HYBRID TAG RECOMMENDATION IN COLLABORATIVE TAGGING SYSTEMS

by

Marek Lipczak

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
March 2012

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "HYBRID TAG RECOMMENDATION IN COLLABORATIVE TAGGING SYSTEMS" by Marek Lipczak in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dated: March 15, 2012

External Examiner: _____

Research Supervisor: _____

Examining Committee: _____

_____

Departmental Representative: _____

DALHOUSIE UNIVERSITY

DATE: March 15, 2012

AUTHOR: Marek Lipczak

TITLE: HYBRID TAG RECOMMENDATION IN COLLABORATIVE TAGGING SYSTEMS

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: Ph.D. CONVOCATION: May YEAR: 2012

Signature of Author

Table of Contents

List of Tables

List of Figures

Abstract

The simplicity and flexibility of tagging allows users to collaboratively create large, loosely structured repositories of Web resources. One of its main drawbacks is the need for manual formulation of tags for each posted resource. This task can be eased by a tag recommendation system, the objective of which is to propose a set of tags for a given resource, user pair. Tag recommendation is an interesting and well-defined practical problem. Its main features are constant interaction with users and availability of large amounts of tagged data. Given the opportunities (e.g., rich user feedback) and limitations (e.g., real-time response) of the tag recommendation setting, we defined six requirements for a practically useful tag recommendation system. We present a conceptual design and system architecture of a hybrid tag recommendation system, which meets all these requirements. The system utilizes the strengths of various tag sources (e.g., resource content and user profiles) and the relations between concepts captured in tag co-occurrence graphs mined from collaborative actions of users. The architecture of the proposed system is based on a text indexing engine, which allows the system to deal with large datasets in real time, while constantly adapting its models to newly added posts. The effectiveness and efficiency of the system was evaluated for six datasets representing a broad range of collaborative tagging systems. The experiments confirmed the high quality of results and practical usability of the system. In a comparative study the system outperformed a state-of-the-art algorithm based on tensor factorization for the most representative datasets applicable to both methods. The experiments on the characteristics of tagging data and the performance of the system allowed us to find answers to important research questions adapted from the general area of recommender systems. We confirmed the importance of infrequently used tags in the recommendation process and proposed solutions to overcome the cold start problem in tag recommendation. We demonstrated that a parameter tuning approach makes a hybrid tag recommendation system adaptable to various datasets. We also revealed the importance of the utilization of a feedback loop in the tag recommendation process.

List of Symbols and Abbreviations Used

NLP	Natural Language Processing
ECML/PKDD	European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
p	p-cores processing
sf	Singular fraction
τ	Kendall's τ correlation coefficient
O_i	Output value of node i in spreading activation algorithm
I_j	Input value of node j in spreading activation algorithm
$v_i \in V$	Node i in the set of graph nodes
$e_{ij} \in E$	Edge connecting nodes i and j in the set of graph edges
p_{merge}	Merging coefficient
LFU	Least Frequently Used
LRU	Least Recently Used
$recall$@5	Recall at 5
$F1$@5	F1-score at N
MAP	Mean average precision
PITF	Pairwise Interaction Tensor Factorization
FM	Factorization Machines
SVM	Support Vector Machines
Tf-Idf	Term frequency, inverted document frequency

Chapter 1

Introduction

Collaborative tagging is a recently popularized information management approach. Free-form tags allow large communities of users to collaboratively create accessible repositories of Web resources (e.g., references to scientific literature in *CiteULike*[1], bookmarks in *Delicious*[2] or programming questions in *Stack Overflow*[3]). Each resource is entered into the system in the form of a *post*, which combines a *resource*, a *user* who is posting it and a set of *tags*. Thanks to the tags, the traditional hierarchical data structure design based on directories created by system editors (e.g., *Open Directory Project*[4]) is replaced by flexible tag-based pseudo-taxonomies defined jointly by users. Tagging turns a cumbersome classification problem, in which each resource should be assigned a place in a predefined hierarchy of classes, into an unstructured categorization problem, in which each resource is related to a set of loose ad-hoc user-defined categories [22, 25]. Despite the fact that users have complete freedom in choosing their tags, it is widely believed that constant interaction with posts of other users leads to collaborative actions and emergence of a pseudo taxonomy, called *folksonomy*. Although the term folksonomy refers to the pseudo hierarchy of tags, it is often used to refer to the complete data structure created in collaborative tagging systems (folksonomy graph), or even as a synonym of collaborative tagging systems. To remain consistent with previous work we use the term folksonomy in all presented meanings. However, to avoid ambiguity we clearly state if we refer to folksonomic relations between tags, folksonomy data structure or folksonomy system.

Relations between three basic elements of a post allow us to represent the folksonomy data structure as a tripartite hyper-graph of resources, users and tags. Each post can be then understood as a set of hyper-edges that connect resource, user and tag. The *folksonomy graph* is a frequently used representation of the folksonomy data

[1] http://www.citeulike.org/
[2] http://delicious.com/
[3] http://stackoverflow.com/
[4] http://www.dmoz.org/about.html

structure [32]. From the perspective of our work, it seems reasonable to extend the graph by a fourth element — words extracted from the resource content. To avoid the need of extending the general formalism established for folksonomies [32] and to simplify the description, we use the concept of *tag profile*, a specific data structure that can be extracted from a folksonomy. Tag profile can be defined for an element of any type (i.e., user, resource, tag, content term) as a set of tags that co-occurred with the element in any of the posts gathered in the repository. The profile contains also the information about the number of such co-occurrences (*frequency*).

The simplicity and generality of the tagging concept has led to a wide variety of systems that use tags to organize information. Classification schemes have been proposed to organize the folksonomy systems [26, 68]. Folksonomy systems can be divided based on the tagging reasons (does it benefit the author or the audience of the post?) [26]. Another classification scheme could be based on the type of the posted resources or the audience [26]. In this work we refer to all of these aspects; however, the main focus is on a classification scheme proposed by Vander Wal [68], who divided folksonomies into *broad folksonomies* and *narrow folksonomies*. In broad folksonomies the same resource (e.g., a bookmark or reference to a scientific publication) can be added to the system by many users. These users are not the authors of the resource, but each of them can use a personal set of tags to describe it. In narrow folksonomies a resource (e.g., blog post or forum entry) can be added to the system only once by its author. The author is also the only person who tags the resource. As we will show, the two folksonomy types have different characteristics and usually are addressed with different tag recommendation techniques. This thesis, however, proposes a hybrid tag recommendation system applicable to both broad and narrow folksonomies.

1.1 The Tag Recommendation Task

One of the main advantages of tagging is the lack of a predefined classification system, which reduces the constraints imposed on users [25]. On the other hand, when users are not supported by the classification system, they have to come up with a set of descriptive tags on their own. This is a cumbersome and labour intensive task. The objective of a tag recommendation system (or tag recommender for short) is to ease this process and propose potentially useful tags. The ability to choose the tags from a set of proposed options reduces the cognitive effort of a generation task to

a recognition task [19]. Apart from its practical importance, tag recommendation is an interesting research problem. The traditional understanding of a recommendation task is assigning a set of potentially interesting items to a user [60]. In general, the task can be accomplished in two ways [1]: by utilizing the information about the previous actions of a community of users (collaborative filtering approach) or the attributes of users and items (content-based approach). Tag recommendation extends this two-dimensional space of users and items by the dimension of tags. The additional dimension increases the complexity of the problem, but at the same time opens new potential sources of information that can be exploited by the recommendation models. Discovery of the most useful tag sources was a key point of the design of our system.

Tag recommendation is a prediction task, since the system aims to predict a set of tags that are likely to be accepted by the user as good descriptors of the resource in the incoming post. Therefore, research on tag recommendation can bring useful insights about the modelling of collaborative tagging systems and more general aspects like the motivation of tagging, the importance and usefulness of tags from a personal and social point of view. Finally, in the long term, this work can potentially lead to automatic tagging systems, in which the work of users would be completely replaced by automatic agents, which would be able to crawl Web resources and place them in the folksonomic structure. We keep all of these aspects under consideration, however, in this work we are mostly interested in a practical tag recommender, which can be applied to a wide range of collaborative tagging systems.

1.1.1 Tag Recommendation — Practical Aspects

Thanks to large amounts of information accumulated in the folksonomy data structure, tag recommendation can be addressed with a wide variety of machine learning and information retrieval methods. However, unlike many machine learning and information retrieval tasks, tag recommendation does not suffer from insufficient user feedback. Each recommendation is shortly followed by the set of user-selected tags, which can be used to evaluate the recommendation and adapt the models used to generate it. On the other hand, constant interaction with the users puts hard constraints on the minimal system throughput and maximal response time. In addition, the great variability of tags and tagged resources creates other issues that must be considered while designing a practically usable tag recommendation system. Given the practical

aspects of the tag recommendation problem, as well as some common limitations of state-of-the-art recommenders, we defined six requirements for a practically usable tag recommendation system: (1) data sparsity, (2) open-ended vocabulary, (3) generality, (4) adaptability, (5) efficiency and (6) low maintenance cost. The first four requirements should be addressed in the conceptual design stage of the work, while the last two are more related to the implementation and further use of the system.

Data sparsity. In contradiction to the common picture of collaborative tagging systems, they do not produce a dense graph of relations between resources, users and tags. In most cases very little information about the incoming post is already present in the system. Obviously, a more precise recommendation can be provided for popular resources added to the system by users with long posting history, but such entries are just a small fraction of all encountered posts and the system should not be designed to process them exclusively. The system should be able to process all posts entered by the users.

Open-ended vocabulary. Although tag recommendation shares some characteristics with the multi-label classification problem [11], a classification algorithm cannot be directly applied to this task as it assumes a fixed number of classes (limited tag vocabulary). Tag vocabulary is open-ended and constantly extended by users, hence low-frequency tags and newly added tags should not be omitted in the recommendation process.

Generality. Each collaborative tagging system has its own specific characteristics (e.g., personal or social character of posts). These differences are likely to have impact on the tagging decisions made by users, hence they must be taken into consideration while designing a tag recommendation system. Manual tuning of system parameters has obvious limitations; therefore, the recommendation system should be able to automatically adapt to these characteristics.

Adaptability. Tag recommendation is a dynamic process. Each recommendation is instantly followed by the real tags entered by the user. This *feedback loop* constantly brings new valuable information to the system. In fact, in some cases (e.g., modelling

of user interests) the newly added posts are likely to provide the most accurate information. The system should be able to adapt its data repository and recommendation models to them.

Efficiency. Tag recommendations are expected to be delivered to the user shortly after the posting process is initialized by the user. This is the most crucial aspect of the tag recommendation problem. Regardless of the quality of produced tags, the system is useless if it fails to provide them to the user in real time.

Low maintenance cost. Despite its potential to ease the tagging process, tag recommendation is just an additional feature of a collaborative tagging system. To be practically usable the system should be easy to deploy and should not require periodic maintenance tasks (e.g., model re-training). The system should be also able to operate with limited computing resources.

1.2 System Outline and Example Scenario

The objective of our work was to create a conceptual design and an architecture of a tag recommendation system that meets all of the presented requirements. To achieve high quality of recommended tags we designed a hybrid tag recommendation system based on the four most useful sources of tags: the content of the posted resource, tags that were previously used for the same resource (resource profile), tags that were used by the user who is adding the current post (user profile) and finally the co-occurrence graphs that represent relations between content terms and tags. The tags from different sources are combined into a processing stream, the objective of which is to utilize specific advantages of the sources and contain their shortcomings. The detailed discussion of the processing stream and its stages is presented in Section 4.1. Here we present an example scenario that outlines the main points of the conceptual design of the system (Fig. 1.1).

A user who would like to visit the east coast of Canada is gathering information about interesting places she can visit during the trip. One of the encountered websites (`www.halifaxfortourists.ca`) contains information about Halifax. As Halifax is the largest city of Atlantic Canada this website could be useful in the future. The user decides to use a social bookmarking system (e.g., *Delicious* or *BibSonomy*) to save

the bookmark of the website. At the moment the user clicks the post button in her browser, the information about the post is transferred to the collaborative tagging system. The system receives the ID of the resource, which in the case of bookmarks is the website URL, together with the textual content of the resource including its title. As the user is logged into the system, her ID can also be retrieved. Given this information the system starts the tag recommendation process. Its objective is to predict, which tags would be chosen by the user for the given resource. In our tag recommendation system, the recommendation process starts with the tags extracted from the content of the resource, specifically its title ((`Halifax For Tourists --- Visitor Information Centre`)). The title is a short and precise source of potential tags. The set of content based tags is a narrow description of the resource, which represents solely the perspective of the resource author. The system is able to refer to its knowledge base recalling previous actions of all users that were tagging resources with similar content. For example, it is able to retrieve all tags that were used when the word "'tourists"' appeared in the content, namely: "travel", "travels", "tovisit" and many others. In general, the relations between content words and tags reveal tags that are synonyms of the word. Another potentially useful source of information are tags that were used together with the content word when it was chosen as a tag. On the contrary, these tags are likely to be related but not synonymous (e.g., "halifax" could be used together with "canada", "novascotia", "transport"). The related tags enrich the description of the resource, so they are added by the system to the pool of tags considered for recommendation.

