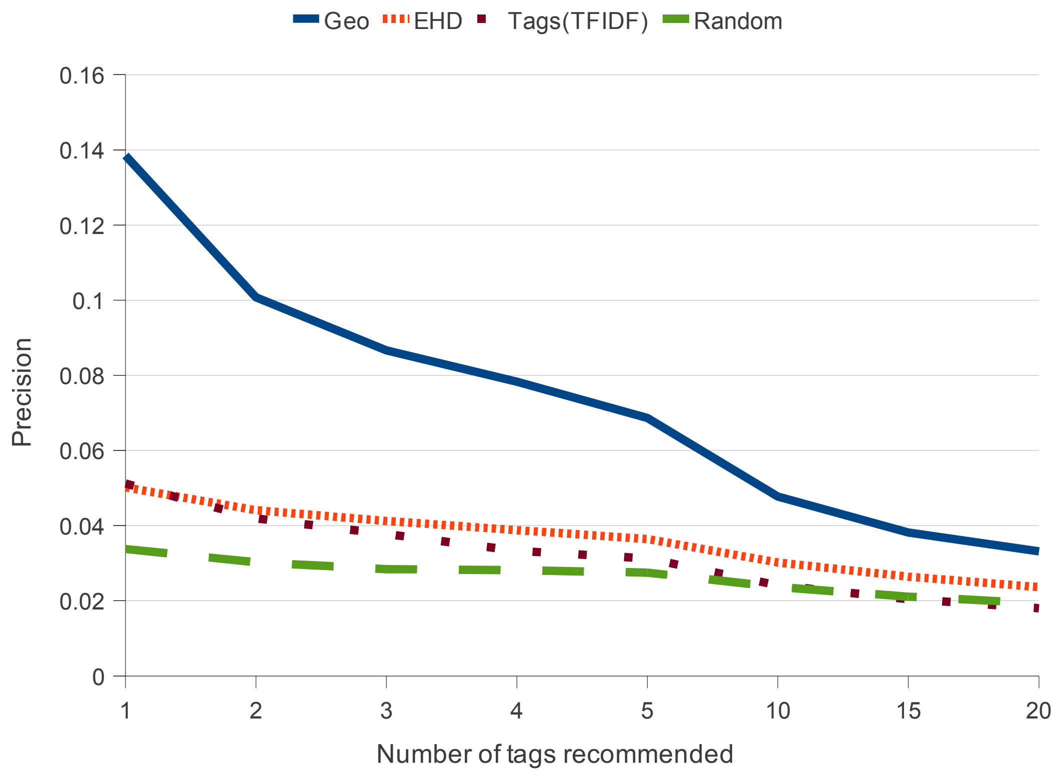

Figure 6.5: Micro precision for geographical, low-level (edge histogram descriptor), tag, and random image features. Precision for geographical coordinates is higher than all other image representations.

For more than 10 tag recommendations, random feature performs slightly better than the tag features, this is due to the micro evaluation measures (see Equations 6.8–6.10). In random feature, tags are equally distributed among different clusters, and because some of the common tags like *night* or *sky* are correct for many images, therefore overall performance of the random feature remains comparable to the tag based features or low-level image features. However, for macro evaluations (see Equations 6.11–6.13), random feature performs inferior than other image representations. This is because of the reason that macro-evaluation reduces the bias that only few common tags recommended for most of the images.

The Figures 6.8, 6.9, and 6.10 present the so called *macro average over tags* evaluation and were generated in accordance to the evaluation criteria (see Equa-

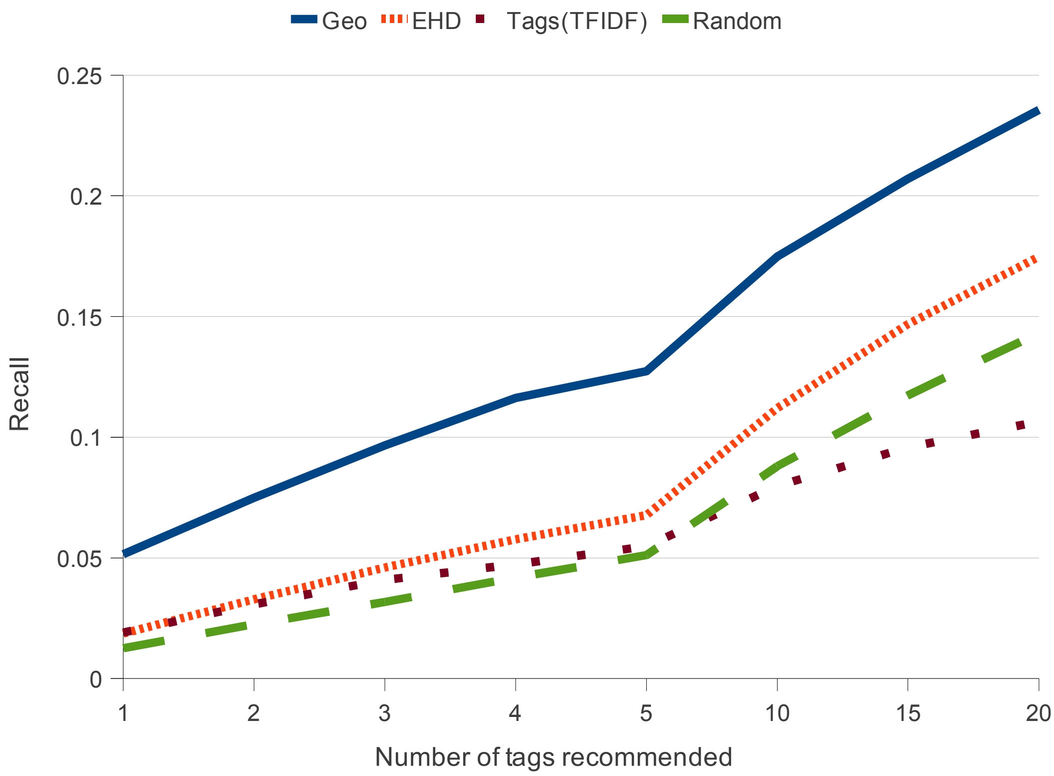

Figure 6.6: Micro recall for geographical, low-level (edge histogram descriptor), tag, and random image features. Recall for geographical coordinates is higher than all other image representations. For more than 5 tag recommendations, random feature performs slightly better than the tag feature due to the micro recall evaluation measure. When using macro-recall, tag feature performs better than the random feature.

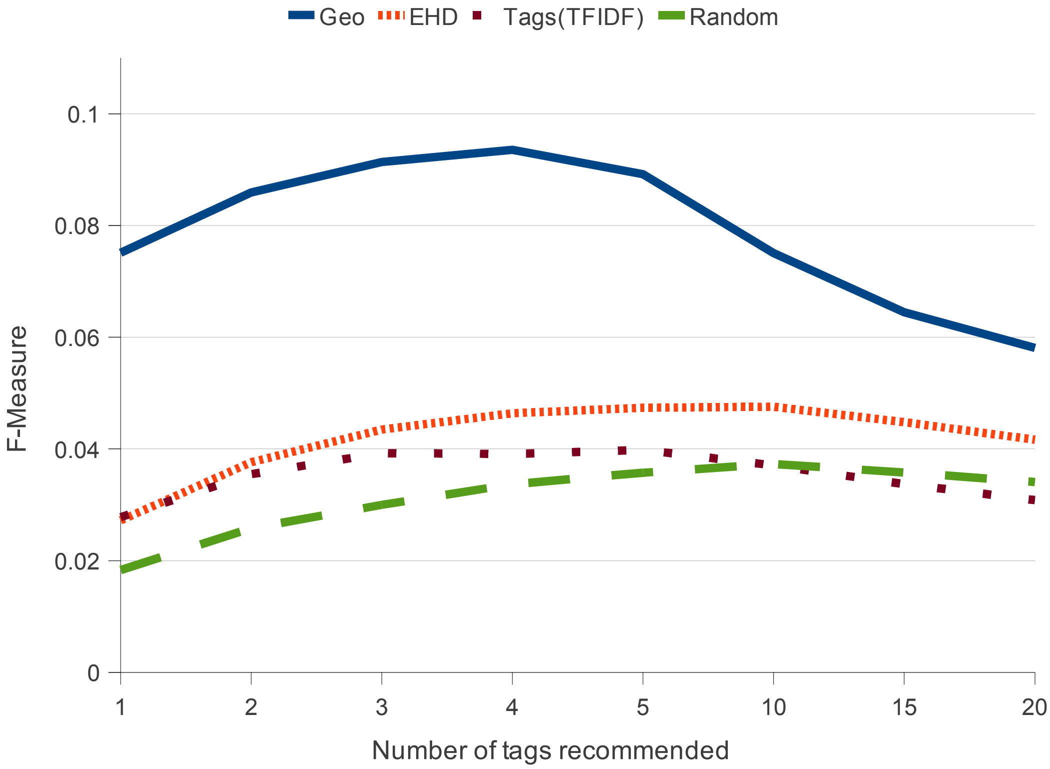

Figure 6.7: Micro F-measure for geographical, low-level (edge histogram descriptor), tag, and random image features. Geographical feature performs better than all the other image representations.

tion 6.11), (see Equation 6.12), and (see Equation 6.13) respectively.

Similar to the micro average evaluation, the results here are significantly better for geographical coordinates, while the performance in case of textual tags, low-level features, and random clustering is almost the same. For exactly one tag being recommended, the macro precision for geographical coordinates amounts to 0.1584, for low-level features - 0.0521, for textual tags - 0.05, and the baseline is 0.0312.

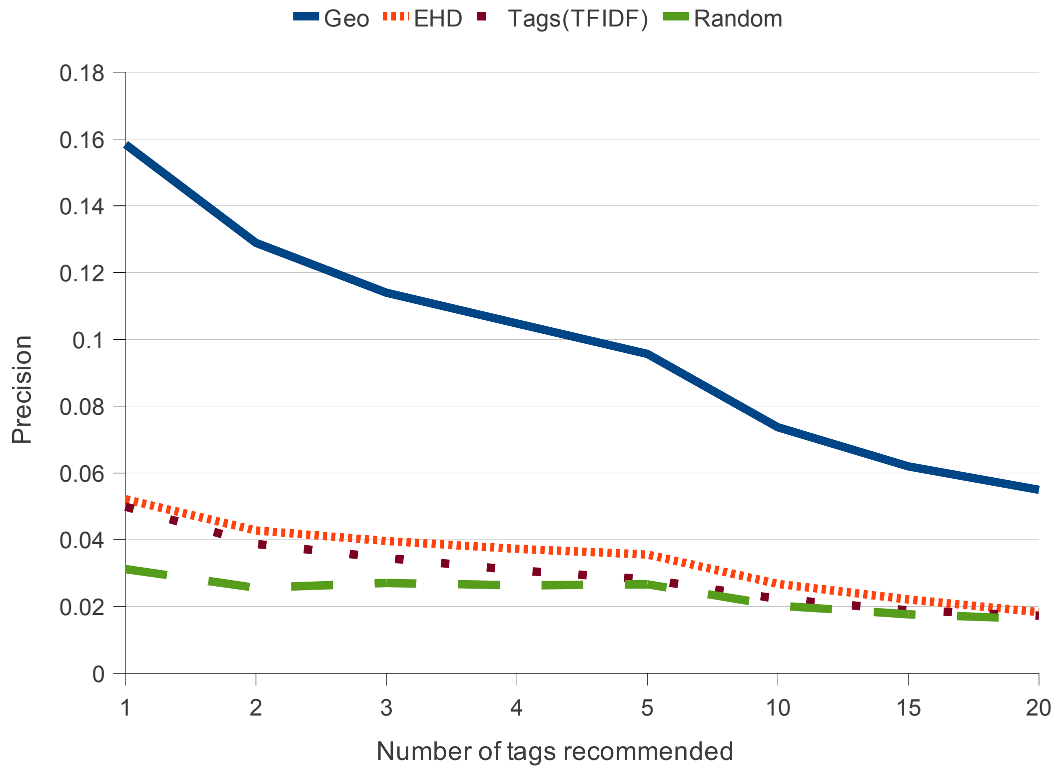

Figure 6.8: Macro precision for geographical, low-level (edge histogram descriptor), tag, and random image features. Precision for geographical coordinates is higher than all other image representations.

The results also suggest the threshold for the number of tags that should be recommended for a new resource. For a tag recommender whose focus is more on the correctness of the tags, only the top few tags should be recommended, as obvious from Figure 6.8, when more tags are recommended, the precision of the recommendation gets lower. In the case of a tag recommender system which does not require the tag recommendations with higher precision, ten tags can be

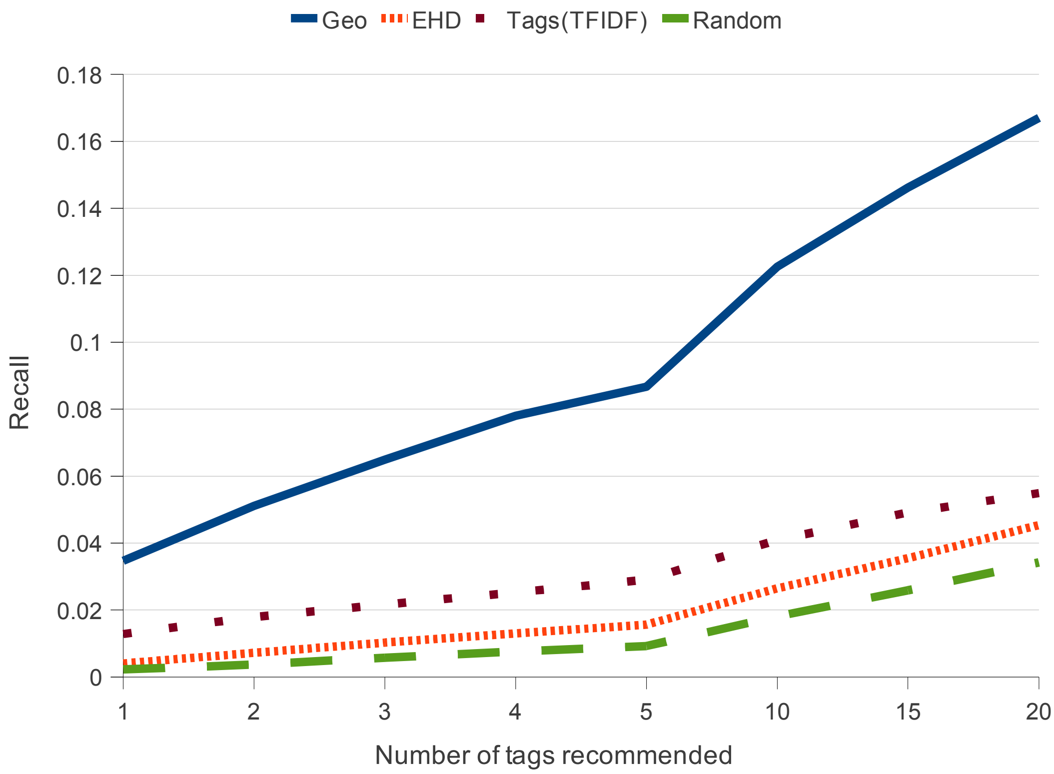

Figure 6.9: Macro recall for geographical, low-level (edge histogram descriptor), tag, and random image features. Recall for geographical coordinates is higher than all other image representations. Random feature performs worse than all other features.