Halifax For Tourists is a popular website that has been already tagged by other users. These tags formulate a tag profile of the website. The profile is a noisy source of tags, as it contains very personal tags (e.g., "placestovisit", "conferences08"). However, the collaborative actions of users clean the profile bringing the most descriptive tags (e.g., "halifax", "travel", "canada") to the top of the popularity ranking of tags for the website. Therefore, the most popular (i.e., the most frequently used) tags in the tag profile are added to the pool of tags considered for recommendation. So far, this pool contains a broad range of tags that are somehow related to the resource.

At the same time the system takes the user perspective into consideration. A large number of previously tagged resources creates a rich tag profile of the user. The profile is also very noisy as it describes all interests and activities of the user. It is

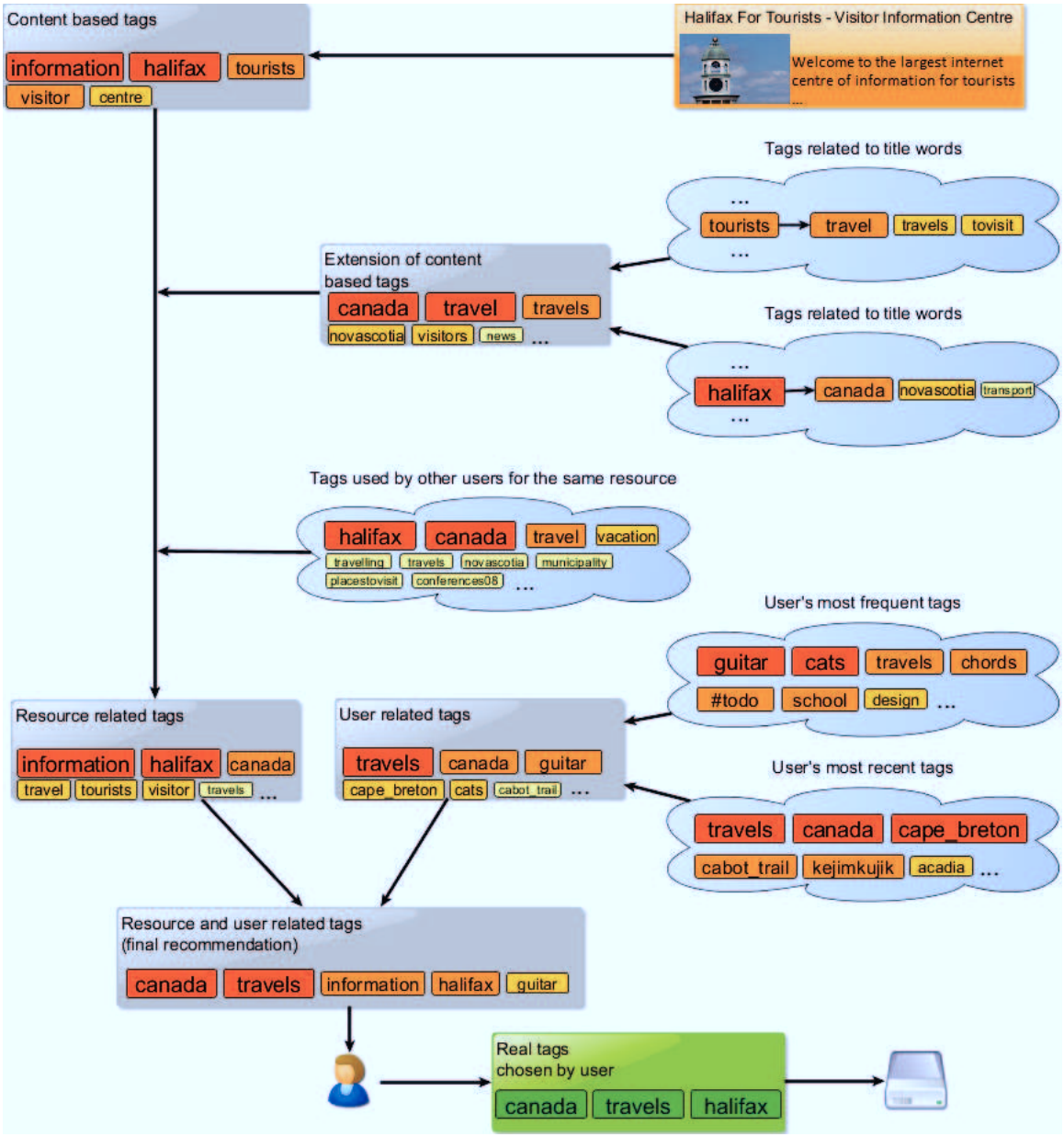


Figure 1.1: Example scenario of a tag recommendation process. The system uses various sources of tags to build the final recommendation. The only external source of tags is the content of the posted resource. The sources internal to the system and example tags that can be extracted from them are presented as clouds. The size and colour of a tag represent its importance. After the recommendation is presented to the user, she picks a set of tags. These tags are later used to update the data repository and parameters of the system.

not likely that all most popular user's tags ("guitar", "cats", "travels", "chords") would be applicable to the currently posted resource. The system can also use the most recently used tags ("travels", "canada", "cape_breton", "cabot_trail"). As she was recently gathering resources related to her Atlantic Canada trip the most recent tags contain general tags related to the trip together with specific tags that describe places to visit. The tags that are likely to be used for the current resource should be both popular and recent (e.g., "travels", "canada").

In the final stage of the recommendation process the system combines tags that are related to the resource and the user. The tags that can be found in both sets are most likely to be used for the current post. Based on previous user's actions the system knows that she likes to reuse her own tags, therefore, the personal tags are given more preference when the final set of recommended tags is compiled.

Given the recommendation, the user makes the decision about tags that are suitable for the posted website. Some of them can be taken from the set of recommended tags, other are entered manually by the user. By submitting the post the user becomes the reviewer of the tag recommendation process. If the recommendation was useful, a large number of recommended tags will be found among the user-chosen tags. The tags provided by the user are used to learn the system parameters. A learning algorithm incorporated into the recommender system performs automatic *parameter tuning* to optimize the results based on the tags entered to the system by users. This approach allows the recommender to automatically adapt to characteristics of a specific collaborative tagging system. In addition, the system uses the content of each entered post to update the resource and user profiles as well as the associations between content words and tags. Immediate access to real tags entered by the user is an essential feature of the tag recommendation process as it allows the recommender to significantly improve the quality of proposed tags.

To make the recommendation practically usable it must be delivered to the user instantly after the posting process is started. Otherwise the user would ignore the recommendations and start to enter tags on her own. The system uses a text indexing engine to represent the co-occurrence graph as well as resource and user profiles. In this way the information can be efficiently retrieved from the data repository. We should expect that some of the tags (e.g., "information"), resources and users will appear in the system more often than others. Our system exploits this property

to increase the efficiency of the recommendation process. The system has an additional layer of caches that contain the information about the most frequently or most recently queried elements. Cached elements can be retrieved much faster and significantly decrease the time needed to produce the recommendation.

All described steps (recommendation, parameter tuning and indexing) are performed in real-time while the post is processed. The system needs no additional re-indexing or re-training runs, which makes it simple to use and maintain from the administrator point of view.

1.3 Research Questions

The design and implementation of a fully operational hybrid tag recommendation system allowed us to throughly investigate the nature of the tag recommendation problem and find responses to a set of open research questions relevant to the problem in specific and to the design of recommender systems in general.

Research question 1 *What is the practical usefulness of various tag sources in the tag recommendation problem?*

One of the basic features of tagging systems is that they put no constraints on the users in terms of the tag vocabulary they want to use. To come up with a set of relevant tags the system has to retrieve them from the data stored in the tagging system and possibly other external sources (e.g., Wikipedia) or extract them from the content of the resource. One of the objectives of our work was to discover which of these act as reliable sources of tags. To achieve it we measured their performance in terms of the precision and recall of extracted or retrieved tags. We examined the following tag sources: the textual content of the resource with specific attention to the resource title, the tags previously assigned to the same resource by other users (resource profile), the tags used previously by the user (user profile), tags assigned to resources by "similar" users and tag co-occurrence graphs, which provide a set of tags related to a given tag or keyword.

Research question 2 *What is the importance of heavy-tail and long-tail elements in the tag recommendation process?*

The use of open-ended tag vocabulary leads to another practical question: are the infrequently used tags really important in the recommendation process? Most of the tag recommendation systems explicitly focus on the tags that have already been frequently used in the system. To determine what percentage of tag assignments is missed this way, we examined the characteristics of tag usage. Specifically, we observed what percentage of tags come from the small group of frequently re-used tags (the heavy tail of the distribution) and from the large group of infrequently used tags (the long tail of the distribution). We also looked at the performance of the tag sources for the most frequently and the least frequently occurring tags. The problem of infrequently occurring elements can be extended to resources and users. Data sparsity caused by the infrequently occurring elements creates the *cold start problem* [61], in which the recommendation system is not able to serve a large number of recommendation tasks because of insufficient information about the elements present in the task. To gain more information about the performance of tag recommendation systems in data sparsity conditions we gradually densified the datasets by removing the infrequently occurring elements. This provided an opportunity for comparison of our system with systems that focus specifically on the frequently occurring elements.

Research question 3 *Is a hybrid tag recommendation system that relies on several possible tag sources able to adapt to tagging style used in a specific tagging system?*

The users of each collaborative tagging system are likely to use a specific tagging style which depends on the type of stored resources, the community of users and most importantly the purpose of tags. In our work, we wanted to determine if a generic learning approach can be used to tune a hybrid tag recommendation system to a specific tagging style. To achieve it, we evaluated the performance of the system and the underlying tag sources for six datasets which are a broad representation of folksonomy systems.

Research question 4 *Can the feedback loop in the recommendation process be utilized to improve the quality of the recommended tags?*

The basic evaluation approach for recommendation systems assumes a strict separation of samples used to train and test the system. This approach; however, does not take into consideration the dynamics of the system in which new information is

constantly entered into the system. In our work, we were interested in the utilization of the feedback loop between the system and the user. One of the potential advantages of the feedback loop is the access to the most recent tagging decisions of the user, which can be considered as the context of the current recommendation. In this way our work contributes to a recently active area of context-aware recommenders [2]. Another possibility opened by the utilization of the feedback loop is online learning of the parameters used in the recommendation algorithm, which fits the work on stream mining [6].

1.4 Organization of the Thesis and Copyrights

The remaining part of the thesis is organized as follows:

Chapter 2 (Related Work) summarizes the related work that was done in the area of tag recommendation and a broader area of the tagging motivation. The discussion of the literature on tag recommendation is a significantly extended version of the discussion from our conference paper [48] presented at *ACM Recommender Systems 2010* conference[5]. The discussion of the tagging motivation problem was adapted from our conference paper [47] presented at *ACM Conference on Hypertext and Hypermedia 2010*[6].

Chapter 3 (Characteristics of Collaborative Tagging Data) introduces datasets used in the experiments on our recommendation system, together with the preprocessing steps. The main objective of the chapter is to describe the potential usefulness of various tag sources in the tag recommendation task. In a series of experiments we examined the statistical characteristics of tagging system datasets focusing on the data sparsity problem. Some of the discussion was adapted from the *ACM Hypertext and Hypermedia 2010* conference paper [47]. In addition, in experiments on p-cores pruning we quantified the amount of tag assignments removed in a commonly used preprocessing approach.

Chapter 4 (System Design) is a detailed description of the proposed tag recommendation system. The chapter includes the description of the conceptual design

[5]This work is based on an earlier work: Learning in Efficient Tag Recommendation, in RecSys '10: Proc. the 4th ACM Conference on Recommender Systems (2010) ©ACM, 2010. http://doi.acm.org/10.1145/1864708.1864741

[6]This work is based on an earlier work: The impact of resource title on tags in collaborative tagging systems, in HT'10: Proc. the 21th ACM Conference on Hypertext and Hypermedia (2010) ©ACM, 2010. http://doi.acm.org/10.1145/10.1145/1810617.1810648

of the system (Section 4.1) and system architecture (Section 4.2). The description of the system was adapted from the *ACM Recommender Systems 2010* conference paper [48].

Chapter 5 (Evaluation) discusses various evaluation techniques that can be found in the literature on tag recommenders and provides rationale for the evaluation methodology used in the thesis. The chapter presents the evaluation of the system from the perspectives of system effectiveness — the quality of recommended tags (Section 5.1) and efficiency — the throughput and response time (Section 5.3). Elements of the recommendation process that are evaluated include: the parameter tuning approach, the use of online content adaptation, the performance of specific tag sources and their impact on the final results, as well as the performance of the cache layer in system architecture. The evaluation was conducted on six real-life datasets. We used the time-stamps of each entered post to reproduce the process of folksonomy dataset formulation to create the most realistic off-line evaluation scenario. The results confirmed the ability of the system to produce high quality tags. At the same time, the system is practically usable for folksonomy datasets counted in millions of posts. These sections present a significantly extended version of the discussion from the *ACM Recommender Systems 2010* conference paper [48] and a journal article [49] from *ACM Transactions on Intelligent Systems and Technology* [7]. In addition, the chapter presents a comparative evaluation with two state-of-the-art systems (Section 5.2).

Chapter 6 (Additional Aspects of Tag Recommendation) discusses three specific problems related to tag recommendation task. The performance of the system for frequently used tags (Section 6.1), extraction of content-based tags 6.2 and learning of tagging patterns (Section 6.3). This chapter is a significantly extended version of the discussion from our journal article [49] from *ACM Transactions on Intelligent Systems and Technology*.

Chapter 7 summarizes the work highlighting the main contributions of the thesis and suggests potential areas of future work.

[7] This work is based on an earlier work: Efficient Tag Recommendation for Real-Life Data, in ACM Transactions on Intelligent Systems and Technology, 3, 1, (2011) ©ACM, 2011. http://doi.acm.org/10.1145/2036264.2036266

Chapter 2

Related Work

Reviewing the work related to our problem we should consider two areas of research: a specific area of tag recommendation and more general area of tagging motivation. Tag recommendation has recently become a very active field. We present a broad range of tag recommendation approaches with the strongest emphasis on their practicality. The objective of a tag recommendation system can be considered as the prediction of tags that a user would like to use for a resource. Therefore, while designing a tag recommendation system it is important to consider experimental studies on the motivation of tagging. We present an overview of tagging models proposed to explain user tagging motivations and, what follows, observed properties of folksonomy data structure. Finally, we present the studies of the impact that textual content of a resource can have on tagging decisions.

2.1 Tag Recommendation Systems

Despite the fact that tag recommendation is a relatively new problem, a wide variety of tag recommendation algorithms has already been presented. Tag recommendation systems can be divided into three categories: graph-based, content-based and hybrid systems. Graph-based systems utilize the relations between users, resources and tags represented in the folksonomy graph. In most cases graph-based recommendation is addressed with collaborative filtering methods. Content-based systems are based solely on the textual metadata related to the resource. Hybrid systems combine these two types of input.

2.1.1 Graph-based Recommendation

Jäschke et al. [35] proposed a graph-based tag recommendation system based on FolkRank, an adaptation of PageRank to folksonomy graph. Given a resource-user pair the system increases their weights in the folksonomy graph and runs FolkRank

to spread the weights in the graph. Tags with the highest weights are returned as recommendations. The process has to be run for each incoming post, which makes the system inefficient. Symeonidis at al. [66] used a generalization of Singular Value Decomposition to model the relations between users, resources and tags. Each of such triplets is assigned a probability value. Given a user and resource, the system simply returns the most probable tags related to them. Hence, the recommendation process in an already trained system is very efficient. The idea was extended by Randle et al. [57]. As both methods rely on tensor factorization, the efficiency and scalability of the training process is questionable. Apart from the efficiency problem, the main limitation of graph-based methods is the sparsity of the folksonomy graph. The commonly accepted approach to reduce this problem is graph pruning up to the point where all nodes have at least p edges (p-cores) [3, 35]. The pruning process results in a greatly limited dataset which questions the practical usability of proposed systems. The way to bridge the gap between the need for data pruning and usability was proposed by Krestel et al. [43]. The authors applied the Latent Dirichlet Allocation to the dense core of a folksonomy to extract topics, which are later used to recommend additional tags for infrequently tagged resources.

2.1.2 Content-based Recommendation

Content based recommenders can be divided into three subcategories. The first subcategory utilizes the content of the posted resources as a source of features that are later used in a classification algorithm. One of the first content-based recommenders was presented by Lee and Chun [44]. The system recommends tags retrieved from the content of a blog, using an artificial neural network. The network is trained based on statistical information about word frequencies and lexical information about word semantics extracted from WordNet. Song el al. [64] viewed the tag recommendation task as a multi-label classification problem. A Gaussian process framework is used to create a classifier that trains on the content of web resources (title and short description). Each class corresponds to a topic that is represented as a profile of tags. The tags from different profiles are later combined to create the final recommendation. Unlike graph-based recommenders, these methods are not limited by the uniqueness of resources. However, their practicality is still in question as they rely

on computationally intensive machine learning algorithms. In addition, these methods are only able to re-use the tags that are already present in the tagging system frequently enough, so classification models can be built for them. On the other hand an important advantage of this approach is its generality. Despite the fact that most of the systems extract the feature-sets from the textual content of the resource, the methods are not limited to this form of content specifically. For example, Weston et al. [69] proposed a method of image labelling based on visual features called *visterms*.

Another subcategory of content-based recommenders utilizes information retrieval techniques. TagAssist [65] is a tag recommendation system designed for blog posts. The recommendation is built on tags previously attached to similar resources. The resources are retrieved using a text search engine. In an additional step the co-occurrence of tags is used to unify tags with similar meaning to improve the overall consistency of tag vocabulary. Graham and Caverle [23] uses a text search engine combined with feedback model, analogous to well-known *Rocchio feedback* approach. Tags from the related resources are combined using weighted nearest-neighbour model. The feedback loop allows the user to iteratively improve the quality of retrieved tags. Musto et al. [54] addresses the problem of tag personalization. Again, the basic set of tags is taken from the most relevant resources using the title of the currently posted resource. However, in further steps they explicitly utilize the previous posts of the user, therefore we decided to classify their system also as a hybrid tag recommender. The information retrieval approach allows the systems to gather a diverse vocabulary of tags that can describe the posted resource. On the other hand, these tags are only indirectly related to the resource, hence the precision of such recommendation is questionable. In addition, these methods suffer from the same problem as the previous category of content-based recommenders — the limited vocabulary of tags. To be recommended, the tag should occur in a large number of relevant documents, therefore only frequent tags are likely to be recommended.

The third subcategory of content-based recommenders uses keyword extraction techniques to extract tags directly from the textual content of the resource. Chirita et al. [12] proposed a tag recommendation system that extracts tags from a website content. Aside from basic scores used to estimate the usefulness of a website keyword as a tag (e.g., term frequency), the system matches the website content or its individual keywords against the personal repository of user's documents. Additional

tags are extracted from related personal documents. This way the system is able to get access to additional tags that cannot be found in the website text. These tags represent the personal perspective of a user. Medelyan et al. [51] proposed a tagging system *Maui*, which is purely based on keywords that can be found in the resource content. Maui is an extension of a well-known key-phrase extraction system *Kea* [18]. The system runs a binary classification algorithm for each word or phrase from the resource content. The feature set includes term frequency, distance from the beginning of the document, length of the word and features based on the occurrence of the word in Wikipedia corpus. In general, the keyword extraction methods have direct access to the content of the resource, hence they are more likely to extract precise tags. In addition, they do not rely on tags that were already frequently used in the tagging system. On the other hand, they are limited by the vocabulary of resource, which is likely to be biased by the resource author.

2.1.3 Hybrid Systems

Hybrid tag recommendation systems try to combine the advantages of resource content and folksonomy graphs. As they usually start the processing with the resource content, they are often classified as content-based methods. Graph and content based systems usually tailor a well known machine learning or information retrieval approach to the tag recommendation problem. In comparison, hybrid systems try to utilize specific strengths of several information sources in folksonomies. Such approach allows them to be more efficient and process a wider variety of posts, hence it makes them more practical. Our system follows this tag recommendation approach. Among many proposed hybrid systems we mention three that are most related to our work. Tatu el al. [67] proposed a system based on tags extracted from resource and user profiles. The set of tags is extended using NLP techniques and later merged with content based tags. A system by Ju and Hwang [39] scans the content of previously tagged documents to evaluate the likelihood of a content word being used as a tag. The likelihood is later used as a score for words that occur in the content of currently posted resource. The content based tags are linearly combined with tags from resource and user profiles. Musto et al. [54] based their system on a search engine. The system retrieves resources, which textual content is related to the posted resource title, and

builds the recommendation based on prominent tags from their profiles. Specific attention is given to resources posted previously by the author of the current post — their tags are weighted higher when tags from all relevant resources are combined.

An interesting perspective of hybridization in tag recommendation was presented by Gemmell et al. [21]. Unlike other approaches, their system is based solely on the information extracted from the folksonomy graph and does not use the resource content. The system is based on six simple recommendation models, which include the most frequent tags from resource and user profile as well as four collaborative filtering methods with different ways of calculating the similarity between users and resources. The authors tested the performance of the hybrid, as well as its components, showing interesting differences in tagging behaviour between various datasets. Combination of simple recommenders that exploit specific data dimensions is able to match or outperform state-of-the-art graph-based approaches [59]. It is important to mention that the authors extracted p-cores from each of the datasets to focus on the dense core of the folksonomy graph. P-cores can be extracted for broad folksonomies only and even then, they contain only a small fraction of all posts entered into the system. Therefore, p-cores completely change the character of the tag recommendation problem. The objective of our work was to design a tag recommender capable of producing a recommendation for all posts entered into a system. In our opinion, this is a more realistic problem.

2.1.4 ECML/PKDD Discovery Challenges

An opportunity to compare different tag recommendation approaches was created by two ECML/PKDD Discovery Challenges in 2008 [33] and 2009 [17]. The challenges were organized by the administrators of *BibSonomy* system. The first challenge had a single general tag recommendation track. The competing recommender systems were trained and tested on a complete set of posts that were entered to *BibSonomy* prior the challenge. A subset of posts with the latest time-stamps was separated as a testing set. The top places in the challenge were taken by hybrid tag recommenders [45, 67, 40], including our own system (predecessor of the current system), which finished second. It confirms that combination of tags from different tag sources is a crucial feature of a tag recommendation system. Given the dataset and the evaluation approach another important aspect of successful system design was utilization of resource title as the

main source of tags. Our system followed this concept. Its main drawback was too strong emphasis on the tags extracted from tag profiles of users. We underestimated the noisiness of this source of tags.

In the 2009 challenge, the tag recommendation systems were compared on three tasks. The first task, content-based recommendation, was almost an exact replication of the task from the previous challenge. The second task, graph-based recommendation, used only the posts from the p-core of the folksonomy graph (p = 2). Finally, in the bonus online recommendation task, the systems were connected to *BibSonomy* and their recommendations were used and judged by real users. Our own system [46] submitted to the challenge achieved two first places (content-based and online recommendation) and one third place (graph-based recommendation). The key to its success was simplicity and utilization of the combined advantages of various tag sources: resource title, resource and user profiles as well as the associations between title words and tags. The main drawbacks of the system was an inefficient data structure that represented the folksonomy graph and the need for manual parameter tuning. These problems made the system less practical for large datasets. The system presented in this thesis is an extension of the system submitted to the challenge. Among other improvements it resolves these two problems.

2.2 Motivation of Tagging

The objective of a tag recommendation system is to simplify the tagging process by proposing tags that users would find useful. Hence, before designing a system we should understand the motivation behind user tagging behaviour. We present an overview of generative models of folksonomy data structure, which represent the tagging motivation. In addition, we discuss the potential impact of the textual content on tagging decisions.

2.2.1 Tagging Models

Tagging is a complex process which involves actions of a large community of users. To make it easier to understand we usually view this process as a combination of tagging models. The three most frequently discussed tagging models are the collaborative, personal and shared knowledge model. The *collaborative model* [9, 16, 22, 62]

assumes that, while tagging, a user takes into consideration tags attached to the same resource by other users. This can happen directly when the user adopts a resource from someone else or indirectly when the user assigns tags suggested by a tag recommendation system, which draws the recommended tags from the resource profile. This model is the basis of the folksonomy self-organization assumption. The *personal model* [46, 55, 62] assumes that the user treats the collaborative tagging system as a personal repository of web resources, ignoring its collective character. In this case, the main aim of the user is to re-use personal tags to organize an individual library of resources. The *shared knowledge model* [16, 22, 25] assumes that all users comprehend the content of the tagged resource in a similar way, hence they should come up with a similar set of tags to describe it as they are pulling the tags from a shared repository of descriptions that capture the semantics of the resource. The collaborative and personal models are in obvious contradiction and are quite easy to characterize. On the other hand, the role of shared knowledge model is hard to identify because of the vague nature of the resource semantics and the fact that its effect can be confused with that of the two other models.

The first models of tagging behaviour to explain observed folksonomy characteristics (differences in popularity of tags, stabilization of tag proportions and power-law in resource profiles) were based on generative processes which assumed a common vocabulary of tags from which users draw their decisions [22, 9, 25]. They all assumed collaborative behaviour of users. Recently, Dellschaft and Staab [16] extended the collaboration based generative model considering the impact of shared knowledge vocabulary to match additional folksonomy characteristics (e.g., sub-linear growth of tags). In contrast, Rader and Wash [55] showed that user's tagging decisions are more affected by the need of personal profile organization than the impact of collaborative suggestions. These results were confirmed by the work of Wetzker et al. [71], who suggested that users develop their personal vocabulary and proposed a method to map it to the general folksonomy vocabulary. Evidence that statistical characteristics of folksonomies (e.g., emerging power-law tag distribution in resource profiles) are not caused by collaborative behaviour of users was provided by Bollen and Halpin [7]. Based on the results of a user study they concluded that power-law distributions emerge independently of the availability of collaborative suggestions. Krause et al. [42] showed that folksonomies and so called *logsonomies*, which are data

structures created based on search log data, have similar characteristics. The similarity occurs despite the fact that good tags are not likely to be good query terms and vice-versa [29]. It may suggest that patterns observed for folksonomies can be in fact typical for any kind of tripartite data structure of users, resources and keywords, even if there is no collaboration between users. This leads to the conclusion that a more general model (e.g., shared knowledge model) could be an explanation of folksonomy characteristics.