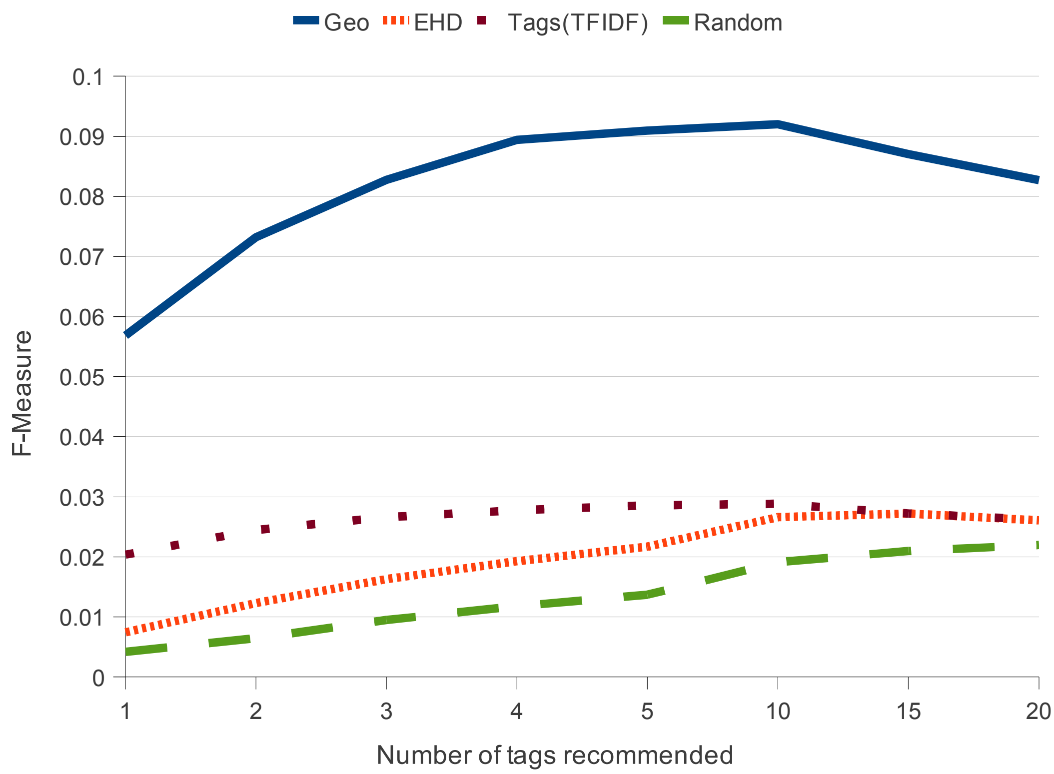

Figure 6.10: Macro F-Measure for geographical, low-level (edge histogram descriptor), tag, and random image features. F-Measure for geographical coordinates is higher than all other image representations. Random feature performs worse than all other features.

recommended. As shown in Figure 6.10, the best results (in terms of F-Measure) are achieved when ten tags are recommended, this is the best compromise between the precision and the recall for the tag recommendations.

Although we avoided fine tuning of experiments for most of the cases, we performed experiments to compare effectiveness of different alternative methods related to the features of the resources. In the Figures from 6.11 to 6.15 results of further evaluations are presented. The Figure 6.11 explains why using the simple Euclidean distance has appeared to be sufficient in our approach. The results remain almost the same when using Manhattan distance.

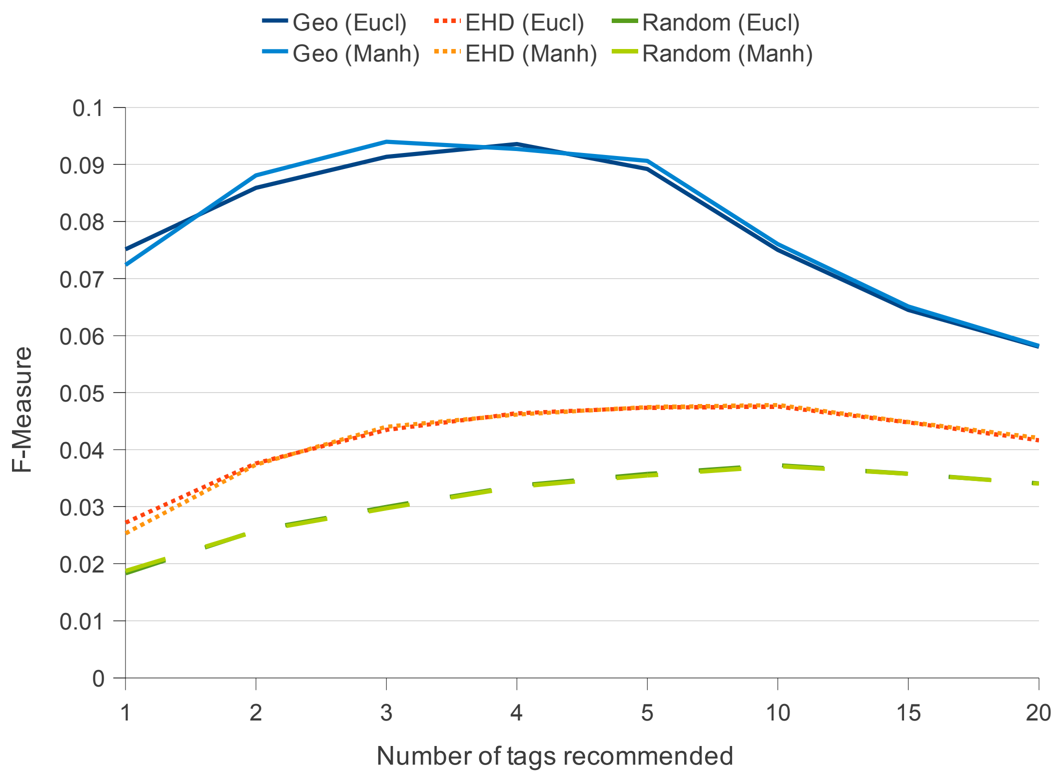

Figure 6.11: Micro F-Measure comparison of Manhattan (Manh) and Euclidean (Eucl) distances for non-text based features. Dark lines show the results obtained using Euclidean distance and gray lines show results obtained using Manhattan distance. Performance for both distances is almost same for all the image features.

Using different low-level features did not significantly affect the performance of the tag recommender system. As one can see in the Figure 6.12, that the Edge Histogram Descriptor (EHD) performs almost the same as the Color Layout (CL) feature in terms of micro F-Measure.

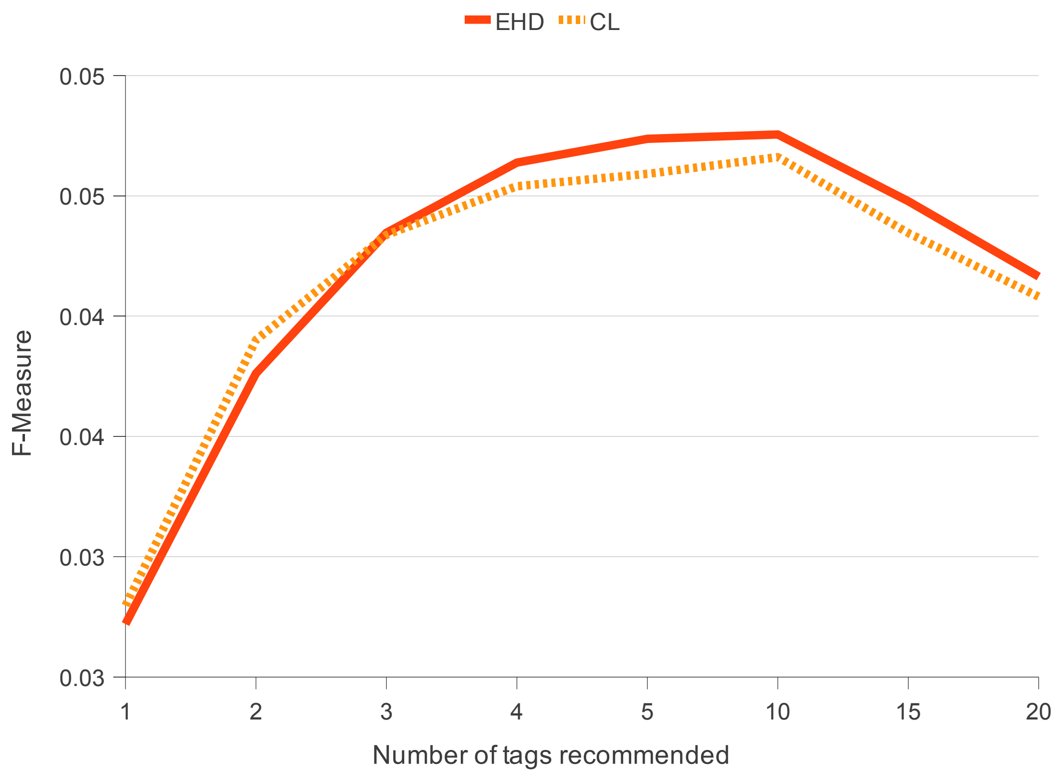

Figure 6.12: Micro F-Measure comparing results of two different low-level features Edge Histogram Descriptor (EHD) and Color Layout (CL). Both of the low-level image features performs almost the same.

When measuring similarity between resources based on low-level features, a variety of options are available, we evaluated the similarity between images using histogram intersection (see Equation 6.7) in addition to simple Euclidean distance. Figure 6.13 shows the micro F-Measure applied the two different distance measures.

Euclidean distance for tag recommendation performs significantly better than the histogram intersection. The performance of tag recommendation using Euclidean distance starts to decline when more than 10 tags are recommended, however, in case of histogram intersection the performance continues to improve for 15 and 20 tag recommendations.

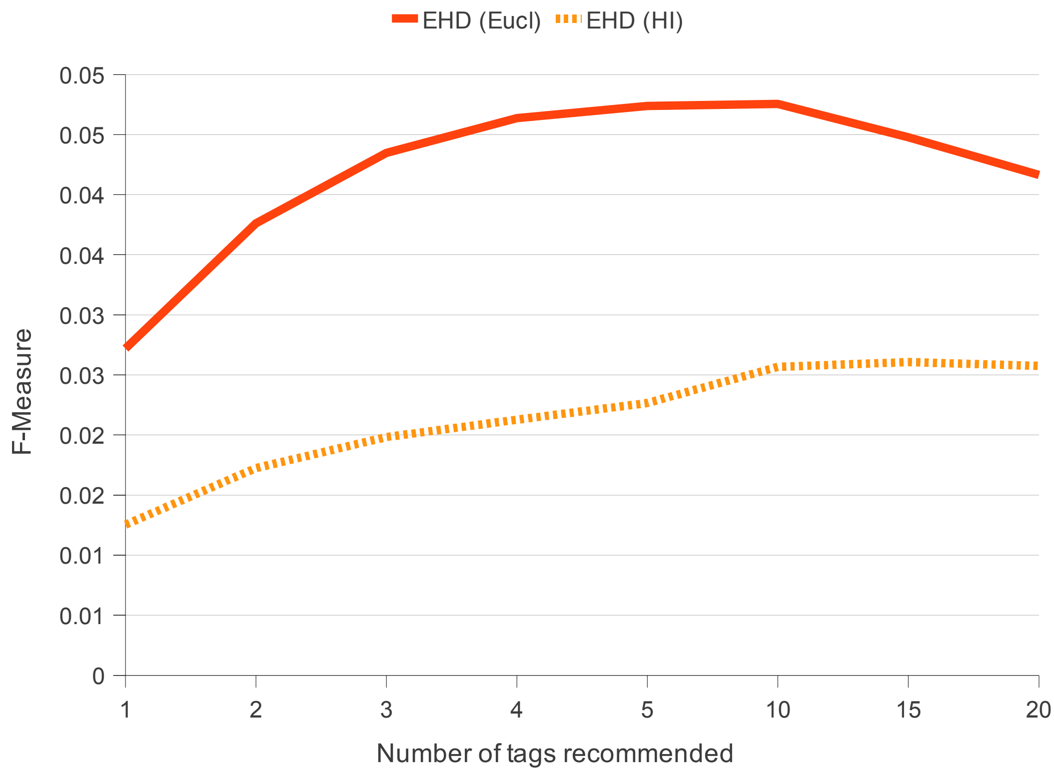

Figure 6.13: Micro F-Measure comparison of Euclidean (Eucl) and Histogram Intersection (HI) metrics for MPEG-7 Edge Histogram Descriptor. Recommendations based on Euclidean distance perform significantly better than the histogram intersection measure.

Cosine similarity was used for tag based feature space and Figure 6.14 shows a clear advantage of the Cosine distance over the Euclidean distance for tag based features.