Currently we observe an increasing amount of evidence that the collaboration between users is not an important factor that impacts tagging decisions [7, 47]. Conversely, most studies suggest a personal character of tags. The characteristics of folksonomies are also likely to be impacted by the shared knowledge model.

2.2.2 Personal Motivation of Users

Experiments on tagging models revealed the personal character of tags. Users tend to reuse their own tags [55] and built personal vocabulary of tags [71]. However, the personalization process can be extended to capture not only the vocabulary but also personal tagging motivation. Körner el al. [41] suggested that users can be classified as *describers*, who use tags to create verbose descriptions that can be later utilized in search and *categorizers*, who use tags to built a personal classification scheme, which can be later browsed. The ability of a tag recommendation system to recognize the type of the user and adjust the character of recommended tags to it is likely to improve system's accuracy [41].

2.2.3 Resource Content as a Source of Tags

A study by Heymann et al. [29] showed that 50% of tags used in a social bookmarking system — *Delicious* could be found in the text of a website they describe. It confirms that the textual content of a resource can be a very rich source of tag recommendations. The same study showed that website titles on their own contained 16% of tags. Title words are also likely to be among the most frequently used tags for a resource. High overlap between tags and resource content (specifically the title) agrees with the shared knowledge model of tagging. In this case, however, the knowledge is shared between the author of the resource and the tagger.

Chapter 3

Characteristics of Collaborative Tagging Data

In this chapter we introduce the six datasets used in the work. In a series of presented experiments we examined the statistical characteristics of tagging system data focusing on the data sparsity problem. The main objective of the chapter is to describe the potential usefulness of various tag sources in the tag recommendation task. We discuss three main tag sources used in the system — resource profile, user profile and resource title. We also discuss the applicability of a commonly used recommendation approach — collaborative filtering. Finally, we examine the impact of the p-cores pruning on the characteristics of the dataset. The information presented in this chapter was used to design the presented hybrid tag recommendation system.

3.1 Datasets

To gain information about the characteristics of collaborative tagging data and later to evaluate the proposed tag recommendation system we used datasets from six collaborative tagging systems, including three broad and three narrow folksonomies. The datasets represent a wide variety of tagging systems in terms of the size, type of posted resources and time-span. Below we present a description of each used dataset.

3.1.1 Broad Folksonomies

BibSonomy dataset. *BibSonomy*[1] is a repository of webpage bookmarks and references to scientific publications. BibSonomy administrators make their dataset available every half a year. The dataset used in our experiments contains all public posts entered into the system before July 2010 as well as the metadata information of the posted resources.

[1] http://www.bibsonomy.org/

CiteULike dataset. *CiteULike*[2] is a repository of references to scientific publications. The full *CiteULike* dataset, available for research purposes, is updated daily. The dataset we used contains posts entered into the system before July 2010. Unfortunately, the *CiteULike* dataset does not contain resource information, including the resource title. To obtain this information we queried the system retrieving BibTeX metadata information for resources tagged with one out of 3000 mid-frequency tags. From this metadata we extracted titles of resources that could be found in 65% of the posts of the original dataset. Only these posts were used in the experiments. During this process no information about real user IDs was accessible.

Delicious dataset. *Delicious*[3] is a popular social bookmarking site. Despite the fact that *Delicious* does not make its dataset publicly available for research purposes, its size and popularity makes it a frequent object of crawling. To evaluate our system we used a combination of two *Delicious* snapshots, the first snapshot contained the full post profiles of over $13,000$ users [5], the other much larger snapshot contained profiles of over $900,000$ users [70]. As the former does not contain post time-stamps and the latter does not contain resource titles, both snapshots had to be merged. Although matching the posts from both datasets was not trivial as the user ID in both datasets was obfuscated, it was feasible thanks to a similar approach to crawling, the highly overlapping time span of posts and the large size of user profiles (Fig. 3.1(c)). To combine the datasets we matched tag-based and resource-based profiles of users from both datasets, which in most cases gave strong one-to-one overlap. In the experiments we used the posts of overlapping users which resulted in a sample of around nine million posts for which all needed information was known. The posts were entered into *Delicious* between September 2003 and April 2007.

3.1.2 Narrow Folksonomies

Stack Overflow dataset. *Stack Overflow*[4] is a "questions and answers" forum for programmers. Stack Overflow administrators make all information gathered in the system publicly available monthly. The dataset we used contains posts entered into the system before August 2010. We used only the "question" posts, which were

[2] http://www.citeulike.org/
[3] http://delicious.com/
[4] http://stackoverflow.com/

tagged by their authors. Each post contains a short description of a programming problem and its title.

BlogSpot dataset contains blog posts from *blogspot.com* domain (owned by *Blogger* service) crawled by *Spinn3r*[5] between August 1st and October 1st, 2008. The dataset was released for the ICWSM 2009 Data Challenge[6].

WordPress dataset. Analogously to *BlogSpot* dataset, *WordPress* dataset contains blog posts from *wordpress.com* domain extracted from the ICWSM 2009 Data Challenge dataset.

3.1.3 Preprocessing

All tags used in the experiments were lowercased. In addition, we cleaned the datasets looking for two types of posts that can bias the evaluation of tag recommendation systems:

Imports — Collaborative tagging systems allow users to import their resources from external repositories (e.g., browser bookmarks) or other collaborative tagging systems. In most cases the posts are given the same automatically created set of tags. It is especially important to remove such posts because they can strongly bias the results of tag recommendation evaluation. We eliminated posts which contained tags that likely marked the imported posts (e.g., "firefoxbookmarks"). In addition, we removed large groups of resources with the same tags, posted by a single user in a short time period and posts of the same user with different tags, if the time difference between two posts was lower than two seconds. This technique is not effective for *Delicious* data because, while importing posts from browser's bookmarks folder, *Delicious* copies the original time-stamp of the bookmark and uses sub-folders names as tags, making these posts hard to distinguish from real posts. The import removal step resulted in significant changes of the dataset characteristics (e.g., reduction of the number of *BibSonomy* posts by 60%).

[5] http://www.spinn3r.com/
[6] http://www.icwsm.org/2009/data/

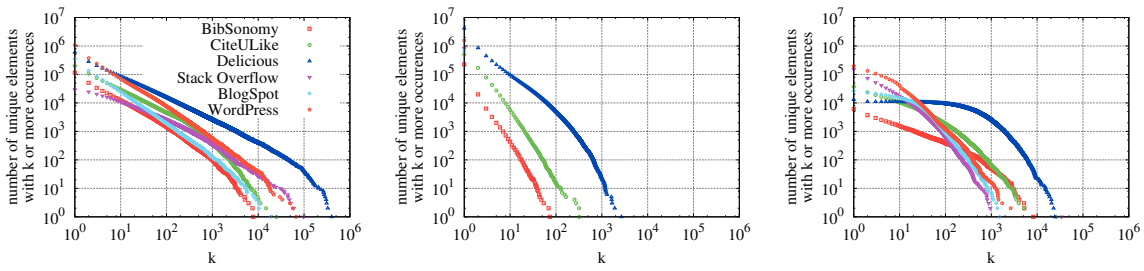
Spam — Tagging systems are frequent targets of spamming as they are well positioned in search engines and allow quick addition of content. Similarly to imported posts, a large number of spam posts of similar pattern could potentially bias the performance of tag recommendation system. We manually browsed all datasets looking for spamming activity, but we did not find any large group of suspicious posts. For the *Delicious* dataset it is likely due to the fact that one of the source datasets was crawled based on *fan* links between users, who were not likely to link to spammers. All the companies and services, that released the other datasets, use some measures of spam prevention.

3.2 Statistical Characteristics

The basic statistical information about the number of posts, tags, resources and users in all datasets is presented in Table 3.1. To gain more detailed information about the statistical properties of tags, resources and users, we ploted the complementary cumulative frequency distribution for unique elements (Fig. 3.1). In the plots, each point corresponds to the total number of unique elements that can be found in the dataset more that k times. This is a commonly used approach to present a well-known characteristic of socially created data structures — a heavy-tailed distribution with a small number of very frequent elements (in the bottom right corner of the plot). To present the difference between datasets we decided to plot the frequency distribution instead of the commonly used likelihood of occurrence. The distribution for unique elements does not directly represent the importance of elements with k occurrences on the recommendation process. For example it is hard to determine if a large number of unique low frequency tags suggests that the system should focus on these tags only. Although the number of unique high frequency tags is much lower, they are frequently re-used in posts, hence it is possible that they will constitute the majority of tags that are chosen by users. To get direct access to this information we plot the cumulative distribution function of a random variable, which is defined as the random draw of an element with overall frequency equal to or less than k from the set of all occurrences of the elements of given type in posts (Fig. 3.2). The distribution illustrates how likely it is to find a resource or a user with the overall frequency equal to k or less in a randomly chosen post. As each post can have more than one tag, for tags the distribution illustrates how likely it is to find such tag among all tagging assignments

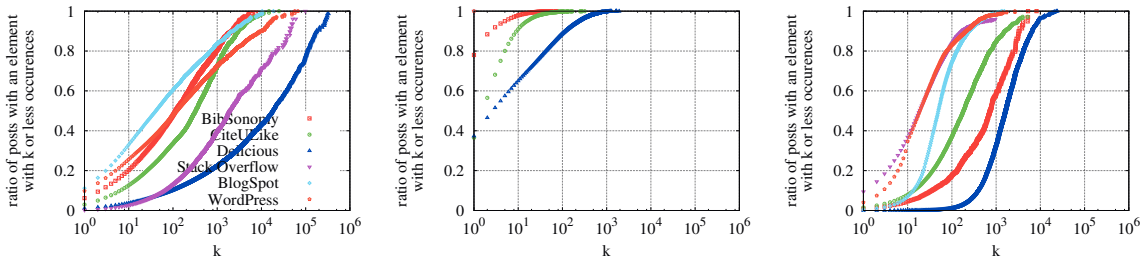
Table 3.1: Statistical information about the six datasets used in the experiments. *Top freq* is the number of occurrences of the most frequent element.

	posts	**tags**			**resources**		**users**	
	total	total	distinct	top freq	distinct	top freq	distinct	top freq
BibSonomy	262,856	1,035,230	114,963	7,739	225,011	74	6,073	8,219
CiteULike	911,383	3,158,224	200,691	24,929	501,874	323	37,388	25,934
Delicious	8,890,876	29,807,506	601,547	399,927	4,172,960	2,673	13,079	24,176
Stack Overflow	833,510	2,490,489	29,240	97,885	833,510	1	151,766	948
BlogSpot	667,052	1,898,283	348,408	18,619	667,052	1	26,379	1,528
WordPress	1,575,704	7,108,380	1,074,567	67,039	1,575,704	1	197,737	8,023



(a) Tags (b) Resources (c) Users

Figure 3.1: Complementary cumulative frequency distribution for unique elements. The distributions for a specific element in all datasets are plotted together.



(a) Tags (b) Resources (c) Users

Figure 3.2: Cumulative distribution function of a random variable defined as the occurrence of a element with overall frequency equal to or less than k in a post. The distributions for a specific element in all datasets are plotted together.

done by users. We used the two presented distributions to discuss the characteristics of three post elements (i.e., tags, resources and users) from the perspective of tag recommendation.

3.2.1 Statistical Properties of Tags

The complementary cumulative frequency distribution for unique tags (Fig. 3.1(a)) shows the difference in the number of low frequency tags among various datasets; stable behaviour of mid-frequency tags which follow the power-law; and expected frequency decay among high frequency tags [8]. Unlike words from a textual corpora, tags do not follow the Zipf's law strictly. Zipf's law is an empirical law formulated for textual corpora [50]. According to the law, the frequency of a word is inversely proportional to its rank in a list of words sorted by the overall frequency of occurrence. As a result the rank-frequency plot or cumulative frequency distribution forms a straight line in the log-log scale plot. For most collaborative tagging systems, the number of low frequency tags is higher than expected. This behaviour confirms previous observations of tag characteristics [9, 8]. It was attributed to the co-existence of a number of low frequency tags, which can characterize the resource [9] or the hierarchical structure of tags [8]. Our personal experience suggests that low frequency tags contain an extensive amount of words or multi-word phrases that are specific to the user and proper names that are specific to the resource. The latter is especially noticeable for blog datasets which contain a large number of unique tags (Table 3.1). As a result, it is unlikely that these tags will be re-used in the future. It is interesting to notice the different behaviour of low frequency tags for *Stack Overflow* dataset. In this case the number of tags is much lower than expected. In our opinion, the users of *Stack Overflow* system put more effort into making their posts more descriptive for others, hence avoid specific tags. The inconsistency with Zipf's law can be observed for high frequency tags as well. For some datasets, these tags are used much less frequently than expected. This characteristic was attributed to the existence of sets of semantically equivalent tags that are used interchangeably [8]. However, in our opinion, it is caused by a low usefulness of a tag that is used too often. For example, users do not use tag *website* for each resource in *Delicious*, because it is clear that resources gathered there are websites.

The presented observations suggest that the tag recommendation approach should

be mostly focused on the low frequency tags. It is not confirmed by the cumulative distribution of tags in posts (Fig. 3.2(a)). For most datasets, majority of tags used by users have overall frequency higher than 100. On the other hand, there is a non-negligible fraction of unique or low frequency tags (especially for blog datasets) that should be taken into consideration during the recommendation process.

3.2.2 Statistical Properties of Resources

As in narrow folksonomies each resource is used only once the distribution of unique resource frequency can be plotted for broad folksonomies only (Fig. 3.1(b)). However, on the contrary to the common assumption, even for broad folksonomies the resources are rarely re-used by different users (steep decrease in the plots). Among nearly nine million posts in *Delicious* dataset only $2,673$ contained the most frequent resource (0.03%). Other datasets have similar properties. As a result, the usefulness of tags used previously for the same resource is questionable.

The same conclusions can be drawn observing the distribution of resources in posts (Fig. 3.2(b)). However, the ratio of posts, which contain frequently used resource differs between datasets. The resources with ten or more occurrences can be found in over 30% of *Delicious* posts. For *BibSonomy* the fraction of such posts is negligible.

3.2.3 Statistical Properties of Users

The complementary cumulative distribution of the unique user frequencies suggests the rich-get-richer behaviour (Fig. 3.1(c)). We can observe a small number of users with a very large number of posts and a large number of users with small number of posts. The effect of crawling can be noticed in user distribution for the *Delicious* dataset, where we only have the information about the tail of the distribution (users with a large number of posts).

The cumulative distribution of users among posts shows that in all cases most of the posts are created by mid-frequency users (Fig. 3.2(c)). For *Stack Overflow* and *WordPress* datasets over 60% of posts are posted by a user with overall number of 10 or more posts. For other datasets this ratio is much higher. Therefore, in most cases, there is enough information to build a user profile and recommend user related tags. In addition, for broad folksonomies datasets the majority of posts are entered by users with 100 and more posts. In such case we can assume that the system has

extensive knowledge about the user interests and tagging patterns.

3.3 Importance of the Resource Title in the Formulation of Resource and User Profiles

So far, we have examined the statistical properties of elements that can be found in the folksonomy graph. Another potential source of information useful in the recommendation process is the content of the resource. In our work, we decided to focus on a specific element of the content, namely the resource title. There are three main reasons for this choice. First, title seems to be the only form of textual element that is present in all types of resources, from web-sites stored in Delicious to songs stored in Last.fm. Second, title seems to serve a similar purpose as tags, which is to be a short and concise description of the resource. Finally, thanks to shortness and simplicity title does not require costly preprocessing steps.

To examine the relation between title words and tags we ran two experiments. In the first experiment we looked for a statistical confirmation whether the occurrence of the term in the title is related to its use as a tag. In the second experiment we observed the overlap between the title words and the most frequent tags from the profile of the resource. For this reason we had to limit the datasets for which we present the results of the experiments to two broad folksonomies — CiteULike and Delicious. In addition, we worked on an older version of the CiteULike dataset with $200,291$ posts. Narrow folksonomies could not be used in the experiment because each resource is tagged only once in them. Although BibSonomy is a broad folksonomy, it contains a small number of resources with rich tag profiles, therefore we were not able to extract a sufficient number of samples for the experiments.

To check if the occurrence of a term as a tag is related to its occurrence as a title word, we examined terms that were used at least 100 times as a tag or could be found in a resource title of at least 100 posts ($36,558$ terms for *Delicious* dataset and $2,155$ terms for *CiteULike* dataset). This threshold was chosen to remove the potential noise caused by low frequency terms. For each term we checked in how many posts the term can be found (a) as a tag, but not in the title, (b) in the title but not as a tag and (c) both as a tag and in the title. We extracted terms for which the number of posts in each of the three sets was at least five ($17,821$ terms for *Delicious* dataset and $1,532$ terms for *CiteULike* dataset). We ran the *Pearson's chi-square test of*

independence for each of these terms. In each case the *null* hypothesis (independence of tags and title words) was rejected with high confidence $p < 0.0001$. Hence, for these terms we are able to confirm that use of a term as a tag is related to its occurrence in the title.

We manually browsed the list of terms that were rejected from the experiment because of an insufficient number of samples. We focused on the terms, which, despite being popular as tags, could not be found in the title. We found that a significant part of these tags (30% for *Delicious* dataset and 54% for *CiteULike* dataset) matched `(w+W)+w` regular expression pattern, where `w` stands for a letter and `W` stands for a non-letter character used to separate words. These tags are complex terms composed of two or more words (e.g., "social_networks"). In this case it is likely that the relation between the title and tags exists as well, but is too complex to be captured by our experiment.

To get quantitative information about the overlap of title words and tags, we processed all posts in both datasets counting the number of times a tag can be found in the title of tagged resource. The experiment shows that 15% of tags in *Delicious* dataset and 26% of tags in *CiteULike* dataset can be found in the title. The outcome for the *Delicious* dataset agrees with the results obtained by Heymann et al. [29]. The large difference between the datasets is likely to be caused by the character of the resources. The title of a web page is usually shorter and less descriptive than the title of a scientific publication, hence the former is likely to provide fewer terms that can be useful as tags.

The potential importance of the title in the formulation of resource profile was revealed by the second experiment in which we took the profiles of frequently posted resources and calculated the likelihood of a title word being highly ranked in the profile (number of times the tag with rank k was found as the title word, divided by the number of tested resources). We set the threshold of accepting the resource as frequently tagged at 100 for *Delicious* dataset and 20 for *CiteULike* dataset. The choice of the threshold value followed the work by Heymann et al. [30], who showed that the list of the top 100 tags in the resource profile originates mainly in the first 100 posts. Unfortunately because of a low number of frequently posted resources we had to lower the threshold for *CiteULike* dataset. To reduce the bias caused by the variance in the number of posts per resource, we decided to use only the first 100

(or 20) posts to build the resource profile. For 40% of the tested resources for the *Delicious* dataset (50% for the *CiteULike* dataset) the top ranked tag in the profile was found in the title (Fig. 3.3). The probability of having a tag-title co-occurence rapidly decreases with the rank of the tag in the profile, which shows that title contains few high quality words that are used as tags frequently. On the other hand, the cumulative ratio of title words being used as top k tags is constantly growing with the increasing value of k, even for high k. Possibly these words are not good descriptors of the resource and they were used as tags only because they were noticed in the title. On average 60% of title words can be found among the top 100 or 40 tags of resource profiles for *Delicious* and *CiteULike* dataset respectively (Fig. 3.3).

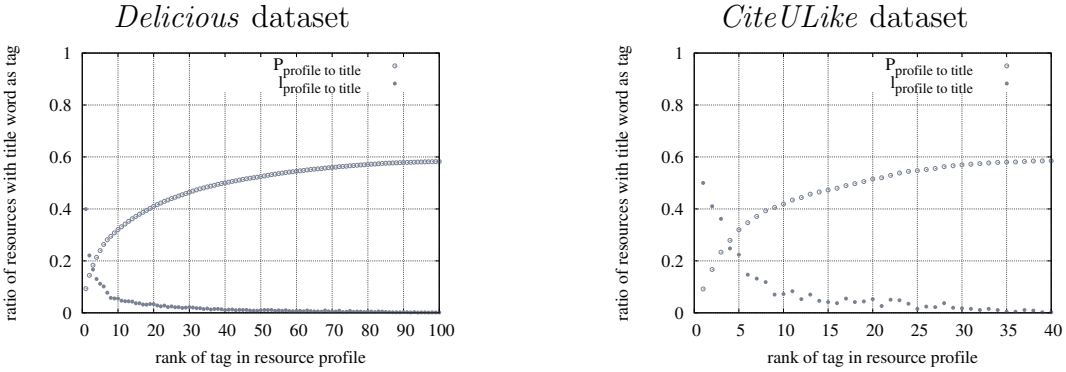


Figure 3.3: The overlap of title and resource profile: $l_{profile\ to\ title}$ — what percentage of profile tags with rank k can be found in the title, $P_{profile\ to\ title}$ — what percentage of top k profile tags can be found among title words. Title words are frequently used as the most significant tags in resource profiles.

The results of the experiments confirm that title words are frequently used as tags and often these tags are popular among users tagging the resource. These observations make the title a valuable source of recommended tags. At the same time, we revealed a large group of complex tags, built out of two or more terms, which are potentially related to the title. A potential way to access these tags are term co-occurrence patterns, which match frequently co-occurring title words and tags. This idea is discussed in details in Section 4.1.

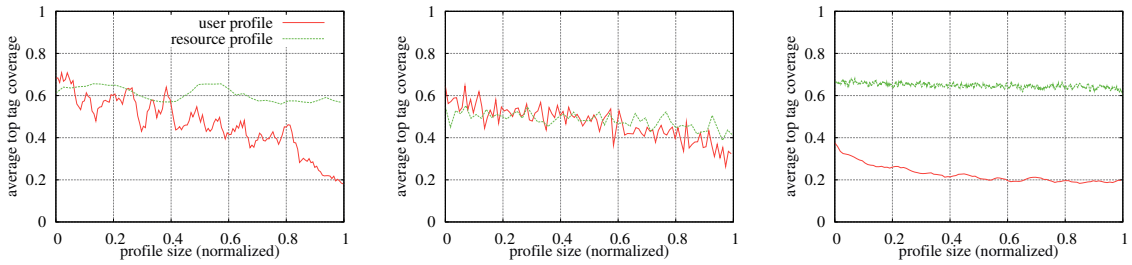
3.4 Coherence of Tag Profiles

The statistical characteristics of broad folksonomies confirmed that in the recommendation process we can assume rich information about the previous tagging decisions

of users. The same cannot be said about the resources, which with the exception of Delicious dataset are sparsely tagged. The tags previously used by users or assigned to resources can be represented by tag profiles. To determine the usefulness of tag profiles in the tag recommendation process we investigated the coverage of the most popular tag in the profile, defined as the ratio of posts of a user (or resource) with a tag. High coverage would suggest frequent re-use of tags making the most frequent tags from the profile a good recommendation. Specifically, we wanted to observe how the number of posts used to build the resource impacts the coverage. To achieve it, we sorted the users based on the number of their posts. For each user we built their tag profile and calculated the ratio of posts with the most frequently used tag (*top tag*). Only users with ten or more posts were used in the experiment. Later we used a sliding window to calculate the average top tag coverage of 500 users over the range of users sorted by the number of their posts, starting with the least active users. An analogous procedure was ran for the resource profiles.

For all datasets, while comparing the results for the least and most active users we can observe that the latter have much lower coverage (Fig. 3.4). It suggests that as more posts are entered by a user, user's profile becomes noisier. This fact decreases the usefulness of user's tag profile in tag recommendation. One of the potential explanation of this fact is the dynamic character of user interests. As shown by Wetzker et al. [71], users tend to use certain tags very actively for some time and then abandon them. To address this characteristic a tag recommendation system based on user's tag profiles should take the temporal character of tag use into consideration. On the contrary, the coverage of resource profiles remains relatively constant independently of the number of posts used to build the profile. This result was expected given previous research which revealed the stability of tag profiles of resources [22, 25]. Comparing the average coverage of user and resource profiles we can notice that the latter are likely to be more accurate sources of tag recommendations. For BibSonomy and Delicious dataset the average top tag coverage for resource profiles exceeded 60% comparing to 46% and 24% respectively. For CiteULike dataset the difference was much lower (48% comparing to 47%). Considering the statistical properties, we can conclude that although resource profiles are much sparser than user profiles, once a sufficient number of posts is gathered in the system they are likely to be a more accurate source of tag recommendations. High coherence of resource profiles suggests

the possible influence of collaborative behaviour, as a large group of users tend to agree on the most frequently used tag. However, if the impact of a collaborative model was significant we should observe the increasing coverage of the most frequent tags with the number of posts. Following the rich-get-richer principle, once a tag becomes popular it should be re-used more frequently by other users. On the other hand, the coherence can be also an effect of the similar perception of the resource by different users which is explained by the shared knowledge model. This hypothesis is confirmed by the previous experiment in which we revealed high overlap between title words and tags, which suggests the shared knowledge between the author of the resource and the tagger.



(a) BibSonomy (b) CiteULike (c) Delicious

Figure 3.4: Average coverage of the most frequent tag in profile. Tag profile of users become noisier with the increasing number of posts used to build them.

3.5 Experiments on Synonymous Tags

The previous experiments allowed us to determine the three main sources of tags: the tag profile of the resource, the tag profile of the user and the resource title. To gain more information about the characteristics of these sources and their usefulness in the recommendation process we decided to focus on a specific subset of tags. We manually selected a set of pairs of terms, which can be used completely interchangeably to tag a resource. For simplicity we refer to them as synonymous tags, however, two synonymous tags do not have to be synonyms in natural language, which is the case in our study. Given a pair of synonymous tags we could observe the context of using them as tags to determine the sources of information or procedures that impact the choice between them. Again, as we needed the information about three sources of tags we decided to limit the experiments to CiteULike and Delicious datasets.