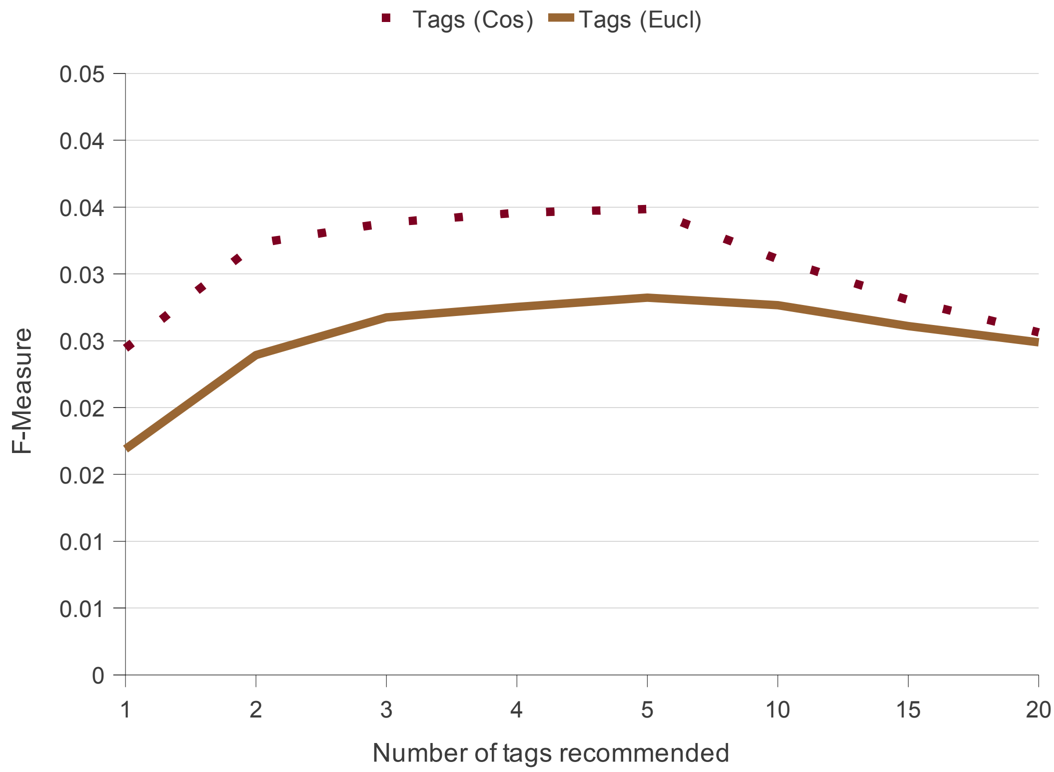

Figure 6.14: Micro F-Measure comparison of Cosine (Cos) and Euclidean (Eucl) distances for tag/text based features. Cosine distance performs significantly better than the Euclidean distance when using tag feature for recommendation.

To investigate the effect of different weight schemes for the tag features, we evaluated the results using two different normalization strategies, which are, term frequency (TF), and term frequency-inverse document frequency (TF-IDF). Although the overall effect of normalization is not significant, but applying TF-IDF normalization on simple tag feature slightly improves the tag recommendation process as shown in Figure 6.15.

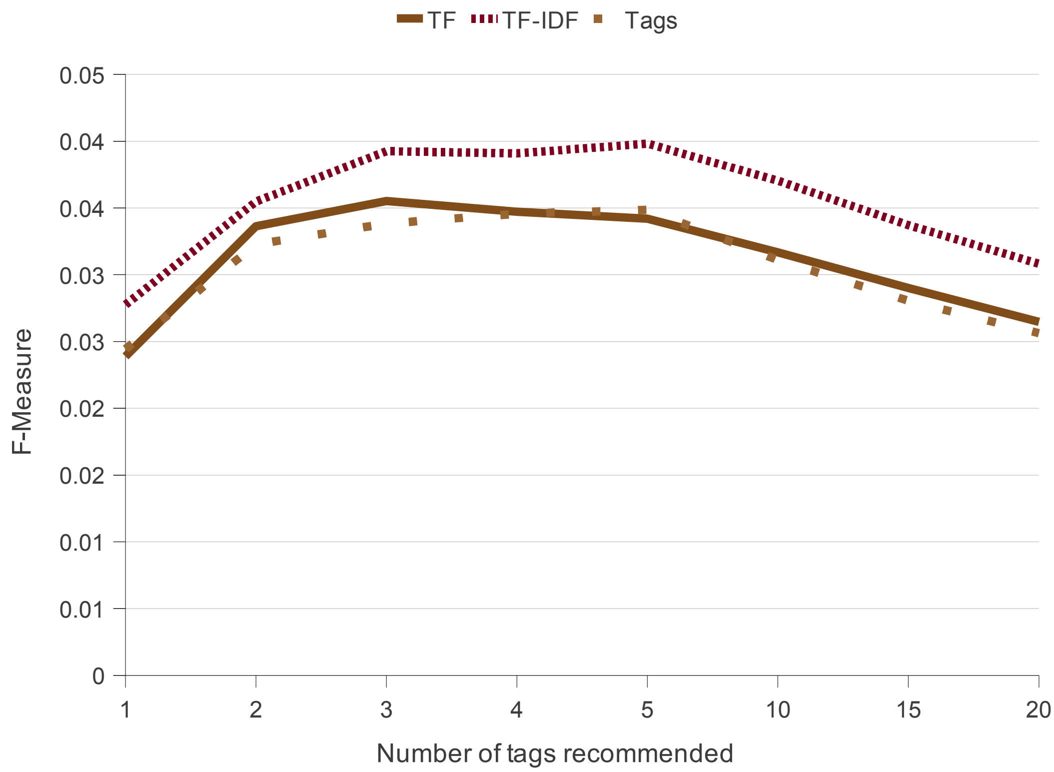

Figure 6.15: Micro F-Measure comparison of Tags features with different normalization strategies (Tag Frequency-TF, Tag Frequency Inverse Document Frequency-TF-IDF). TF-IDF normalized tag features perform better than no normalization and TF based normalization.

6.5 Conclusions

The methods discussed in this chapter exploit three kinds of image description techniques, namely geographical coordinates, tags, and low-level features, to recommend tags for the new resources uploaded to a social tagging system. In order to compare the benefits each of these description types brings to a tag recommender system on its own, we investigated them independently of each other.

First, the existing data collection was clustered separately for the geographical coordinates, tags, and low-level features. Additionally, random clustering was performed in order to provide a baseline for experimental results. Once a new image was uploaded to the system, it was assigned to one of the clusters using either its geographical or low-level representation. Finally, the most representative tags for the resulting cluster were recommended to the user for annotation of the new image. Section 6.4 evaluated tag recommender systems of the architecture proposed.

Large-scale experiments performed for more than $400,000$ images compared the different image representation techniques in terms of precision and recall in tag recommendation. A tag recommender system of the architecture proposed benefits the most from geographical information associated with the images.

One of the important contributions of this chapter is the evaluation on a large-scale image database. For our experiments we used the CoPhIR dataset (Bolettieri et al., 2009) including images uploaded to Flickr by hundreds of thousands of different users. The total number of images in our evaluation dataset was $413,848$.

Another significant contribution was considering the tag recommendation problem separately for images described with tags, geographical coordinates, and low-level features, as well as comparing the results to a baseline achieved based on random clustering.

The results presented in Section 6.4 showed that geographical coordinates are the most helpful image descriptors for tag recommendation, while textual tags and low-level features provide only a slightly better performance than the random baseline. It might sound disappointing, but low-level features and textual tags do not seem to be suitable for tag recommender systems connected to large-scale heterogeneous image databases. However, textual tags and low-level features might be very helpful for small image databases with a clearly defined domain (e.g., medical domain (Müller et al., 2003)) and also in a situation where geographical coordinates are not available.

In the future, we will keep investigating the tag recommendation problem for large-scale heterogeneous image archives. We will further develop our framework to allow comprehensive experimental studies. We will also investigate the problem for some more domain dependent data collections.

Chapter 7

Exploiting Semantics for Classification

Folksonomies are expanding tremendously. Everyday more resources are added to them. As the size of folksonomies increases, it becomes difficult to explore them. The tags help in exploring folksonomies to some extent, but sometimes tags might not be sufficient enough to identify the semantics of the resources available on folksonomies. For example, if a person searches for the landmarks of a city, he might get results related to the people or some objects. However, if we know the semantics of the tags and the resources, we can provide the user with the particular type of resources he is searching for, e.g., those related to the landmarks of a city. Hence we need a classifier which can identify the semantics of the tags and the resources.

In this chapter we propose two methods for identifying the semantics of the tags and the resources. The first method is called T-ORG (Tag ORGanizer). It exploits web resources like web search engines to identify the semantics of a particular tag. It classifies the tags into predefined categories. In contrast to supervised classification, T-ORG uses its own algorithm called T-KNOW (Tag classification through KNOwledge on the Web), which does not require any training data. The second method called TG-SVM (Tag Group Support Vector Machine) exploits the information available in form of social groups available in some folksonomies. Many people upload resources to these groups, created around particular themes.

TG-SVM exploits these thematic groups to identify resources related to a particular class, for example, landmarks of a city. We evaluate the proposed methods based on human evaluation. The results are encouraging and show that it is possible to identify the semantics of tags and resources in folksonomies.

7.1 Exploiting Web Resources for Classification

Users search and browse resources using the tags attached to the resources. Folksonomies also provide "Tag Clouds" (see Section 3.1.3, and Figure 3.2) to browse resources. In a "Tag Cloud", frequently used tags are displayed in a bigger font size and less frequently used tags are used in a smaller font. Sometimes it might become difficult to browse resources related to a particular category. Just consider the scenario in which a user wants to explore vehicle images. Considering the current searching and browsing facilities provided by folksonomies, it seems difficult to browse only a particular type of resources like landmarks, people, or vehicles etc. The problem of exploring resources of a particular type can be solved by additional classification of tags and resources.

Classifying resources into predefined categories provides a mechanism to explore resources related to a particular category. One way of classification is to build a training dataset which is then used to train a (supervised) classifier. Building a training dataset is a time consuming task. The manual classification of resources might also be not feasible, because of the tremendous amount of data present and being added to folksonomies. Therefore a system is required that classifies resources into categories without any supervision.

To handle the problem of classification in an unsupervised manner, we have explored means to automatically organize tags into classes. For this purpose, we have developed a system called T-ORG (Tag-ORGanizer), which classifies the tags and resources of a folksonomy into predefined categories. It helps in browsing resources related to a particular class. The classification of resources is based on the classification of tags attached to these resources. If a resource has two tags related to two different categories, the resource is classified into both of these categories. For example, if a resource is annotated with the tags *Paris* and *Peugeot* and these tags are classified as *Location* and *Vehicle* respectively, then the resource is placed in *Location* and *Vehicle* categories.

We developed a classification algorithm called T-KNOW (see Section 7.1.3) for the T-ORG system. To avoid the efforts that are required for training a classifier, the T-KNOW algorithm exploits a web search engine, few linguistic patters and the context of the tags. It does not require any training data for learning the classification model. We evaluate the proposed algorithm for four different types of tag contexts (see Section 7.1.2).

The tag classification can help a user to browse a folksonomy in a more organized way. Instead of representing different tags in a tag cloud based on their frequency (see Figure 3.2), a tag cloud can display the tag classes. When a user clicks on a tag class, resources belonging to that tag class are displayed. In such way, a user explores different types of tags and resources available on a folksonomy, that is not be possible with a simple tag cloud.

7.1.1 Tag Organization using T-ORG

The purpose of T-ORG is to organize resources by classifying their tags into categories. This process is done by selecting concepts from single or multiple ontologies related to the required categories and then pruning and refining these ontologies. These concepts are considered as categories into which the tags are classified. Figure 7.1 shows the overall process of T-ORG while each step is described as follows:

1. Select Ontologies

The user of T-ORG has to decide about the categories into which the resources are classified. The user selects ontologies relevant to the required categories. Concepts from these ontologies are used as categories. For example, to browse through the images of vehicles on the Flickr website, one would select vehicle ontology. Currently this step is done manually in T-ORG.

2. Prune and Refine Ontologies

After selecting ontologies, they must be pruned and refined for the desired categories. Only those concepts from these ontologies are considered which have some relation to the required categories. Unwanted concepts are pruned. Redundant and conflicting concepts are refined. Missing concepts are also added into the

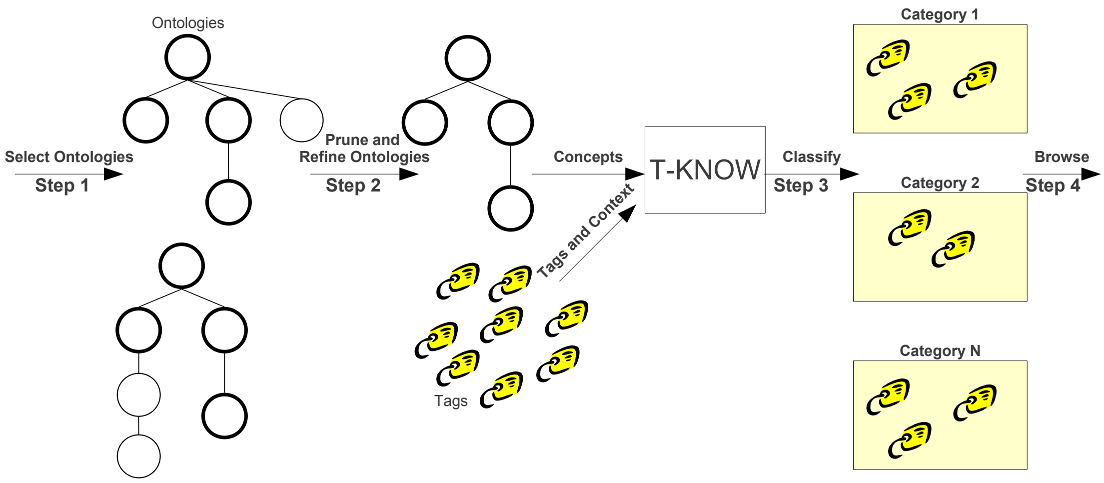

Figure 7.1: Process of T-ORG.

given ontology. For example, to include the images of a *draisine*, one might have to add this concept into a given vehicle ontology. Once the ontology is pruned and refined, its concepts are used as categories. Currently this step is also done manually in T-ORG.

3. Classify Tags using T-KNOW

Classifying the tags is a major step in the process of T-ORG. Once the ontology is selected, pruned, and refined, and categories are extracted from this ontology, then these categories and the context of the tags (see Section 7.1.2) are used for classification. Once all tags are classified into categories, each category is subsumed by its parent category, for example, every tag classified as Train, Bulldozer or Bus is finally classified as Vehicle. Section 7.1.3 describes the detailed process of classifying tags using T-KNOW.