The pair of synonymous tags, that we decided to focus our attention on, is a singular and plural form of the same noun. The fact, that these two forms used as tags convey the same meaning, was pointed out in previous work [10, 71], here we discuss the problem in more details. Most of the tags used in folksonomies are nouns, which is natural given that the aim of the tagging process is categorization of resources [25]. To categorize the resource, the noun can be used in singular or plural form to indicate that the resource (e.g., *blog*) belongs to a given category (e.g., *blogs*). We examined the list of the one thousand most frequent tags to find the popular (singular, plural) pairs of tags. It resulted in 96 pairs for the *Delicious* dataset and 51 pairs for the *CiteULike* dataset. These pairs were used in all the following experiments. We present the list of top ten pairs, sorted by the frequency of the more frequent form, for each dataset (Table 3.2). To confirm that two forms of the same term convey the same meaning when used as a tag, we looked at the resources for which one of the two forms was used at least 10 times. We then compared the sets of resources associated with the two forms of the same tag. The *Jaccard similarity coefficient*[7] [34] averaged over all pairs is 0.83 for the *Delicious* dataset and 0.76 for the *CiteULike* dataset. The large number of (singular, plural) pairs among the most frequent tags and the high overlap between the resources described by the two forms of the same term agree with the intuition that such pairs can be viewed as functional synonyms. Focusing on the pairs of singular and plural forms of the same noun has an additional advantage in our study. Although they can be used interchangeably as tags it is not the case in natural language. Often the form of a term is determined by the longer phrase in which it is used (see Table 3.3 for examples). The situation in which a concept can be represented by both forms of a term, but the form that is used as a tag follows the form found in the title would be clear evidence that tags are influenced by the resource title. We used this idea in the following experiments, observing resource and user profiles in which one or two forms of a synonymous tag pair could be found.

[7] Jaccard similarity coefficient is defined for two sets as the size of their intersection divided by the size of their union

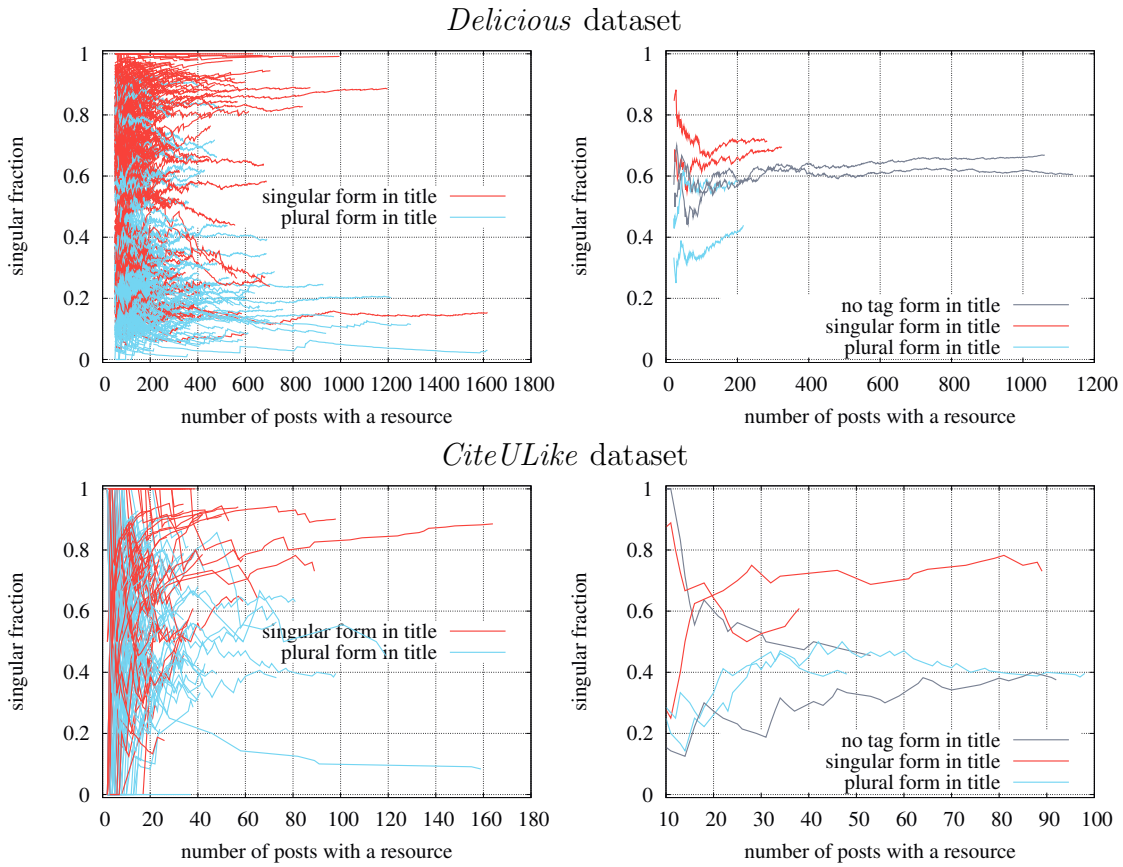
Table 3.2: Top ten pairs out of the list of synonymous tags pairs used in the study. Frequency and rank are calculated based on tags distribution (Fig. 3.1(a)). The terms are sorted by the frequency of the more frequent form. The pairs of a singular and plural form of a tag are frequent in collaborative tagging systems. There is no general rule for the more popular form of a tag (the position of the more popular form is presented in boldface).

	singular		plural	
	frequency	(rank)	frequency	(rank)
Delicious dataset				
blog(s)	303973	(**4**)	146377	(19)
tool(s)	54152	(84)	266422	(**8**)
art(s)	237611	(**9**)	4497	(781)
video(s)	206044	(**11**)	17295	(258)
tutorial(s)	128669	(**24**)	52251	(88)
tip(s)	4052	(853)	106433	(**36**)
book(s)	43546	(106)	103169	(**38**)
game(s)	38411	(120)	102812	(**39**)
article(s)	80000	(**55**)	36120	(127)
wiki(s)	69394	(**66**)	5069	(694)
CiteULike dataset				
review(s)	27579	(**1**)	998	(910)
human(s)	13215	(10)	22420	(**2**)
animal(s)	3333	(173)	15952	(**4**)
model(s)	13563	(**7**)	11716	(16)
protein(s)	13300	(**9**)	7079	(50)
network(s)	12737	(**11**)	10105	(21)
method(s)	4828	(99)	9343	(**28**)
gene(s)	8402	(**35**)	3033	(198)
genetic(s)	6575	(53)	7629	(**44**)
cell(s)	7322	(**48**)	4381	(115)

3.5.1 The Impact of Resource Title on Resource Profile

Knowing that words from the title are likely to be frequently used in the profile of a resource (Section 3.3), we decided to trace post streams of resources, to observe how the use of selected tags changes in time. A post stream [16] is a sequence of posts ordered by time-stamps. In our experiment we limited the stream to posts with a specific resource. We selected resources for which one of the two forms of (singular, plural) pair was frequently used as a tag (threshold of 20 or 5 uses for *Delicious* and *CiteULike* dataset respectively). Each frequently tagged resource for each tested pair

was traced separately. Whenever one of the two tag forms was used, the fraction of singular tags among both singular and plural tags (*singular fraction* or sf) was recorded. If the occurrence of the word in the title has a direct impact on the choice of a tag we should observe it in the value of the *singular fraction*. The presence of the singular form of the tag should make the fraction high, whereas the presence of plural form should make it low.



(a) post streams of all resources traced in the experiment (resources with no tag form in title omitted for clarity)

(b) resources traced for tag *blog* (*Delicious*) and tag *network* (*CiteULike*) — two most frequent of each kind

Figure 3.5: Results for resource profile tracing. The average singular fraction calculated for full profiles of resources with the singular form of the term in the title is higher than the average fraction calculated for resources with plural form in the title.

To present the results (Fig. 3.5(a)), we adapted the visualization method used in [22] and [9]. The *singular fraction* for a cumulated profile of each resource is presented as a single trace as a function of time, measured by the number of posts associated with the resource. The colour coding shows that the form of the title

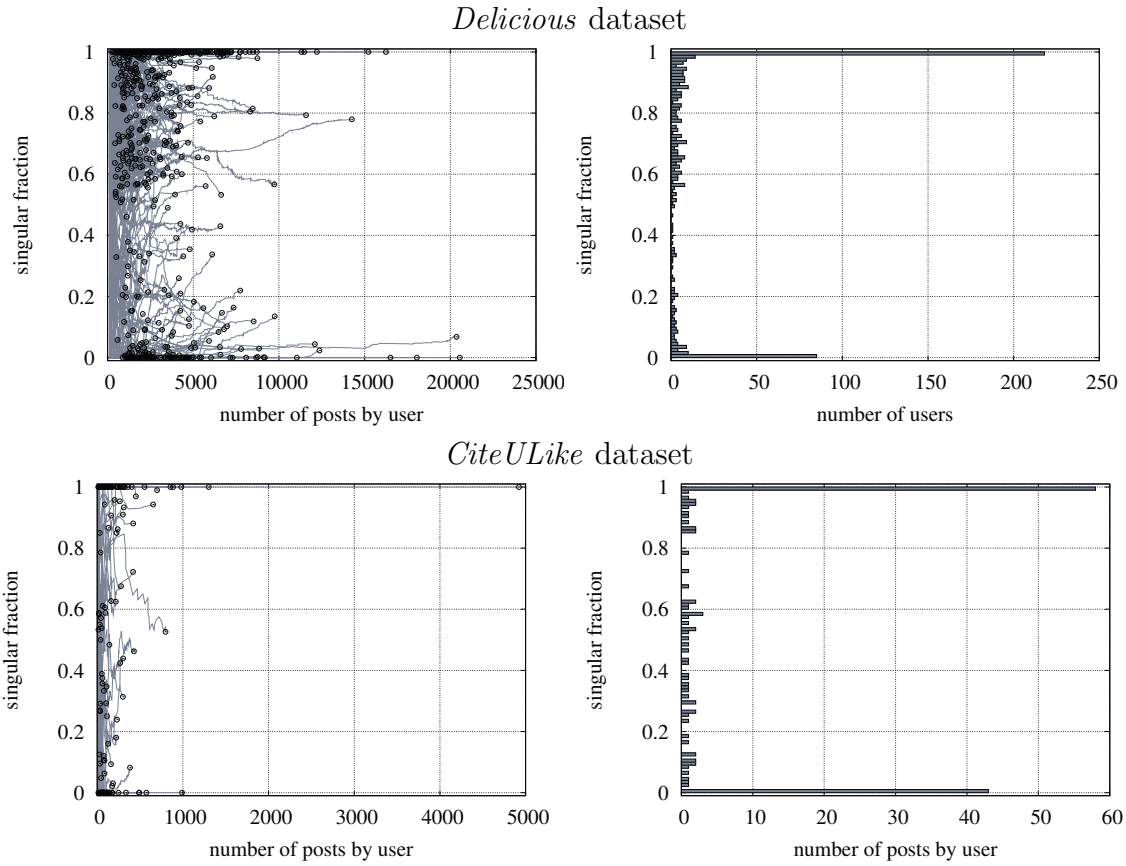
word is correlated with the dominant form of the tag in most cases. For most tested pairs (93% for *Delicious* dataset and 94% for *CiteULike* dataset) the average *singular fraction* calculated for full profiles of resources with the singular form of the term in the title is higher than the average fraction calculated for resources with plural form in the title. The form of the term as title word boosts its frequency as a tag. To confirm that the title has the determinant role in impacting the choice of the tag form we would expect that among tags of a synonumous pair the majority of tags has the same form as the word in the title ($sf_{plural} < 0.5 < sf_{singular}$). Such clear division between *singular fraction* value for the resources with singular/plural form of the same tag in the title was observed for a small fraction of pairs only (25%, e.g., *download(s)*, for *Delicious* dataset and 22%, e.g., *network(s)* for *CiteULike* dataset). For these pairs the dominant form of the tag in the resource profile depends on the form that can be found in the resource title, even if both forms convey the same meaning (Fig. 3.5(b) bottom, and Table 3.3). The other pairs are strongly biased towards one of the forms (e.g., *blog* is the dominant form in *blog(s)* pair for *Delicious* dataset), and, even though the occurrence of the less frequent form in the title influences the choice of a tag, it is often used in a minority of posts for a given resource (Fig. 3.5(b) top, and Table 3.3). Although the title is a factor that impacts the choice of the tag form, in most cases, its impact is not strong enough to overcome the popularity bias caused by some other factors.