4. Browse Resources

After classifying each tag, resources may be browsed according to the categories assigned to their tags. The browser may use information of resources to display them in categories, so that the user browses any particular type of resources classified into these categories.

In the following section, we define different contexts of the tags which are used in the T-KNOW algorithm for classifying the tags.

7.1.2 Exploiting the Context of the Tags

The context of a tag facilitates in understanding the meaning of a tag. We have already discussed the *resource* and the *user* context of the tags in Chapter 4 and their application for improving search in Chapter 5. In this section we define two more contexts, the *tag* context and the *social group* context. Social groups in Flickr[1] contain resources and tags related to particular themes. For example, a group related to vehicles would contain images and tags of vehicles[2] and a group about landmarks would contain images and tags related to the landmarks[3].

To understand the different contexts of the tags, consider the three images in Figure 7.2. The left most image is of "Eiffel Tower". The middle image is "Notre Dame". The right most is the image of a "Cow". The images of "Cow" and "Eiffel Tower" are associated with "Group 1" and "Notre Dame" with "Group 2". The image "Eiffel Tower" has been annotated by the "User A" and the other two images by "User B".

Table 7.1 shows the elements of Figure 7.2 in a tabular form. As an example of a context, the *resource context* for the tag *Paris* in the image *Notre Dame* consists of the tags "*Notre-Dame*, *France*, *Night* and *Lights*". Similarly different contexts of the different tags can be visualized using the Figure 7.2 and Table 7.1. These images in Figure 7.2 and Table 7.1 are used in the following paragraphs to explain the different contexts.

Table 7.1: Details of the images in Figure 7.2.

Image	**Tags**	**User**	**Group**
Eiffel Tower	Eiffel Tower, Paris, France, Miniatures, Eiffel, Eyeful, Big	A	1
Notre Dame	Notre-Dame, France, Night, Lights, Paris	B	2
Cow	Savoie, France, 2001, Field, Cow	B	1

[1] `http://www.flickr.com/groups/`, last accessed in October 2010

[2] `http://www.flickr.com/groups/cardirectory/`, last accessed in October 2010

[3] `http://www.flickr.com/groups/historicalbuildingslandmarks/`, last accessed in October 2010

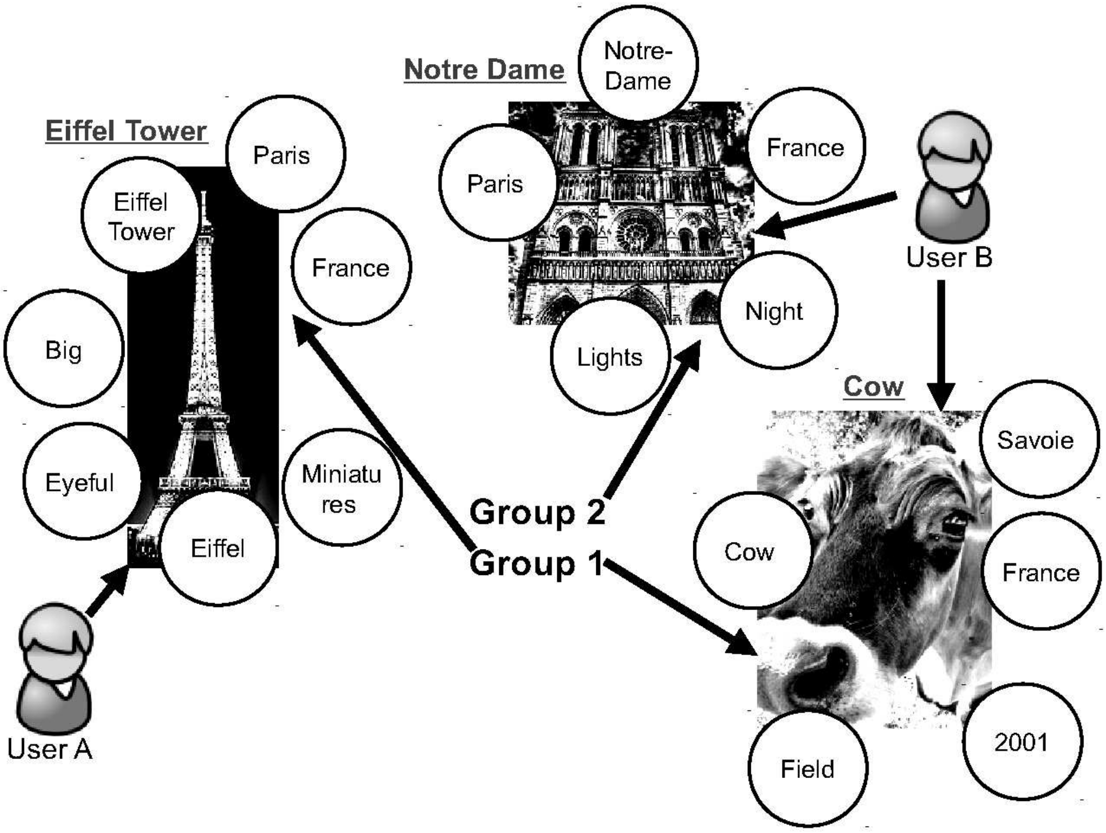

Figure 7.2: Sample images with tags.

7.1.2.1 Resource Context (R)

The resource context of a tag t associated with a resource r consists of the tags which appear in the resource r. To formally define the different contexts of the tags, we use the same formulation of folksonomies used in Chapter 3, Section 3.2. A folksonomy F is defined as a tuple $F := (U, T, R, Y)$, where U, T, and R are finite sets representing users, tags, and resources respectively. Y represents the tag assignments by the users U, using the tags T for the resources R. The tag assignments Y are denoted as: $Y \subseteq U \times T \times R$. In addition to these sets, we also use the set of groups G which exists in some folksonomies (like Flickr). Now the Resource Context of a tag t associated with the resource r is defined as follows:

$$C_R(t, r) = \{t' \in T \backslash \{t\} \mid (u, t', r) \in Y\} \tag{7.1}$$

We are also interested in the frequency of t_i in resource r_j to create a bag of words using this context. In the case of Flickr it is at most 1, because one tag can occur only at most once in a resource. $W_R(t, r)$ represents the number of times tag t appears with resource r.

$$W_R(t, r) =\mid \{(u, t, r) \in Y\} \mid \tag{7.2}$$

We get the Resource Context of a tag t of resource r using $C_R(t, r)$ and for each tag t in the Resource Context of tag t, we get its number of occurrences in resource r using $W_R(t, r)$. We also define a bag-of-words resource context representation of a tag t appearing in resource r, i.e. by

$$B_R(t, r) = \{(t', W_R(t', r)) \mid t' \in C_R(t, r)\} \tag{7.3}$$

Note that B_T, B_{SU}, and B_{SG} can be defined in the similar manner for the *tag*, the *user*, and the *social group* contexts respectively, as we will see in the following sections.

Consider that we want to classify the tag *Paris* of the image Eiffel-Tower in Figure 7.2, only the tags of the image Eiffel-Tower are selected as the context, i.e. C_R*("Paris", Eiffel-Tower)* = {*"Eiffel Tower", "France", "Miniatures", "Eiffel", "Eyeful", "Big"*}. The bag-of-words representation of the tag *Paris* of Eiffel-

Tower will be B_R(*"Paris", Eiffel-Tower*) = {(*"Eiffel Tower", 1*), (*"France", 1*), (*"Miniatures", 1*), (*"Eiffel", 1*), (*"Eyeful", 1*), (*"Big", 1*)}.

7.1.2.2 Tag Context (T)

In the case of Tag Context, we select all tags associated to the resources having the tag t, except the tag t itself. Tag Context is defined as

$$C_T(t) = \{t' \in T \backslash \{t\} \mid (u, t, r) \in Y \wedge (u', t', r) \in Y\} \tag{7.4}$$

For creating a bag of words representation (like Equation 7.3) using this context, we define $W_T(t, t')$ that represents the number of times the tag t appears with the tag t'.

$$W_T(t, t') =\mid \{t, t' \in T \mid (u, t, r) \in Y \wedge (u, t', r) \in Y\} \mid \tag{7.5}$$

We get the Tag Context of a tag t using $C_G(t)$ and for each tag t' in the Tag Context of tag t, we get its number of occurrences with tag t using $W_T(t, t')$. We define a bag-of-words tag context representation of a tag t appearing with the tag t'', i.e. by

$$B_T(t, t'') = \{(t', W_T(t', t'')) \mid t' \in C_T(t, t'')\} \tag{7.6}$$

Consider that we want to classify the tag *Paris* of the image Eiffel-Tower. All tags of images having the tag *Paris* are selected as the Tag Context except the tag *Paris* itself. In the example of Figure 7.2, Eiffel-Tower and Notre-Dame have the tag *Paris*, so all the tags of the images Eiffel-Tower and Notre-Dame are added to the context of the tag *Paris* except the tag *Paris* itself, and number of occurrences of each of these tags with tag t is calculated using W_T. Thus, B_T(*"Paris", Eiffel-Tower*) = {(*"Eiffel Tower", 1*), (*"France", 2*), (*"Miniatures", 1*), (*"Eiffel", 1*), (*"Eyeful", 1*), (*"Big", 1*), (*"Notre-Dame", 1*), (*"Night", 1*), (*"Lights", 1*)} is the bag-of-word representation constructed using Tag Context of the tag *Paris*.

7.1.2.3 User Context (U)

In the case of User Context of a tag t, we select all the tags used by a user u, except the tag t itself. The User Context of tag t of user u is defined as follows:

$$C_{SU}(t, u) = \{t' \in T \backslash \{t\} \mid (u, t', r) \in Y\} \tag{7.7}$$

For creating a bag-of-words representation (like Equation 7.3) using this context, we define $W_{SU}(t, u)$ that represents the number of times tag t is used by the user u.

$$W_{SU}(t, u) = \mid \{(u, t, r) \in Y\} \mid \tag{7.8}$$

We define a bag-of-words user context representation of a tag t used by the user u, i.e. by

$$B_U(t, u) = \{(t', W_U(t', u)) \mid t' \in C_U(t, r)\} \tag{7.9}$$

Consider that we want to classify the tag *Paris* of the image Notre-Dame that belongs to user B. All tags of images that belong to the user B are selected as the context except the tag *Paris* itself. In the example of Figure 7.2, the images Notre-Dame and Cow belong to the user B, so all the tags of the images Notre-Dame and Cow are added to the context of the tag *Paris* except the tag *Paris*. Thus, $B_{SU}\{$(“Paris”, Notre-Dame) = (“Notre Dame”, 1), (“France”, 2), (“Night”, 1), (“Lights”, 1), (“Savoie”, 1), (“2001”, 1), (“Field”, 1), (“Cow”, 1)$\}$ is the bag-of-word representation constructed using the user context.

7.1.2.4 Social Group Context (SG)

In the case of Social Group Context of tag t that is present in groups g, we select all the tags of all resources present in the same group g, except the tag t itself. The Social Group Context is defined as follows:

$$C_{SG}(t, g) = \{t' \in T \backslash \{t\} \mid (u, t', r) \in Y \land g \in \text{ Group}(u, r)\} \tag{7.10}$$

Where $Group(u, r)$ is a function which returns the groups that contain the user u and resource r.

For creating a bag-of-words representation using this context (like in Equation 7.3), we define $W_{SG}(t, g)$ that represents the number of times tag t appears in the group g.

$$W_{SG}(t, g) = \mid \{(u, t, r) \in Y \mid g \in \text{ Group(u,r)}\} \mid \tag{7.11}$$

We define a bag-of-words social group context representation of a tag t appearing in a group g, i.e. by

$$B_G(t, g) = \{(t', W_{SG}(t', g)) \mid t' \in C_{SG}(t, g)\} \tag{7.12}$$

Consider that we want to classify the tag *Paris* of the image Eiffel-Tower that belongs to group 1. All tags of images present in group 1 are selected as the context except the tag *Paris* itself. In the example of Figure 7.2, the images Eiffel-Tower and Cow are present in group 1, so all the tags of the images Eiffel Tower and Cow are added to the context of the tag *Paris* except the tag *Paris* itself. B_{SG}*("Paris",group-1) =* $\{$*("Eiffel Tower", 1), ("France", 2), ("Miniatures", 1), ("Eiffel", 1), ("Eyeful", 1), ("Big", 1), ("Savoie", 1), ("2001", 1), ("Field", 1), ("Cow", 1)*$\}$ is the bag-of-word representation constructed using social group context.

7.1.3 Tag classification using T-KNOW

The algorithm of T-KNOW uses lexico-syntactic patterns and web search engine for finding the appropriate categories of the tags. Given a list of tags and the categories, the algorithm of T-KNOW classifies these tags into categories. It builds queries by combining the linguistic patterns (Hearst, 1992; Cimiano et al., 2005) and the category names. It then searches these queries on a web search engine. The process of classifying tags using T-KNOW is shown in Figure 7.3. In what follows, we describe in more detail the steps shown in Figure 7.3.

Assume a tag like *Paris* to be classified in a context (see Section 7.1.2) as depicted in Figure 7.2

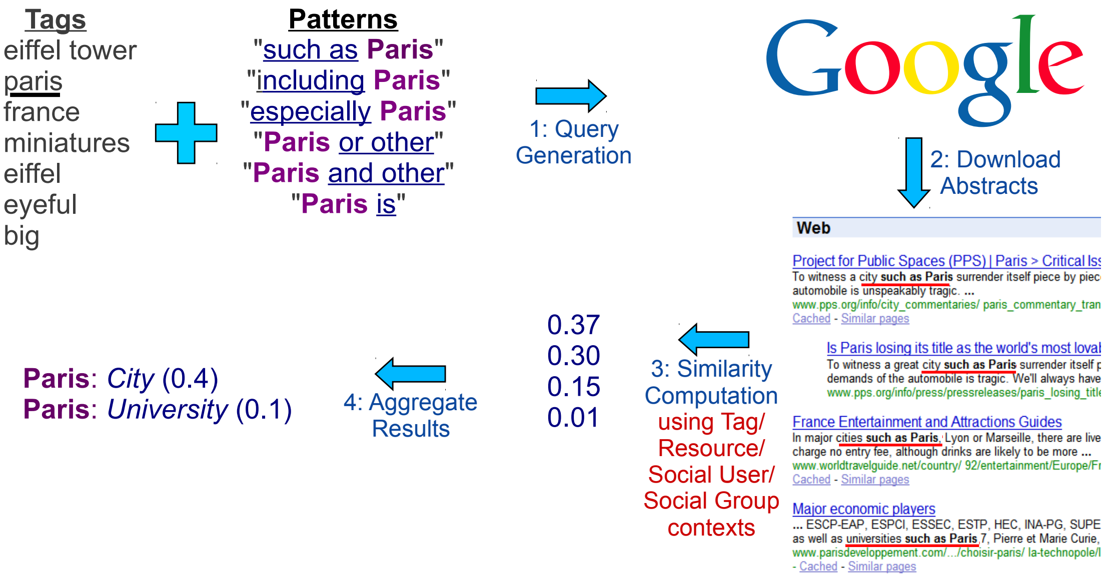

Figure 7.3: Process of T-KNOW.