Table 3.3: The ratio of singular form of a tag for example resources related to the concept of blogging (*Delicious*) and complex networks (*CiteULike*). The form of term found in the title boosts its frequency as a tag.

resource title	sf
Delicious dataset	
Blog Software Breakdown	0.68
(...) Create your **Blog** Now — FREE	0.68
Blog software comparison chart	0.67
(...) Where **Blogs** Meet Maps	0.58
CiteULike dataset	
Folksonomy as a complex **network**	0.73
Exploring complex **networks**	0.39
Complex **networks**: Structure and dynamics	0.39
Statistical mechanics of complex **networks**	0.38

3.5.2 Synonymous Tags in User Profiles

To observe the personal use of synonymous tags, we ran the previous experiment focusing on profiles of users, not resources. We picked users who used one or both forms of a tag frequently (at least 50 times for *Delicious* dataset and 10 times for *CiteULike* dataset). This time we were not able to classify the post stream traces based on the occurrence of one of the forms of the term in resource title, because users tag various resources. However, even neglecting the occurrence of the term in the title, the traces of user profiles lead to interesting observations. Most of the users pick a single form of a tag and use it consistently every time they tag a resource related to the concept represented by this tag. As most of the user profile traces have extreme values of *singular fraction* they overlap on the trace plot (Fig. 3.6(a)). To make this fact clear we present a histogram of final values of *singular fraction* for each user post stream (Fig. 3.6(b)). The histograms for two example pairs of tags (*blog(s)* and *network(s)*) show that the majority of users use the form of a tag, which generally is more popular, this suggests the impact of the shared knowledge model. For some tags one of the forms can simply "sound better" for the majority of users. Nevertheless, a large group of users uses the other form only. Observing the profiles of users we found that they tend to keep one of the forms (singular or plural) for all the tags they use. It can be considered as a confirmation of a hypothesis that users tend to limit the number of tags and tag forms in their profiles, which suggests a strong impact of the personal model. This behaviour is one of the factors that keep the constant inflow of both forms of a tag to the resource profile. Therefore, it could act as an explanation for the results observed in the previous experiment. However, at the same time can be considered as its contradiction. Most of the users are likely to completely disregard any external influence, including the title, as they have already decided on the tag that is going to represent a concept throughout their posts. So where does the impact of the title visible in the resource profiles come from? It is important to notice that the experiment on the user profiles illuminated the behaviour of a specific group of users, who used the interesting tag frequently. Such frequent tags could be of special interest to the users as defining their general area of interests. We could imagine another group of users who used the same tag infrequently. For them the tag is most likely just an additional tag which only specifies the description of the resource.



(a) *singular fraction* for frequent users of tags blog(s) (*Delicious*) and tags network(s) (*CiteULike*), circle marks the fraction for full profile

(b) *singular fraction* value histogram for full profiles of frequent users

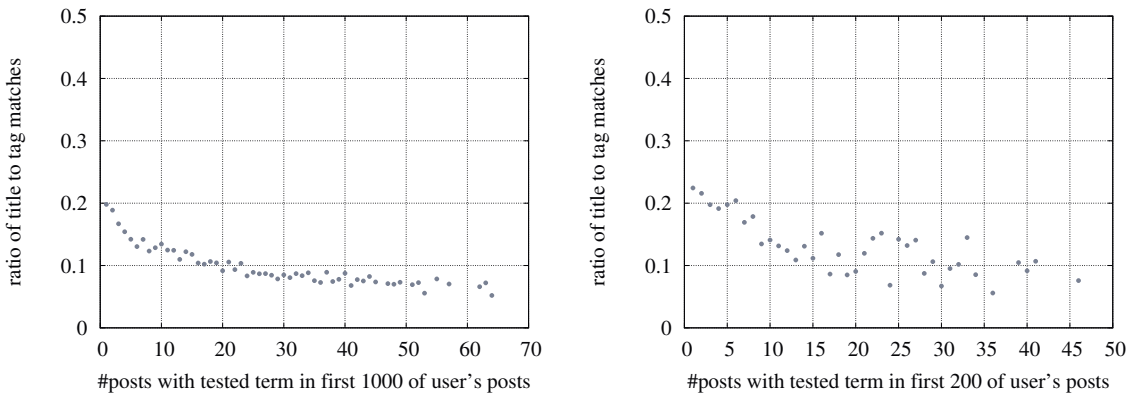
Figure 3.6: Results for user profile tracing. Users tend to constantly use a single form of a tag.

Our hypothesis was that drawing a tag from the title is more likely for "infrequent" users than "frequent" users. To confirm this hypothesis we picked a set of users with large profiles ($p_{user} > N$, where $N = 1000$ for *Delicious* dataset and $N = 200$ for *CiteULike* dataset) who used at least one tag from the list of synonymous tags (192 tags for *Delicious* dataset and 102 tags for *CiteULike* dataset). To eliminate the bias caused by different sizes of user profiles we limited them to the first N posts entered by the user. The users were chosen separately for each of the traced tags. For each user/tag pair we recorded how many times k the tag was used among the first N posts of the user. Later, for each tag and each value of k we checked whether the use of the tag co-occurred with the occurrence of the term as title word. This allowed

us to calculate the ratio of title to tag matches — the number of times the tag was used and it appeared in the resource title — to the total number of times the tag was used. To avoid the need for arbitrary choice of the threshold of k that would separate "infrequent" and "frequent" users we aggregated the results for each value of k separately. Despite the high variability of results for high k, some correlation between the frequency of tag use k and co-occurrence of title words and tags can be observed. The *Pearson's correlation coefficient* between k and the ratio of title to tag matches is equal to -0.21 for *Delicious* dataset and -0.25 for *CiteULike* dataset. High variability of results, which affected the value of correlation coefficient, was caused by problems with finding a representative set of users who used the tag a specific number of times k, when k is high. In most cases, for high k the ratio of title to tag matches could be calculated based on the information from a single user/tag pair which makes the results noisy. To reduce the noise, we combined the results for all tested tags and discarded results for k if the number of users, for which we recorded the data, was lower than 10. Despite the fact that this procedure limited the maximal value of k, for which we had any information, it reduced the noise and revealed the pattern of decreasing ratio of title to tag matches with growing k (Fig. 3.5.2). The probability of a tag being drawn from the title by the user decreases with the number of times the tag was used by this user. Hence, when choosing a tag, "infrequent" users are more likely to be influenced by the title than "frequent" users. In the tag recommendation setting the results suggest that both sources of tags: the title and the user profile can complement each other.

3.6 Tag-based and Resource-based Similarity of Users

Looking for other potential sources of tags we turned our attention towards collaborative filtering, the most commonly used recommendation approach. Collaborative filtering is based on the assumption that similar users are likely to make similar choices. The concept is clear in the standard recommendation setting in which each user can be represented as a vector of items. Given the user vectors we are able to calculate the similarity between each pair of users and recommend the items from users who are similar to the given user. In the tag recommendation setting the similarity between users can be calculated in two ways based on the resources they tag and tags they use. It increases the complexity of the problem because we look for matching



(a) *CiteULike* dataset

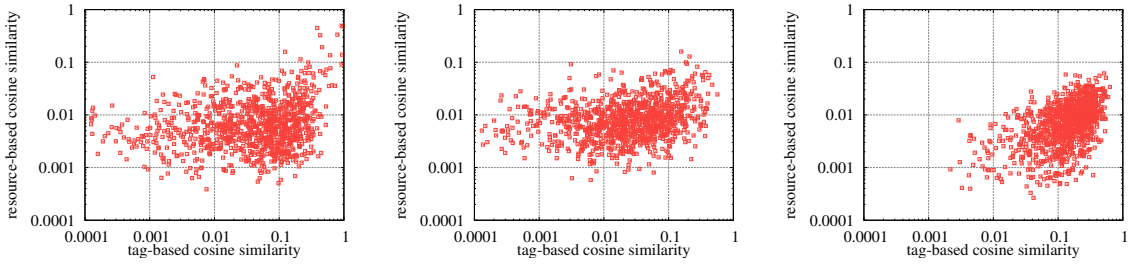
(b) *Delicious* dataset

Figure 3.7: The percentage for tag-title matches in relation to the number of occurrences of a tag in user profile. "Ratio of title to tag matches" is the number of times the tag was used and it appeared in the resource title to the total number of times the tag was used. Infrequently used tags are more likely to be drawn from the resource title.

users that share both the interests (represented as common resources) and tagging style (represented by common tags). To test the potential usefulness of collaborative filtering approaches in tag recommendation we observed correlation between the pairwise user similarity calculated based on tags and resources. If the correlation between two scores is high we can conclude that users interested in similar resources tag them in similar way. Therefore while recommending tags for a user, resource pair we should take into considerations the tags assigned to the resource by similar users. Conversely, low correlation would suggest that resource of interests and the tag vocabulary used to describe them are independent for each user. In such case, mining the tags assigned to the resource by similar users would give similar results as recommending the most frequent tags from the resource profile. To observe the correlation between the two similarity types we calculated two cosine similarity scores, first based on the tag profiles of two users, second based on users' resources, for each pair of active users. In the experiment we set up a threshold on the minimal number of posts by a user to select roughly 3000 of most active users from each of the broad folksonomies. We used the following thresholds: 10 posts for BibSonomy, 50 posts for CiteULike and 1000 posts for Delicious.

Before discussing the correlation results we have to once more mention the sparsity

problem. Among all pairs of users used in the experiment only 2% of pairs for BibSonomy dataset and 4% of pairs for CiteULike dataset had non-zero similarity scores both for tags and resources. This ratio was much higher for Delicious — 95%, which is likely to be caused by much larger profiles of users. Even if there is an overlap between tag profiles of users or the set of their resources the cosine similarity score is usually very low (Fig. 3.8), which suggests that the overlap is just coincidential. These issues limit the practical usefulness of collaborative filtering approach. On top of that, the cloud plots of two similarity measures demonstrate low correlation between both scores for BibSonomy and CiteULike, some correlation can be observed for Delicious data. These results are confirmed by Pearson correlation coefficient which is 0.33 for BibSonomy, 0.39 for CiteULike and 0.51 for Delicious. The results of the experiment show that both low overlap between user profiles and low correlation between similarity scores calculated for tags and resources question the usefulness of collaborative filtering approaches for tag recommendation. In this situation recommending tags assigned to a resource by similar users (collaborative filtering) should give similar results as recommending the tags frequently attached to the resource by any user. This conclusion seems to be confirmed by the experiment presented by Jäschke et al. [35], who compared the results of collaborative filtering approach with the tags most frequently assigned to a resource. We look futher into this issue in the comparative evaluation of the system (Section 5.2).



(a) BibSonomy (b) CiteULike (c) Delicious

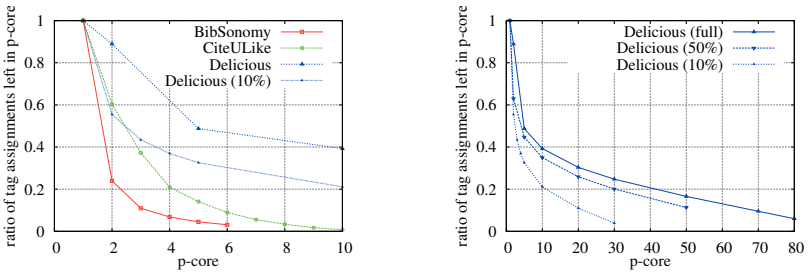
Figure 3.8: Correlation between tag-based and resource-based similarity for user profiles. For each dataset we randomly picked 1000 pairs of users with non-zero similarity scores. The cloud plot demonstrates low correlation between both scores for BibSonomy and CiteULike, some correlation can be observed for Delicious data.

3.7 P-cores Pruning

Some recommendation techniques, including approaches based on collaborative filtering, are unable to deal with the infrequently occurring elements. A commonly adapted approach to the evaluation of these algorithms is p-cores pruning [21, 35, 59]. P-cores extraction is an iterative process commonly used to extract densely connected components from graphs [3]. Tagging datasets can be represented as tripartite hyper-graphs in which the edge is replaced by a *tag assignment*, defined as a triple that consist of a single user, resource and tag. In this setting, in each iteration, p-cores extraction removes tag assignments, which contain user, resource or tag that overall occur less than p times. The removal of any tag assignment can cause that other tag assignments would contain elements that overall occur less than p times, which creates a need for another iteration. The process ends when in a full iteration no tag assignments were removed. The result of p-cores pruning is a dataset in which all posts contain resources, users and tags that occur at least p times in the dataset. As in narrow folksonomies each post contains a unique resource, p-cores pruning can be applied to broad folksonomies only. In our work, we were interested in the practical impact of the p-cores applied to tagging data, mainly the number of tag assignments that is removed in the pruning process. To observe that, we applied the p-cores extraction algorithm for various values of p to the three broad folksonomies.

As expected the maximal value of p for which we can generate a non-empty p-cores set depends on the size of the dataset. For the smallest dataset (BibSonomy) the algorithm returns no tag assignments for $p = 7$. In general, the maximal value of p is rather low considering the size of the datasets. Comparing the ratio of tag assignments left in the dataset after the p-cores pruning we can notice that for BibSonomy dataset almost 80% of them are removed in the first step for $p = 2$. As a result, even the smallest threshold of p-cores pruning can completely change the characteristics of the dataset. For CiteULike dataset this level is reached for $p = 4$ which is still relatively low. Only Delicious dataset seems to be more resistent to p-core pruning, 80% of tag assignments are removed at $p = 40$. There are two factors that are likely to influence the ratio of tag assignments left in the p-cores — the size of the dataset and the type of resources stored in the system. To check which one has greater importance we repeated the experiment for Delicious dataset limited to a certain number of posts. We extracted two samples of Delicious posts, with 50% and roughly 10% of the earliest

posts. The second sample contained the a similar number of tag assignments as the full CiteULike dataset. For low values of p both CiteULike and Delicious 10% datasets retain a similar number of tag assigments, the difference grows with the growth of p. The Delicious sample also reaches much higher maximal non-empty p-core (over 30). It suggests that the sparsity of the dataset is mostly related to its size, but the outcome of p-cores pruning for high p depends on the specific characteristics of the dataset.