Step 1: Queries are generated by concatenating the tag and the clues, e.g. "such as **Paris**" is a query generated by combining the clue "such as" and the tag "Paris"

Step 2: The queries are searched using a web search engine and abstracts of search results are downloaded, e.g. "To witness a **city** such as *Paris* surrendered itself . . . " is a search result abstract downloaded for the query "such as Paris"

Step 3: The similarity between each abstract and context of tag is computed, e.g. between the abstract "To witness a city such as Paris . . . " and context of the tag "Paris" (Eiffel tower, France, miniatures). If similarity is above a certain threshold value, then depending upon the clue used, the abstract is matched against the pattern, e.g. the abstract "To witness **city** such as *Paris* . . . " is matched against the Hearst pattern (Hearst, 1992) "**CONCEPT** such as (*INSTANCE*,?)+ ((and—or) INSTANCE)", where CONCEPT is the expected category and INSTANCE is the tag. Hence "City" is extracted as an expected category of the tag "Paris" from this abstract.

Step 4: The results are aggregated and the category having highest similarity with the tag's context is returned, e.g. for the tag "Paris" the category "City" is returned, because it has higher similarity than the other category e.g. "University"

The pseudo-code for T-KNOW is shown in Figure 7.4. *CN* is the total number of clues used. *clue(t,i)* is a function which returns a *query* string by concatenating the tag (t) with a clue (i) (value of i ranges from 1 to CN). This query is searched on the web search engine using an API[1]. The function *download_search_abstracts(query,n)* takes the query (*query*) and number of abstracts required (n) as parameters and returns the abstracts of search results found for the given query. The cosine measure is calculated between each abstract (a) and context (ctx) of the tag (t). If the value of the cosine measure is above a certain threshold, then the abstract (a) is considered for further processing. Patterns (find the complete list in (Cimiano et al., 2005), and example in step 3) for clue i are matched against the abstract a using the function *pattern_match(a,i)*. If the pattern is matched, then the category of the current tag is extracted. The category having the highest similarity with context of the tag is returned.

There are multiple ways for computing the similarity between the search result and the tag depending upon the context of the tag. We have proposed four methods (see Section 7.1.2) for selecting the context of a tag. For measuring similarity between the search result and the context of a tag, the cosine measure is computed between the bag of word representations of the abstract of the downloaded search result $\vec{a}$ and the context $\vec{c}$ of the tag t. If this cosine measure is above a certain threshold value, the result is considered for further processing.

7.1.4 Evaluating T-ORG based Classification

In order to evaluate our system, we have used images, tags, user, and group information from the Flickr website. We asked two persons to classify the data into four categories. We have then classified the same data set using T-KNOW in order to evaluate T-KNOW.

[1]For experiments presented in Section 7.1.4, we have used the Google API.

```
TKNOW(Tag t, Context ctx) {
  for i = 1 to CN {
    query = clue(t,i)
    abstracts = download_search_abstracts(query,n);
    foreach a in abstracts {
      sim = calculate_similarity(a,ctx);
      if (sim > threshold) {
        if (pattern_match(a,i)) {
          c = get_category(a);
          Res[c] = Res[c]+sim;
        }
      }
    }
  }
  return maxarg_c Res[c];
}
```

Figure 7.4: Pseudo-code of T-KNOW.

7.1.4.1 Experimental Setup

To organize tags into predefined categories, we have chosen four categories “Person”, “Location”, “Vehicle”, and “Organization”. To get ontologies related to these categories, we have searched Swoogle[1] (Ding et al., 2004) for general purpose ontologies and used the ontology OntoSem[2]. From this ontology, we have used concepts and sub-concepts of *vehicle*, *organization*, *place*, *geopolitical-entity*, and *human* as categories. We have used a total of 932 concepts as categories from this ontology.

After selecting the categories, we have gathered data from groups present at the Flickr website. Users post their images to the different groups on the Flickr website. One group usually contains images related to the topic of that group. For example, the vehicles group contains images of vehicles. We have searched for groups related to the topics (i) people, (ii) locations, and (iii) vehicles

[1] `http://swoogle.umbc.edu/`

[2] `http://morpheus.cs.umbc.edu/aks1/ontosem.owl`, last accessed in October 2010

using the group search facility provided by the Flickr API, and then selected three groups from each topic. We have selected only those groups which had at least 100 images and 25 members. The groups selected were candid_celebrity, 35212032@N00 (famous people), politicians, CarDirectory, classic_cars, vehicles, PraiseAndCurseOfTheCity, signcity, and cities. Out of these groups, only the "famous people" group had 27 members and 165 images, all other groups had at least 100 members and more than 500 images. We have then randomly selected 21 images from each of these nine groups. There were a total of 1754 tags in all of these 189 images.

We asked two persons K and S (human classifiers) to classify the tags into the classes defined in the ontology. They did not have any kind of information about this research and method. They have classified all the tags regardless of the language and spelling mistakes, which has of course affected the results of T-KNOW because the algorithm of T-KNOW uses English patterns for identifying categories. For example, the users have classified the tags *Russia* and *Russland* (German word for Russia) as location, whereas T-KNOW was unable to identify *Russland*, as this is not an English word and hence is not supported by the pattern library used. A spreadsheet was provided to each human classifier with resources, tags, and links to the original Flickr images, Wikipedia, and the web search engine. For example, if a user finds a tag *Essen* (a German city as well as the German word for meal) and is unable to decide about its category, he can view the image (in which this tag is present) on the Flickr website, if this image is not helpful to identify the tag, he can search it in Wikipedia[1], and still if it is unclear, then he can find it in a web search engine[2]. Human classifiers (K and S) agreed upon a classification of only 1166 tags out of 1754 tags.

7.1.4.2 Results

This section contains the results obtained by classifying tags using T-KNOW with different contexts and threshold values. Table 7.2 shows the number of tags and resources classified manually (by user K) and using the T-KNOW algorithm with the threshold value of 0.0 and the Social Group (SG) context.

[1] `http://en.wikipedia.org/wiki/Special:Search/essen`
[2] `http://www.google.com/search?hl=en&q=essen`, last accessed in October 2010

Table 7.2: Number of tags and resources classified per category by User K and T-KNOW with Threshold = 0 and the Social Group context.

	Resources		**Tags**	
Category	**User K**	**th=0, SG**	**User K**	**th=0, SG**
Location	139	155	519	485
Organization	39	54	89	67
Person	86	107	287	229
Vehicle	69	64	259	109
Other	155	177	600	864

We have used F-measure and Cohen's Kappa for evaluation of our method. F-measure is a common measure in information retrieval. To compute the F-Measure we set:

$$A = \textit{number of correct classifications by test}$$
$$B = \textit{number of all classifications by Gold Standard}$$
$$C = \textit{number of all classifications by test}$$

Then we define the Precision, the Recall, and the F-measure as follows:

$$\text{Precision} = \frac{A}{C} \tag{7.13}$$

$$\text{Recall} = \frac{A}{B} \tag{7.14}$$

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7.15}$$

In our evaluation, user K is the gold standard, and test is either the user S or the system T-ORG. Figure 7.5 displays the F-measure with user K defining the gold standard and T-KNOW using different threshold values and contexts and it also shows the F-measure of the classification of user K and user S (shown as a constant line).

Due to the possibility of classification that might occur just by chance, we have also calculated the Cohen's Kappa (Cohen, 1960) between a user's classification

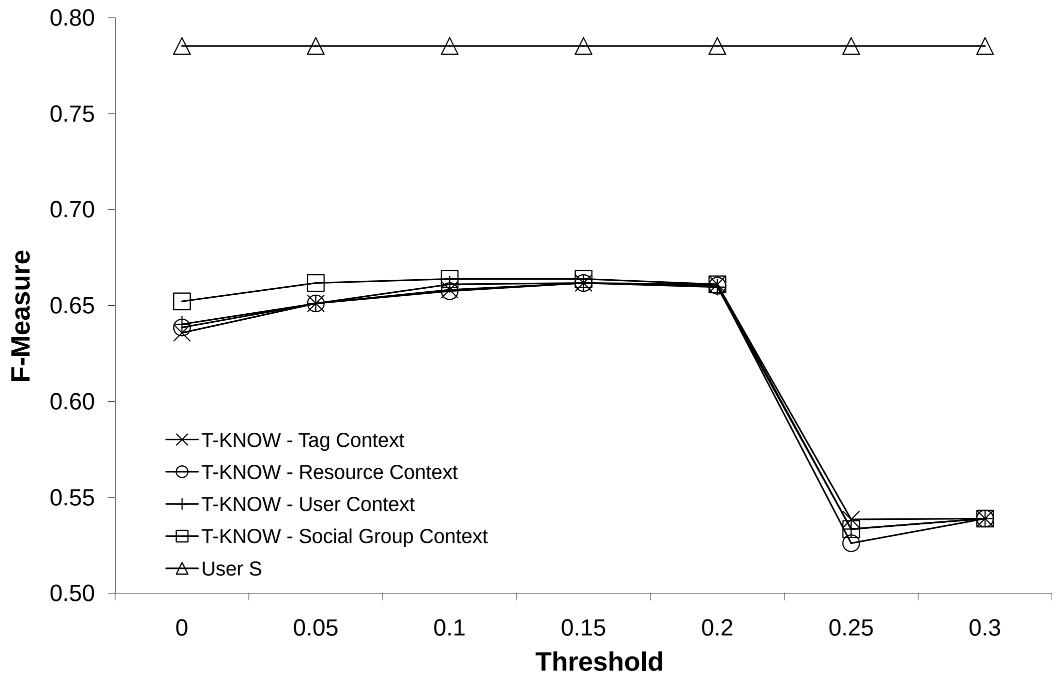

Figure 7.5: F-Measure with user K defining the gold standard.

and the system's prediction. Cohen's Kappa is defined as:

$$K = \frac{P_0 - P_c}{1 - P_c} \tag{7.16}$$

Where P_0 is the observed agreement between classifiers and P_c is the agreement occurred due to chance. If the two classifiers agree completely, then the value of Cohen's Kappa is 1. Figure 7.6 shows the Kappa values of the classification of user K and T-KNOW (with different threshold values and contexts) and it also shows the Cohen's Kappa value between the classifications of user K and user S (shown as a straight line).

The task of organizing resources by classifying tags in a folksonomy is not trivial. It is observed that two humans classifying the same data set might not totally agree with each other, as observed in the case of human classifiers of user K and user S, the kappa value was 0.53, whereas this value would be 1 in the case of complete agreement between the classifiers.

Table 7.2 shows the number of tags and resources per category. The difference between number of resources or tags classified by different classifiers per category

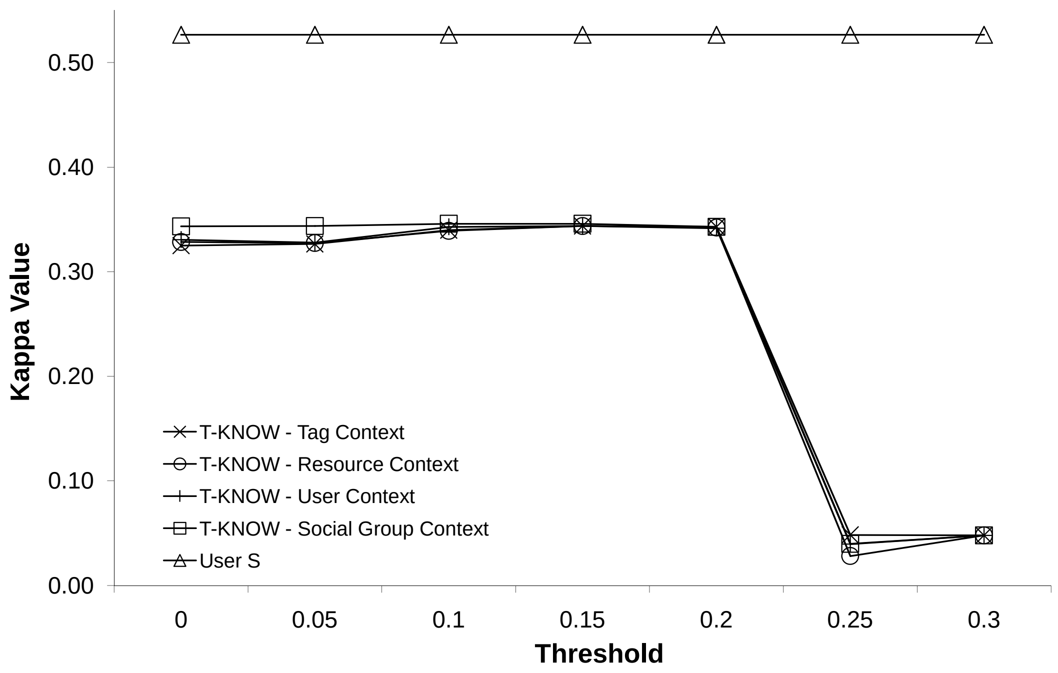

Figure 7.6: Cohen's Kappa values for classification of T-KNOW and User S with user K defining the gold standard.

is small. As the average resources per user were 1.39 in the data set, the difference between F-measures of Resource (R) and User (U) contexts is hardly visible. We believe that if there are more resources per user, then the results of classification will be different for these context types. The best F-measure obtained was 0.66 with the social group context for the threshold values of 0.10 and 0.15.

The F-measure is affected by the problem of classification by chance. Therefore we have calculated Cohen's Kappa (Cohen, 1960) to measure the agreement between two users and between T-KNOW and user K. The majority class ("Other" in our case) scores zero in Cohen's Kappa (Cohen, 1960). F-measure lacks this property. The Cohen's Kappa between classification of users K and S was 0.53 (shown as a straight line in Figure 7.6), which shows the disagreement between the classifications of human users. Best kappa value for gold standard (user K) was 0.35 with Social Group (SG) context and using threshold of 0.10 or 0.15.

The results show that, the different approaches for selecting a context are statistically not significantly different. Among the different contexts of the tags, the Social Group (SG) context has given slightly better results. This is because the tags which are chosen as context belong to the same type of resources/images (as a group mostly contains same type of resources). In the case of other contexts, the tags of the resources with different subjects are selected as context, which affects the results of the classification. In the next section, we exploit the social groups for classifying resources in a folksonomy.

7.2 Exploiting Social Groups for Classification

As discussed in the previous section, the social groups on folksonomies like Flickr provide useful contextual information in identifying the semantics of the tags. We have used the social group context in the T-ORG system to identifying the categories of the tags. In this section we focus on exploiting the information available in the social Flickr groups for identifying the semantics of the resources, particularly for identifying images representing the landmarks of a city. The literature refers to the task of identifying landmark images as the *landmark finding* problem.

Researchers have proposed different applications for solving the landmark finding problem. One of such applications called *World Explorer* (Ahern et al., 2007) is the current state-of-art system. The system has a reasonable performance, but it only works with geo-tagged photos (supplied with geographical coordinates). The problem is that many interesting places around the world are still represented by photos without geo-tags and their landmarks cannot be found using World Explorer. The focus of our research is to exploit the tagging features and the social Flickr groups to train a classifier with minimum efforts which can identify the landmark photos.

Recognizing a landmark in a photo is a hard task: First, content-based image analysis has very limited capabilities to solve this problem in general, given that photos are taken in different light and weather conditions, from different viewpoints and angles. Second, text-based or tag-based methods are much more appropriate for this task, but they do not have extra information if a tag represents a landmark or a family photo taken in a city. We propose to obtain this extra information from social groups in which users are involved. Flickr is enriched with plenty of photo groups related to landmarks, cars and other types of objects and themes. We exploit these groups for classifying resources related to the landmarks of a city.

7.2.1 TG-SVM for Landmark Classification

We propose a method TG-SVM (Tag Group Support Vector Machine) which exploits the tags and the social Flickr groups to train a classifier to identify landmark photos and tags. The method requires minimum human efforts. It only requires the links to the relevant Flickr groups. The system automatically trains a classifier based on the data retrieved from the Flickr groups. The method also ranks all the suggested relevant tags by their representativeness.

It is also possible to generalize our approach for other problems like car finding, mobile phone finding, etc. Although, due to high cost of user studies, in this chapter we test the performance of our method for landmarks only. The proposed solution is among the first ones to solve the landmark finding problem by exploiting tags and information from the photo communities. It does not use low level image features or GPS-coordinates. The presented user study shows that our approach outperforms the state-of-the-art World Explorer.

For the rest of the chapter we will consider that the landmark finding application has to automatically create a summary of photos, giving a comprehensive overview of landmarks at some place of interest. We will decompose this task into several sub-problems, as presented in Figure 7.7.

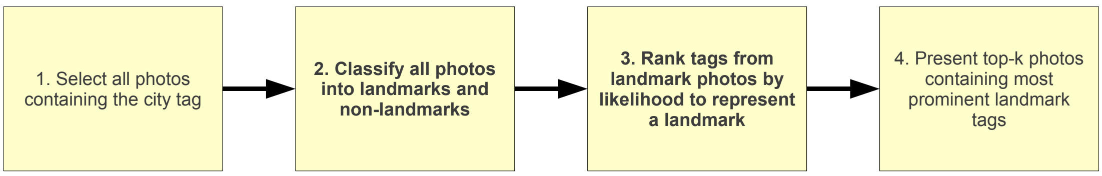

Figure 7.7: Decomposition of Landmark Finding Problem.

The first step consists of selecting a set of photos related to a particular city. Since we do not consider geo-tagged photos, we rely on a simple heuristic of having the city name (and also the country name in the case of an ambiguous city name) as a tag associated with a photo. This way we may miss many relevant photos, but for our task it is not a problem, since we still get a lot more photos than we need for a summary generation. In the second step all collected photos are automatically classified as either landmarks or non-landmarks.

It is important to understand that at this point, we do not have a summary of city landmarks. We have just a list of pictures classified as landmark or non-landmark. What we want to achieve is a list of names representing city landmarks and based on these names create a comprehensive city landmark summary. In the third step, tags of the photos classified as landmarks are ranked according to their likelihood of representing city sights. Once a ranking score is available for all tags in the set, in the fourth step we select top-k most representative tags. For each of these k tags we retrieve a Flickr photo which has as tags both the name of the city, as well as the landmark tag.

For returning a Flickr picture satisfying the conditions described above, we use the Flickr API[1] for tag-based search and sort the pictures by relevance. In this chapter, we focus on the two most important steps of the Figure 7.7, which are the second and the third steps.

In the following we present the details of the main sub-problems composing our landmark finding method. We focus on step 2, classification of photos into landmarks and non-landmarks, and step 3, selecting the most representative landmark tags.

For understanding the algorithms presented in this section, we reuse the same formalization of folksonomies ($F := (U, T, R, Y)$) as discussed in Chapter 3 and earlier in this chapter, in Section 7.1.2.1. In addition to the standard formalization, we define some additional terms. $f_r(t)$ denotes the number of times a tag t appears with a resource r. The normalized tag frequency $TF_r(t)$ of a tag t in a resource r is then defined as follows:

$$TF_r(t) = \frac{f_r(t)}{\sum f_r(t')}, (u, t, r) \in Y, (u, t', r) \in Y, t' \in T, u \in U, \tag{7.17}$$

Inverse Resource and User Frequencies, like Inverse Document Frequency in IR,

[1]`http://www.flickr.com/services/api`, last accessed in October 2010

are computed as follows:

$$IRF(t) = log\left(\frac{\mid R \mid}{\mid \{(t,r), u \in U, r \in R, (u,t,r) \in Y\} \mid}\right) \tag{7.18}$$

$$IUF(t) = log\left(\frac{\mid U \mid}{\mid \{(u,t), u \in U, r \in R, (u,t,r) \in Y\} \mid}\right) \tag{7.19}$$

From the set of pictures containing a city tag, we want to select photos representing landmarks. For classification we use a SVM (Support Vector Machine) binary classifier (Vapnik, 1999). SVM is a state-of-the-art method for classification. We use the SVMLight implementation (see (Joachims, 2002)) of SVM. For every picture we create a feature vector based on the tags which were used to annotate it and the SVM classifier assigns each photo to either "landmark" or "non-landmark" category. We assign weights to the tags in the feature vectors based on the usage of tags among resources and users as follows:

$$F(r) = [TF_r(t_1) \cdot IRF(t_1), TF_r(t_2) \cdot IRF(t_2), \cdots, TF_r(t_{|T|}) \cdot IRF(t_{|T|})] \tag{7.20}$$

We tested several weighting schemes, however, the combination given by Equation 7.20 provided the best results.

One of the main challenges for SVM or other machine learning technique is to create a good training set. Once a model is learned based on the labeled data from the training set, the SVM classifies unseen examples based on the learned model. Our hypothesis is that some of the Flickr groups like "Landmarks around the world" can serve as positive examples, while arbitrary general groups, like "Birds" or "Airplanes" represent negative examples. The idea to use the Flickr groups as training data can be used for any arbitrary photo classification task beyond the landmark finding problem. If a relevant group of photos exists on Flickr, one can use it as a training data to find more photos on the same topic within Flickr. For example, "CAR directory" or "Mobile Phones" groups can be helpful for finding photos of cars and mobile phones. Nevertheless, applicability of the Flickr groups for such tasks needs to be studied with additional experiments.

Once we have selected a set of city photos and filtered only landmark-related ones, the third step consists of ranking the tags by how well they represent landmarks. What we would like to achieve is a ranked set of tags representing landmarks specific to a particular city. When looking at the whole dataset, we would like to give low score to common tags. The assumption is that representative landmark tags appear in landmark photos, but not very common among the whole collection of images (globally). Let us consider R as the set of all photos (both landmark and non-landmark related ones), and T the associated set of tags. Supporting this first assumption, we compute IRF (see Equation 7.18) of the considered tag. If a tag is frequently used to tag photos in the dataset, it has a low $IRF_{R,T}(t)$[1] value and vice versa. Similarly, if a tag is globally very common amongst users, it must be scored low. This is achieved by computing IUF, $IUF_{R,T}(t)$ (see Equation 7.19).

After defining global scoring factors, we come to local measures computed on part of the collection with landmark photos only. When considering the dataset containing only pictures associated to a particular city and classified as landmarks, our assumption is that common tags should be scored high. Let us represent the set of landmark-related photos selected for a city as R_c and the corresponding tag set as T_c. If a tag is common among the photos for a particular city, this tag might represent some famous entity of the city, e.g. some museum, or an old and famous building. Let $nrt_c(t)$ be a number of times a tag t appears within landmark photos for a city c. Then we compute the normalized *City Tag Frequency*, $CTF(t)$, as follows:

$$CTF(t) = \frac{nrt_c(t)}{MAX(nrt_c(t'))}, t, t' \in T_c \tag{7.21}$$

Similarly, if a tag is used frequently by users, then it might represent a landmark or a famous place of the city. Let $nut_c(t)$ be the number of users using a tag t for the landmark photos for a city c. We compute the normalized *City User Tag Frequency* $CUTF$ as follows:

$$CUTF(t) = \frac{nut_c(t)}{MAX(nut_c(t'))}, t, t' \in T_c \tag{7.22}$$

The decision values returned by the SVM classifier against the classified photos

[1]Computation is relative to R and T

represent a confidence measure of the classification. Let d_r be the decision value for the photo r and let R_t be all the resources associated with a tag t. The confidence value $CONF(t)$ for the tag t is calculated as:

$$CONF(t) = log\left(\sum_{r \in R_t} d_r\right) \tag{7.23}$$

We combine all the above mentioned factors that affect the ranking of the tags and compute a representativeness score for each tag t occurring along with the resources classified as landmarks of a city c. The representative score of each tag for a city c is computed as follows:

$$SCORE(t) = IRF_{R,T}(t) \cdot IUF_{R,T}(t) \cdot CTF(t) \cdot CUTF(t) \cdot CONF(t), t \in T_c \tag{7.24}$$

The overview of the system as described in this section is shown in Figure 7.8.

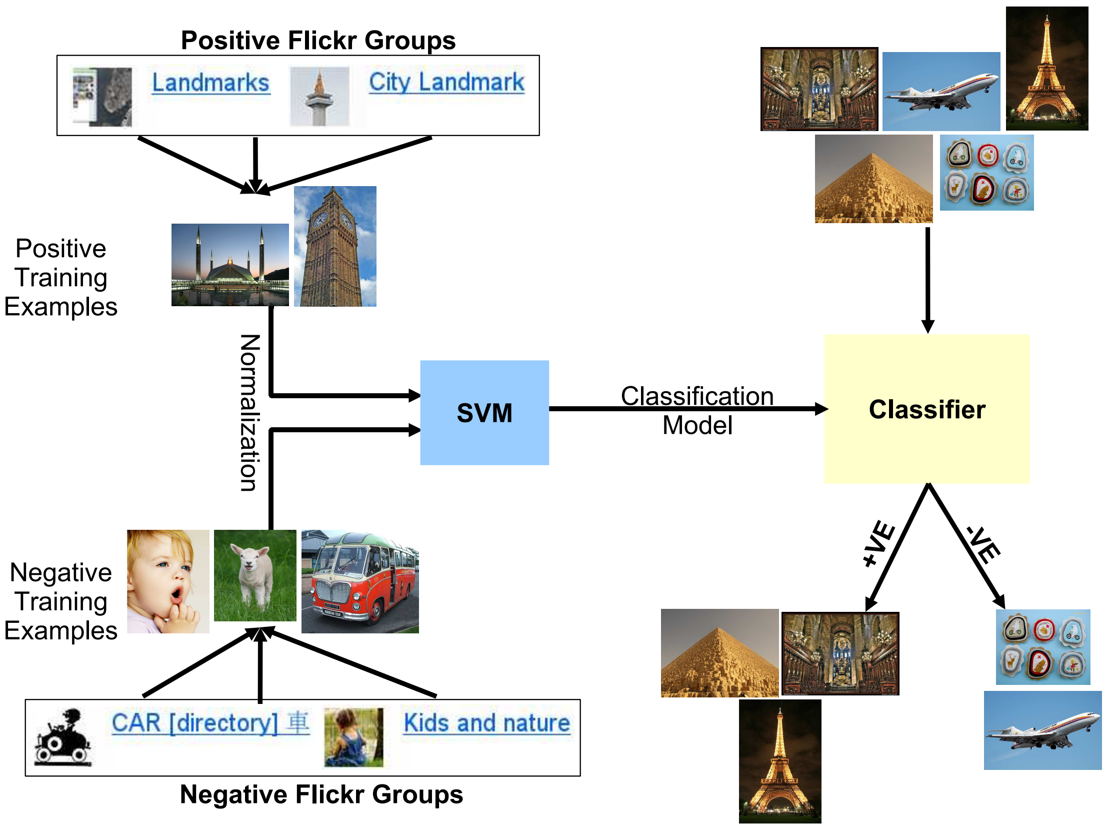

Figure 7.8: Example of a system classifying landmark photos.

7.2.2 Evaluating Landmarks Classification

The goal of our experiments is to evaluate the performance of the algorithm in finding city landmarks. We evaluate the accuracy of city landmark findings on more than 600,000 images for a list of 50 different cities. The results of this analysis have been collected through a user survey. Additionally, with this user study we also compared our results against results produced by an existing system called *World Explorer* which solves the same problem. Since World Explorer uses as input for its algorithms the pictures with the GPS information – i.e. richer input data than we needed – our aim was to obtain at least comparable quality.

7.2.2.1 Datasets and Evaluation Setup

We use different datasets for training and evaluating the proposed method. For the training dataset we use more than 400,000 images and for the test dataset we use more than 200,000 images. The detail for each of these datasets is given as follows:
Training Data (DS_{train}**)**: The training dataset was used for training the landmark vs. non-landmark classifier. The DS_{train} dataset was constructed by downloading 430,282 photos from several Flickr groups, uploaded by 57,581 different users. For positive examples we manually picked few groups like “Landmarks”, “Landmarks around the world”, “City Landmarks”, etc. As negative examples we used groups like “Airplanes”, “Birds”, “Cars”, “Mobile Phones”, etc. The dataset thus created contains 14,729 positive examples (related to landmark groups) and 415,553 negative examples (related to general groups). None of these 430,282 photos was included in the test dataset. This is real-world data so “positive groups” might also contain some non-landmark photos and vice versa. However, no additional noise reduction technique has been applied.
Test Data (DS_{test}**)**: This dataset from Flickr consists of pictures corresponding to 50 cities (for which World Explorer (Ahern et al., 2007) has at least 10 landmark tags), 60% European ones and the rest of 40% representing Asian, North-, South- American and Australian cities. We downloaded 4,000 to 5,000 photos per city, so that in total we gathered 232,265 photos, uploaded by 32,409 different users. Pictures from dataset DS_{test} were used for testing the classifier, after a

model was learned based on DS_{train}.

For the evaluation setup we recruited 20 volunteers among our colleagues. Each user was asked to evaluate two result sets for 10 randomly selected cities out of the set of 50. The selection process picked each city so that by the end of the experiment it was evaluated by at least 4 users. Two photo summaries were mixed on a single screen, with one result set created using our algorithm and one coming from the World Explorer API. The users did not know which system produced which photo, as the photos from the two systems were randomly interleaved. Each photo was supplied with a title and a single landmark tag produced by either World Explorer or by our algorithm and used to retrieve this photo. A radio button was placed near each photo, where users could select between "landmark", "non-landmark", and "don't know" options. The users were asked to judge if a photo is a landmark or not, in total producing between 400 and 500 judgments per user. The experiment took about 30 minutes per user.

Participants were instructed that a landmark photo must (1) contain a whole landmark or large part of it and (2) the landmark must be a main topic, not just a background for a person photo. Users were allowed to use photo title and tag as hints when they could not decide based on the picture only.

7.2.2.2 Evaluation Results

We observed quite different user assessment patterns, some participants considered as landmarks lots of photos, while some others accepted only few of them. As a first analysis, we measured the performance of the two algorithms for each city separately. Having each city assessed by 4 users, we applied simple majority vote aggregation function.

In Table 7.3 we present micro-average (averaged across all judgments per city) precision for the 50 analyzed cities. In total, our method (TG-SVM), outperformed World Explorer (WE) on 30 out of 50 cities (on the left part of the table), i.e. 60% of the cases. On average World Explorer has a precision value of 0.32, and our method, TG-SVM, 0.34.

Results in Table 7.3 show an interesting aspect: for some of the cities the precision values were very good, while for others they were poor. By inspecting

Table 7.3: Results of the Micro-Average Precision for 50 Cities. The left part of the table shows the results where TG-SVM performs better. The right part of the table shows the cities where World-Explorer performs better.

City	**World-Explorer**	**TG-SVM**	**City**	**World-Explorer**	**TG-SVM**
Amsterdam	0.33	**0.40**	Bucharest	**0.62**	0.36
Athens	0.21	**0.28**	Cairo	**0.73**	0.56
Barcelona	0.37	**0.44**	Chicago	**0.33**	0.28
Beijing	0.27	**0.29**	Cologne	**0.53**	0.48
Berlin	0.25	**0.48**	Florence	**0.67**	0.48
Birmingham	0.19	**0.28**	Genoa	**0.50**	0.42
Brasilia	0.40	**0.52**	Hannover	**0.75**	0.33
Buenos Aires	0.06	**0.28**	Leeds	**0.28**	0.24
Dresden	0.56	**0.75**	London	**0.29**	0.16
Glasgow	0.39	**0.40**	Madrid	**0.41**	0.32
Hamburg	0.19	**0.36**	Mexico City	**0.32**	0.08
Helsinki	0.23	**0.30**	Munich	**0.26**	0.25
Hong Kong	0.16	**0.21**	New York	**0.41**	0.27
Istanbul	0.40	**0.60**	Palermo	**0.50**	0.40
Liverpool	0.47	**0.56**	Paris	**0.45**	0.16
Los Angeles	0.09	**0.16**	Rio de Janeiro	**0.38**	0.20
Moscow	0.50	**0.75**	Singapore	**0.21**	0.13
Naples	0.13	**0.40**	Sydney	**0.26**	0.08
Oslo	0.16	**0.17**	Tokyo	**0.25**	0.19
Prague	0.20	**0.48**	Vienna	**0.37**	0.30
Rome	0.42	**0.52**			
Rotterdam	0.25	**0.48**			
Santiago	0.23	**0.28**			
Sao Paulo	0.04	**0.13**			
Seville	0.38	**0.46**			
Shanghai	0.43	**0.50**			
Stockholm	0.14	**0.16**			
Toronto	0.05	**0.24**			
Turin	0.25	**0.48**			
Yokohama	0.10	**0.16**			

the pictures corresponding to London, Paris, or Tokyo we could observe that the majority represented aerial views of the city where the landmarks were extremely difficult to identify, or were not present at all. In contrast to these, for Moscow, Istanbul, etc. the corresponding images depicted indeed the landmarks they also have been tagged with. Results are strongly dependent on the quality of the pictures included in the corresponding city set and consistency of users' tagging behavior.

In Table 7.4 we present the results from each user using macro-average precision, when all photos marked by users as landmarks are normalized by the total number of photos returned by an algorithm. Out of 20 users, 16 preferred our algorithm, 3 considered World Explorer-based results better and in one case the algorithms performed equally well. We obtained 12% improvement in precision with our method over World Explorer (statistically significant at level $\alpha = 0.001$ using paired t-test). These results support our hypothesis that landmark finding based on photo classification can replace geo-tagging based methods in situations where geo-spatial information is not available. They also show that our algorithm significantly outperforms state-of-the-art algorithms for landmark search. There was no particular tuning of the representativeness score as defined by Equation 7.24. Estimating the best combination of these parameters might give additional boost to results' quality.

7.3 Conclusions

This chapter addressed the problem of identifying semantics of tags and resources. Identifying semantics helps in browsing resources of a particular type. We have proposed two different methods for identifying semantics, one based on web resources call T-ORG and the second called TG-SVM (Tag Group Support Vector Machine) based on information available in social groups present on folksonomies. T-ORG uses T-KNOW for unsupervised classification of tags which exploits the web search engine and linguistic patterns. We have proposed to exploit different contexts of a tag. Experimental results show that the classification accuracy for this unsupervised method is indeed encouraging, especially in the light of the low agreement between the classifications done by two humans. The second method

Table 7.4: Results of the Macro-Average Precision for 20 Users for World Explorer and TG-SVM.

User	World-Explorer	TG-SVM
1	0.42	**0.44**
2	0.45	**0.47**
3	0.38	**0.45**
4	0.26	**0.43**
5	0.23	**0.28**
6	0.32	**0.39**
7	0.26	**0.30**
8	0.29	**0.35**
9	0.11	**0.16**
10	0.22	**0.29**
11	**0.45**	0.41
12	0.77	**0.78**
13	0.24	**0.29**
14	**0.22**	0.20
15	**0.40**	0.37
16	**0.27**	0.27
17	0.35	**0.40**
18	0.18	**0.25**
19	0.15	**0.21**
20	0.62	**0.63**
Average	0.33	**0.37**

TG-SVM addresses the problem of identifying resource related to a specific type. Particularly resources related to the landmarks of a city or a region using information gathered from tags and social groups. For finding relevant landmark-related tags we apply an SVM classifier for which the training data is extracted from the thematic Flickr groups. Our results show that the two-class SVM classifier effectively finds landmark photos based on the Flickr Groups training data. User evaluation results demonstrate that our method outperforms a state-of-the-art system. Apart from that the approach introduced here has a potential of being generalizable to help identifying not only city landmarks but also other topical photos, such as “cars”, “mobile phones”, etc.

Chapter 8

Conclusions

This thesis addresses the problems associated with everyday use of folksonomies. The problems that are particularly focused in this thesis include: improving search in folksonomies, recommending tags for resources, and browsing interesting resources. In the following paragraphs, we discuss the contributions of this thesis with respect to these problems.

Improving Search: The sparsity of tagging information in folksonomies leads to the problem of search in folksonomies. We have proposed methods to discover semantically related tags, and use these semantically related tags to reduce the sparseness in folksonomies. Although research work already exists for finding semantic relationships between tags, the contribution of this thesis to the existing research work is the utilization and formalization of contextual information of tags in discovering semantically related tags. Further contribution with respect to improving search in folksonomies includes the exploitation of semantically related tags for enriching the folksonomy data. The experiments presented in this thesis show that it is possible to improve search in folksonomies, particularly for the queries which have only a few relevant resources in the folksonomy.

There could be many applications where results of this thesis can be applied. One is the search in folksonomies. Especially, when a user faces difficulties in finding some particular resources, enriched data can be used to facilitate him. Another application could be the utilization of enriched data in other methods like tag recommendation or clustering. As discussed in this thesis, the results of tag recommendation are affected by the sparsity of the input data. The enriched data can be used to address this problem of sparsity and might help in developing improved tag recommendation methods.

Tag Recommendation: The problem of assigning tags to the resources is evident from the fact that in a data collection of *54* million images from Flickr, on average each image is associated with only *3.1* tags. This could be due to the lack of motivation among users in tagging their resources. In this thesis, we presented a framework for recommending tags to the users, in particular for the new resources which have not been already tagged. The main contribution of the presented framework is that, it allows comparing the performance of different rich media features in the process of tag recommendation. The proposed tag recommendation framework does not require supervised learning, it learns its classification model from already tagged resources and makes tag recommendations for new resources based on their content or metadata information. In comparison, most of the tag recommendation systems require an input of few tags for the new resource to suggest further tags.

The tag recommendation system can be incorporated into a folksonomy, so that when the users upload their resources, they are given suggestions for possible tags related to their resources. The proposed framework can also be used to tag existing untagged resources.

Browsing Interesting Resources: To facilitate the users in browsing interesting resources in folksonomies, we have proposed two novel methods for identifying interesting resources. The first method *T-ORG* and its classification algorithm *T-KNOW* identify the semantics of tags by exploiting the information available through web search engines along with lexico-syntactic patterns. The usage of web search engines and lexico-syntactic patterns minimize the need of supervised training which is required for learning a classifier. Although web search engines and lexico-syntactic patterns have already been used by other researchers to annotate web documents, our contribution in this thesis is twofold: first we formulate the problem of identifying semantics with respect to folksonomies, and second we use the information available in different contexts like users or social groups. Exploiting the contextual information is the novelty of *T-ORG* and had not been explored before.

The second method for identifying semantics of resources exploits information available in online communities. The method *TG-SVM* learns information about landmarks (and possibly other categories) from Flickr groups. *TG-SVM* is the first method to utilize the information available in Flickr groups to identify landmark photos. The performance of *TG-SVM* remains comparable to other state-of-the-art methods.

Once the semantics of tags and resources are discovered, they can be used in a variety of applications. One such application is focused browsers, where users can browse the resources in which they are particularly interested. For example, before visiting a city, it would be interesting to have an idea about the information of landmarks and worth visiting sights in that city. Similarly current browsing facilities can be improved by adding semantics to them. For example, by showing the users a list of categories or facets in which he is interested, a user should be able to narrow down his browsing experience by selecting the categories of his interest.

The contributions of this thesis are not the ultimate solution in addressing the problems faced in folksonomies. There is still a lot of room for improvements and further research. The algorithms and methods implemented for this thesis were tested in a lab environment. Further research is required to integrate the proposed methods into live systems and to build intuitive user interfaces which make the outcomes of this thesis more tangible. The presented methods can be further enhanced by improving their individual steps like clustering, ranking, and classification etc. For example, the clustering method used in tag recommendation is K-Means, there are other sophisticated clustering methods which should improve the system performance. The methods used for discovering semantically related tags might also be improved by exploiting information available in ontologies and external data sources. It still remains to be investigated that the information used from online communities to identify landmark photos can also be used to identify other concepts like persons or events. The application of the *T-ORG* system in personalized environments also remains to be investigated, for example, for organizing the tags and resources of a user, based on his individual requirements.

Bibliography

Rabeeh Abbasi. Query Expansion in Folksonomies. In *Proceedings of 5th International Conference on Semantic and Digital Media Technologies, Semantic Multimedia*, Lecture Notes in Computer Science, Berlin, Heidelberg, 2010, to appear. Springer Verlag. 6, 7

Rabeeh Abbasi and Steffen Staab. RichVSM: enRiched vector space models for folksonomies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 219–228, New York, NY, USA, 6 2009. ACM. ISBN 978-1-60558-486-7. doi: http://doi.acm.org/10.1145/1557914.1557952. 6, 7, 11

Rabeeh Abbasi, Steffen Staab, and Philipp Cimiano. Organizing Resources on Tagging Systems using T-ORG. In *Proceedings of Workshop on Bridging the Gap between Semantic Web and Web 2.0 at European Semantic Web Conference 2007*, pages 97–110, Innsbruck, Austria, June 2007. 7, 8

Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, and Steffen Staab. Exploiting Flickr Tags and Groups for Finding Landmark Photos. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy, editors, *Proceedings of European Conference on Information Retrieval, Advances in Information Retrieval*, volume 5478 of *Lecture Notes in Computer Science*, pages 654–661, Berlin, Heidelberg, 4 2009a. Springer Berlin / Heidelberg. ISBN 978-3-642-00957-0. 7, 8

Rabeeh Abbasi, Marcin Grzegorzek, and Steffen Staab. Large Scale Tag Recommendation Using Different Image Representations. In Tat-Seng Chua, Yiannis Kompatsiaris, Bernard Mrialdo, Werner Haas, Georg Thallinger, and Werner Bailer, editors, *Proceedings of 4th International Conference on Semantic and*

Digital Media Technologies, Semantic Multimedia, volume 5887 of *Lecture Notes in Computer Science*, pages 65–76, Berlin, Heidelberg, 2009b. Springer Berlin / Heidelberg. ISBN 978-3-642-10542-5. 6, 7

Benjamin Adrian, Leo Sauermann, and Thomas Roth-Berghofer. ConTag: A semantic tag recommendation system. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics*, pages 297–304. JUCS, September 2007. 14

Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie Hui-I Yang. World explorer: visualizing aggregate data from unstructured text in geo-referenced collections. In *Proceedings of 7th ACM/IEEE Joint Conference on Digital Libraries*, pages 1–10, Canada, 2007. ACM. ISBN 978-1-59593-644-8. doi: http://doi.acm.org/10.1145/1255175.1255177. 17, 100, 106

Morgan Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, pages 971–980, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: http://doi.acm.org/10.1145/1240624.1240772. 9, 19

Sofia Angeletou, Marta Sabou, and Enrico Motta. Improving Folksonomies Using Formal Knowledge: A Case Study on Search. In *Proceedings of 4th Asian Semantic Web Conference, The Semantic Web*, volume 5926 of *Lecture Notes in Computer Science*, pages 276–290. Springer Berlin / Heidelberg, 2009. 16, 17

J. Arguello, J.L. Elsas, J. Callan, and J.G. Carbonell. Document representation and query expansion models for blog recommendation. In *Proc. of the 2nd Intl. Conf. on Weblogs and Social Media (ICWSM)*. AAAI, 2008. 12

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999. ISBN 020139829X. 64

Tamara Lee Berg and David Forsyth. Automatic Ranking of Iconic Images. Technical Report UCB/EECS-2007-13, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, Jan 2007. 17

Marin Bertier, Rachid Guerraoui, Vincent Leroy, and Anne-Marie Kermarrec. Toward personalized query expansion. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pages 7–12, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-463-8. doi: http://doi.acm.org/10.1145/1578002.1578004. 12

Bodo Billerbeck, Falk Scholer, Hugh E. Williams, and Justin Zobel. Query expansion using associated queries. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 2–9, New York, NY, USA, 2003. ACM. ISBN 1-58113-723-0. doi: http://doi.acm.org/10.1145/956863.956866. 12

Paolo Bolettieri, Andrea Esuli, Fabrizio Falchi, Claudio Lucchese, Raffaele Perego, Tommaso Piccioli, and Fausto Rabitti. CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2, 2009. 11, 38, 58, 60, 65, 81

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998. ISSN 0169-7552. doi: DOI:10.1016/S0169-7552(98)00110-X. Proceedings of the Seventh International World Wide Web Conference. 13

P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proceedings of the 13th international Conference on World Wide Web*, pages 462–471. ACM Press, New York, NY, May 2004. 16

P. Cimiano, G. Ladwig, and S. Staab. Gimme' the context: context-driven automatic semantic annotation with C-PANKOW. In *Proceedings of 14th International Conference on World Wide Web*, pages 332–341. ACM Press, New York, NY, May 2005. 16, 92, 94

J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. 97, 99

Kevyn Collins-Thompson and Jamie Callan. Query expansion using random walk models. In *CIKM '05: Proceedings of the 14th ACM international conference on*

Information and knowledge management, pages 704–711, New York, NY, USA, 2005. ACM. ISBN 1-59593-140-6. doi: http://doi.acm.org/10.1145/1099554.1099727. 12

Marco Cristani, Alessandro Perina, Umberto Castellani, and Vittorio Murino. Content visualization and management of geo-located image databases. In *Proceedings of Extended Abstracts on Human factors in computing systems*, pages 2823–2828, New York, NY, USA, 2008. ACM. 14

Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 325–332, New York, NY, USA, 2002. ACM. ISBN 1-58113-449-5. doi: http://doi.acm.org/10.1145/511446.511489. 12

L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A semantic web search and metadata engine. In *Proceedings of 13th ACM Conference on Information and Knowledge Management*, pages 652–659. ACM, 2004. 95

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/138859.138867. 13

Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd ed. edition, 2006. ISBN 1-55860-901-6. 62, 63

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/992133.992154. 16, 92, 93

Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *Proceedings of 31st international ACM Conference on Research and Development in Information Retrieval*, pages 531–538, New York, NY, USA, 2008.

ACM. ISBN 978-1-60558-164-4. doi: http://doi.acm.org/10.1145/1390334.1390425. 14

A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information Retrieval in Folksonomies: Search and Ranking. In York Sure and John Domingue, editors, *Proceedings of European Semantic Web Conference, The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426. Springer Berlin / Heidelberg, 2006. 3, 10

Jens Illig, Andreas Hotho, Robert Jäschke, and Gerd Stumme. A comparison of content-based tag recommendations in folksonomy systems. In *Postproceedings of the International Conference on Knowledge Processing in Practice*, page (to appear). Springer, 2009. 14, 15

Alexandar Jaffe, Mor Naaman, Tamir Tassa, and Marc Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *Proceedings of 8th ACM international workshop on Multimedia Information Retrieval*, pages 89–98. ACM, 2006. ISBN 1-59593-495-2. doi: http://doi.acm.org/10.1145/1178677.1178692. 17

R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag Recommendations in Folksonomies. In Joost Kok, Jacek Koronacki, Ramon Lopez de Mantaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *Proceedings of Knowledge Discovery in Databases*, volume 4702 of *Lecture Notes in Computer Science*, pages 506–514. Springer Berlin / Heidelberg, 2007. 13, 55

Robert Jäschke, Folke Eisterlehner, Andreas Hotho, and Gerd Stumme. Testing and evaluating tag recommenders in a live system. In Dominik Benz and Frederik Janssen, editors, *Workshop on Knowledge Discovery, Data Mining, and Machine Learning*, pages 44 –51, September 2009. 12

Yan'an Jin, Ruixuan Li, Yi Cai, Qing Li, Ali Daud, and Yuhua Li. Semantic grounding of hybridization for tag recommendation. In Lei Chen, Changjie Tang, Jun Yang, and Yunjun Gao, editors, *Web-Age Information Management*, volume 6184 of *Lecture Notes in Computer Science*, pages 139–150. Springer Berlin / Heidelberg, 2010. 15

Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 079237679X. 103

Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *Proceedings of 15th international conference on Multimedia*, pages 631–640, New York, NY, USA, 2007. ACM. 14, 17

Lyndon S. Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th international conference on World Wide Web*, pages 297–306, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: http://doi.acm.org/10.1145/1367497.1367539. 14, 17

Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 61–68, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-435-5. 15

John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.383970. 12

Marek Lipczak. Tag Recommendation for Folksonomies Oriented towards Individual Users. In *Proceedings of the ECML/PKDD Discovery Challenge Workshop*, 2008. 14

J. MacQueen. Some methods for classification and analysis of multivariate observations. In Lucien M. Le Cam and Jerzy Neyman, editors, *Proceedings* *of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967. 61

Larry M. Manevitz and Malik Yousef. One-Class SVMs for Document Classification. *The Journal of Machine Learning Research*, 2:139–154, 2002. ISSN 1533-7928. 64

B. S. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7 - Multimedia Content Description Interface*. John Willey & Sons Ltd, Chichester, UK, 2002. 58

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, June 1999. ISBN 0262133601. 30

K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979. 62

Leandro B. Marinho and Lars Schmidt-Thieme. Collaborative Tag Recommendations. In H. H. Bock, W. Gaul, M. Vichi, Ph. Arabie, D. Baier, F. Critchley, R. Decker, E. Diday, M. Greenacre, C. Lauro, J. Meulman, P. Monari, S. Nishisato, N. Ohsumi, O. Opitz, G. Ritter, M. Schader, C. Weihs, Christine Preisach, Hans Burkhardt, Lars Schmidt-Thieme, and Reinhold Decker, editors, *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 533–540. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78246-9. 55

Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, and Gerd Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *Proceedings of the 18th international conference on World Wide Web*, pages 641–650, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: http://doi.acm.org/10.1145/1526709.1526796. 11

Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5 (1):5–15, 2007. ISSN 1570-8268. doi: DOI:10.1016/j.websem.2006.11.002. Selected Papers from the International Semantic Web Conference, International

Semantic Web Conference (International Semantic Web Conference 2005). 11, 12

George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/219717.219748. 6, 29

Pierre-Alain Moëllic, Jean-Emmanuel Haugeard, and Guillaume Pitel. Image clustering based on a shared nearest neighbors approach for tagged collections. In *Proceedings of the international conference on Content-based image and video retrieval*, pages 269–278, New York, NY, USA, 2008. ACM. 14

Emily Moxley, Jim Kleban, and B. S. Manjunath. Spirittagger: a geo-aware tag suggestion tool mined from flickr. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 24–30, New York, NY, USA, 2008. ACM. 15

H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler. A Review of Content-Based Image Retrieval Systems in Medical Applications - Clinical Benefits and Future Directions. *International Journal of Medical Informatics*, 73(1):1–23, January 2003. 82

Tae-Gil Noh, Seong-Bae Park, Hee-Geun Yoon, Sang-Jo Lee, and Se-Young Park. An automatic translation of tags for multimedia contents using folksonomy networks. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 492–499, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1572026. URL `http://doi.acm.org/10.1145/1571941.1572026`. 10

Oded Nov, Mor Naaman, and Chen Ye. What drives content tagging: the case of photos on Flickr. In *Proceedings of the 26th annual conference on Human factors in computing systems*, pages 1097–1100, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: http://doi.acm.org/10.1145/1357054.1357225. 9, 19

Oded Nov, Mor Naaman, and Chen Ye. Analysis of participation in an online photo-sharing community: A multidimensional perspective. *Journal of the American Society for Information Science and Technology*, 61(3):555–566, 2010. doi: 10.1002/asi.21278. 9, 10, 19

Simon Overell, Börkur Sigurbjörnsson, and Roelof van Zwol. Classifying tags using open content resources. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 64–73, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-390-7. 16

Jeff Z. Pan, Stuart Taylor, and Edward Thomas. Reducing ambiguity in tagging systems with folksonomy search expansion. In *ESWC 2009 Heraklion: Proceedings of the 6th European Semantic Web Conference on The Semantic Web*, pages 669–683, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-02120-6. doi: http://dx.doi.org/10.1007/978-3-642-02121-3_49. 12

Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st international conference on Scalable information systems*, page 1, New York, NY, USA, 2006. ACM. ISBN 1-59593-428-6. doi: http://doi.acm.org/10.1145/1146847.1146848. 43

I. Peters. *Folksonomies: Indexing and Retrieval in the Web 2.0.* KG Saur Verlag Gmbh & Co, 2009. ISBN 978-3-598-25179-5. 1, 2

Adrian Popescu, Gregory Grefenstette, and Pierre Alain Moëllic. Gazetiki: automatic creation of a geographical gazetteer. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 85–93, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-998-2. doi: http://doi.acm.org/10.1145/1378889.1378906. 17

A. Rae, B. Sigurbjörnsson, and R. van Zwol. Improving tag recommendation using social networks. In *Proceedings of 9th international conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*, Paris,France, 2010. CID. 15

J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ, 1971. 65

Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining Association Rules in Folksonomies. In *Proceedings of Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-34415-5 (Print) 978-3-540-34416-2 (Online). 24

Patrick Schmitz. Inducing Ontology from Flickr Tags. In *Proceedings of Collaborative Web Tagging, workshop at International World Wide Web Conference*, May 2006. 16

Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: http://doi.acm.org/10.1145/1367497.1367542. 13, 55

J. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. PhD thesis, Graduate School of Arts and Sciences, Columbia University, New York, NY, 1997. 63

Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. `http://www.personalinfocloud.com/2005/02/explaining_and_.html`, 2005. 21, 25

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, NY, November 1999. ISBN 0387987800. 103

Sihem Amer Yahia, Michael Benedikt, Laks V. S. Lakshmanan, and Julia Stoyanovich. Efficient network aware search in collaborative tagging sites. In *Proceedings of Very Large Data Base Endowment*, volume 1, pages 710–721. VLDB Endowment, 2008. doi: http://doi.acm.org/10.1145/1453856.1453934. 10

Curriculum Vitae

Rabeeh Ayaz Abbasi

Contact Information

Institute for Web Science and Technologies,
Department Computer Science,
University of Koblenz,
PO Box 201 602,
56016 Koblenz,
Germany

+49 261 287-2765 (off)
+49 261 287-100-2765 (fax)

abbasi@uni-koblenz.de
`http://bit.ly/rabeeh`

Experience

- Researcher at University of Koblenz-Landau. October 2006 - Present
- Software Engineer at Digital Spinners. 2005 - 2006 (1 year)
- Lecturer at Riphah International University. 2004 - 2006 (2 years)
- Teaching Assistant at National University of Computer and Emerging Sciences. 2003 - 2004 (1 year)

Education

- National University of Computer and Emerging Sciences. MS, Computer Science, 2002 - 2004
- International Islamic University. BS, Computer Science, 1999 - 2002

Courses

- Summer School on Ontological Engineering and the Semantic Web (2007)
- German Language Course (B1). did deutsch-institut Frankfurt (2006)

Publications

- Abbasi, R. (2010). Query Expansion in Folksonomies. In *SAMT: 5th International Conference on Semantic and Digital Media Technologies*, to appear, Berlin, Heidelberg. Springer Verlag.
- Abbasi, R. and Staab, S. (2009). RichVSM: enRiched vector space models for folksonomies. In *HT: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 219–228, New York, NY, USA. ACM.
- Abbasi, R., Chernov, S., Nejdl, W., Paiu, R., and Staab, S. (2009a). Exploiting Flickr Tags and Groups for Finding Landmark Photos. In *ECIR: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 654–661, Berlin, Heidelberg. Springer-Verlag.
- Abbasi, R., Grzegorzek, M., and Staab, S. (2009b). Large Scale Tag Recommendation Using Different Image Representations. In *Semantic Multimedia*, volume 5887 of *Lecture Notes in Computer Science*, pages 65–76, Berlin, Heidelberg. Springer Berlin / Heidelberg.
- Abbasi, R. and Staab, S. (2008). Introducing Triple Play for Improved Resource Retrieval in Collaborative Tagging Systems. In *Proceedings of*

Exploiting Semantic Annotations in Information Retrieval (ESAIR), workshop at ECIR.

- Abbasi, R., Staab, S., and Cimiano, P. (2007). Organizing Resources on Tagging Systems using T-ORG. In *In proceedings of Workshop on Bridging the Gap between Semantic Web and Web 2.0 at ESWC*, pages 97–110, Innsbruck, Austria.

Interests

- Research
- Semantics
- Web 2.0
- Social Media
- Data mining