(a) All broad folksonomies, first ten cores

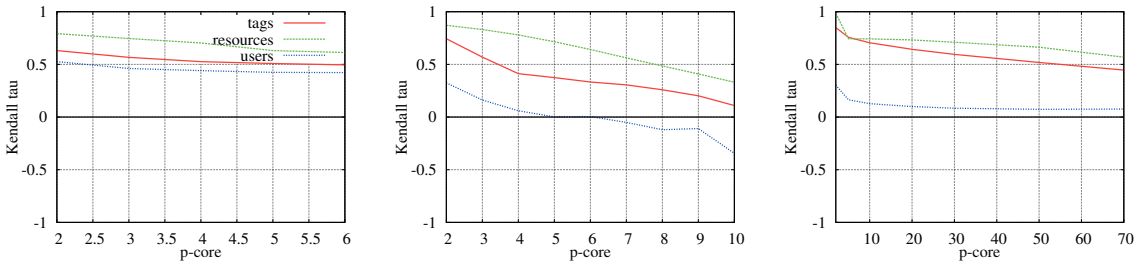
(b) Delcious dataset, full-range of non-empty cores

Figure 3.9: The ratio of tag assignments left in a dataset after the application of p-cores pruning. For BibSonomy and CiteULike datasets, p-cores pruning quickly removes a great majority of assignments. Delicious dataset is more resistant to pruning for small values of p, mainly because of its size.

By definition, p-cores pruning mostly affects the long-tail of the distribution of the folksonomy graph elements (i.e., tags, resources, users), by removing the low frequency elements. However, it is possible that frequently occurring elements will also be affected by the pruning. For example, a user with many posts of unique resources would lose large part of the profile after the pruning. To observe the impact of the p-cores pruning on the distribution of the most frequently occurring elements we compared the original ranking of the top 100 elements with their ranking in the pruned datasets. For this purpose we used Kendall's τ correlation coefficient. The coefficient takes into account the order relation between all pairs of elements in the rankings. Given two rankings $rank1$ and $rank2$ The pair (a, b) is classified as an agreement if $rank1_a < rank1_b$ and $rank2_a < rank2_b$ or if $rank1_a > rank1_b$ and $rank2_a > rank2_b$. The pair is classified as a disagreement if $rank1_a < rank1_b$ and $rank2_a > rank2_b$ or if $rank1_a > rank1_b$ and $rank2_a < rank2_b$. If in any of the rankings both ranks are equal, the pair remains unclassified. Given the counts of agreements and disagreements for all pairs of n ranked elements the correlation coefficient is defined as:

$$\tau = \frac{\#agreement - \#disagreement}{\frac{1}{2}n(n-1)} \tag{3.1}$$

Kendall's τ coefficient decreases for all elements with the increasing number of p in p-cores pruning (Fig. 3.10). Therefore, more aggressive pruning introduces more disturbance even among the most frequently occurring elements. For all datasets, the distribution of resources is the least and the distribution of users is the most affected. It can be mostly observed for CiteULike and Delicious datasets, where even for low values of p the correlation of the original and pruned ranking is very low. It suggests that there is a large group of users which posts mostly unique or infrequently posted resources or use very specific tag vocabulary to describe their resources. These users are strongly underrepresented in the pruned version of the dataset. For CiteULike dataset many of these users are among the most active users. However, in the process of pruning they are completely removed from the dataset. As a result, the user ranking from the original dataset is anti-correlated with the rankings for pruned datasets of high p.



(a) BibSonomy (b) CiteULike (c) Delicious

Figure 3.10: Rank correlation between the original distribution of top 100 tags, resources or users and their distribution after p-cores pruning. For all datasets p-cores pruning mostly affects the user distribution. The difference between the distributions grows with increasing value of p.

As we demonstrated, unless the dataset contains millions of posts, p-cores pruning makes important changes to the dataset even for small values of p. The removal of the majority of posts poses a question about the efficiency of tag recommendation systems tested on pruned datasets. Despite the fact that p-cores pruning focuses on low frequency elements it also affects the most frequent elements, especially the users, therefore it reduces the general representativeness of the datasets, while used in the evaluation of tag recommendation systems.

3.8 Summary

The presented experiments create a complex picture of collaborative tagging data. From the perspective of the tag recommendation problem we can point out three potentially useful sources of tags: resource profile, user profile and resource title. The sparsity of tagging datasets reduces the availability of resource profile; however, when available it becomes an accurate source of tag recommendations. On the contrary, user profile is rich and accessible source of tag recommendations, which usefulness deteriorates with the increasing number of posts, likely due to changes in the interests of users. This issue can be mitigated by taking into account the temporal characteristics of user actions. Additional experiments on user profiles demonstrated that users tend to re-use the same form of a tag keeping their profile coherent. Although, when the tag is less important for them they are more likely to be influenced by other factors (e.g., resource title). The title turns out to be an accurate source of tags, which is, however, limited by its size.

In an additional experiment on p-cores pruning, we demonstrated that, unless the dataset contains millions of posts, even small values of p are likely to remove majority of tag assignments, which affects the characteristics of the dataset, both considering the least and the most occurring elements. It hurts the practical value of the evaluation on pruned dataset both in terms of the effectiveness and the efficiency of the evaluated system.

Chapter 4

System Design

In this chapter we present the conceptual design of the system and system architecture. The system is designed to effectively utilize the advantages of the various tag sources discussed in the previous chapter. At the same time, in system architecture design we focused on the efficiency, scalability and simplicity of the recommendation process.

4.1 Conceptual Design

The proposed system is a hybrid tag recommender composed of five basic recommenders. Such modular structure allows the system to utilize various tag sources and properties of folksonomy data structure created collaboratively by taggers. The three basic tag sources are (a) content of the tagged resource, (b) resource profile, which are the tags used for the same resource by other users and (c) user profile, tags previously used by the user. To extend and refine the set of tags extracted from resource content the system uses two graph-based recommenders which run a *spreading activation* algorithm [13] using *content-to-tag* or *tag-to-tag* co-occurrence graphs. Each of the basic recommenders produces a tag recommendation set which contains tags and scores in [0,1] range. The scores represent the likelihood of a tag being used by the user.

The main idea behind the design of the recommendation process (Fig. 4.1) is to utilize the specific advantages of each source of tags and combine the results produced by each of them to produce the final recommendation. As the system is a hybrid tag recommender that utilizes several tag sources, there is a large space of possible combinations of basic system components. The proposed system structure is based on two objectives. First, we wanted a system with a structure that can be easily explained to the users. Second, to reduce the complexity of parameter learning, we limited to two the number of tag recommendation sets, that are combined at once.

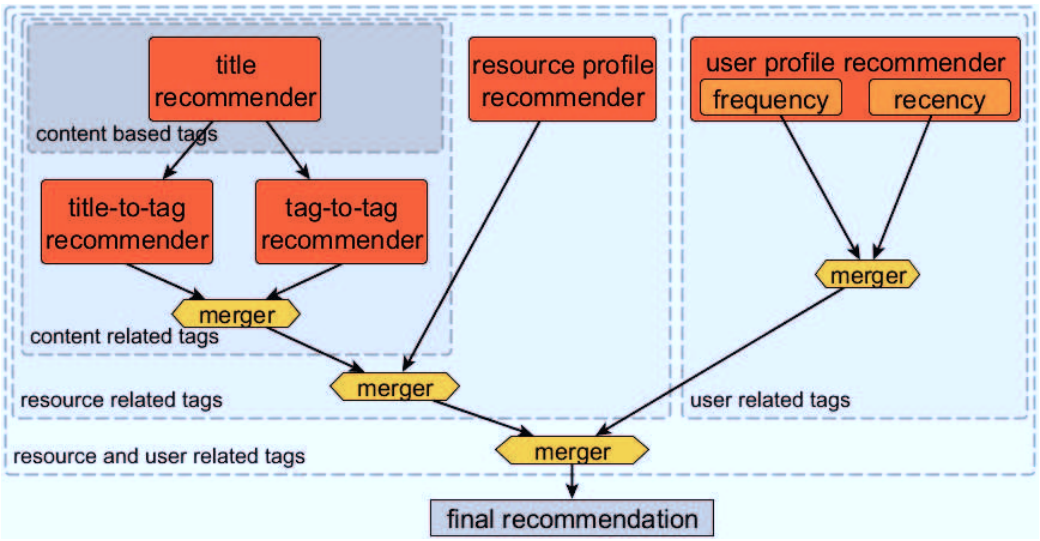


Figure 4.1: The tag recommendation system scheme. Tags from five basic recommenders are merged at different stages of processing.

extracted from the title with the scores attached are used as **content based tags**.

Spreading Activation in Term Co-occurrence Graphs

The set of tags that can be extracted from the title is limited by its size. In addition, is likely to be biased by grammar and lexical rules of natural language as well as the vocabulary of the content author. At the same time, users often choose to modify the tag that describes the concept represented by the title word to maintain the consistency of their profiles. For example, term *network* can be modified to *networks* or more specific *social-network*. One of the ways to get access to related tags is to exploit the co-occurrence of tags in previous posts. Such technique was proven to be successful in tag set expansion tasks in which, a limited set of tags already assigned to a resource was expanded by a larger set of related tags. In previous work, the related tags were accessed using various forms of co-occurrence graphs [63] or by mining association rules [53]. In our system, the access to other forms of the term and related terms is gained using the spreading activation algorithm. Spreading activation is a technique used broadly in various areas from Artificial Intelligence [31] to Information Retrieval [15]. It operates on a weighted graph, in which the weights represent the strength of relations between the connected nodes. Given a set of source nodes the approach can be used to search for related elements in the graph (e.g., concepts in a semantic network [13]). Starting with a set of *source nodes* with

assigned *output values* O_i, the algorithm activates nodes connected to them assigning them an *input value* I_j, which is calculated based on the output score of the source nodes and the weight of the connection using an *input function*. The function can be arbitrary designed to match the problem characteristics. The input is then processed to form output values and the activated nodes are used as source nodes in the next "pulse" of the algorithm. The stopping criterion can be based on a specific number of pulses used in the algorithm or the decay factor combined with firing threshold. The decay factor affects the input function decreasing the value of input value with the increasing number of pulses. The firing threshold removes activated nodes with low output values from the set of source nodes used in the next pulse. In our setting, we use the content based tags as a set of source nodes and run the algorithm on a term co-occurrence graph. We use two types of term co-occurrence relations. The first type represent the relations between words used in the resource title and the tags used to tag the resource. The second type represents the relations between tags that are used together in the same posts.

Title-to-tag Recommender

Title-to-tag recommender runs the spreading activation algorithm on a directed co-occurrence graph of terms, which were used as title words or tags. The graph (V, E), where V is the set of nodes and E is the set of edges, contains all terms ($v_i \in V$) that were used as title word or tag. The terms are connected by directed edges (e_{ij}) from a node v_i to v_j, where i is a title word and j is a tag associated with the same post, in which i can be found as title word. The weight w_{ij} of the edge between two terms is equal to the number of posts, in which they occurred together, divided by the total number of occurrences of the term i as a title word. To avoid overgeneralization, our system accepts the output of the first pulse as the result of the process. The output value O_i is the score attached to a word by the title recommender. The input of an activated tag I_j is used as the score of the title-to-tag recommender. The input function uses the formula for the union of probabilities of independent events (Eq. 4.1) to ensure that the produced scores are in $[0, 1]$ range.

$$I_j = 1 - \prod_i (1 - O_i w_{ij}) \tag{4.1}$$

Tag-to-tag Recommender

An analogous approach can be applied to a *tag-to-tag* graph. The graph captures the relations between tags that frequently co-occur in the same posts. Unlike the title-to-tag graph, this graph is not likely to represent connections between terms that convey similar meaning because most users try to avoid redundancy while tagging. The objective of this graph is to capture hypernymic relations between tags. The system runs spreading activation on the *tag-to-tag* graph using the set of tags extracted from the title. The tags extracted from both graphs are merged producing a set of **content related tags**.

Resource Profile Recommender

The set of tags related to the resource content is extended by the tags extracted from the resource profile (all tags previously used for the resource). The score of a tag extracted from the profile is its frequency (the number of posts in which the tag was found) divided by the total number of posts of the given resource. The collaborative effort of users makes resource profile a very precise source of tags, pushing the best tags to the top of frequency-ranked list. Unfortunately, this source is rarely usable as most of the resources added to broad folksonomies are unique. This is also the case for all resources in narrow folksonomies. This is why in our system resource profile tags are only a supplement to the content related tags. Together they create **resource related tags**.

User Profile Recommender

User profile is a very rich, but noisy source of potential tag recommendations. It is likely to contain tags representing different user's interests and activities, which change dynamically. Tags frequently used in the past are not necessary a good current recommendation. To adjust to this fact, the user profile recommender uses an additional recency-based scoring scheme, to complement the frequency-based scheme, as in the resource profile recommender. Both schemes produce two identical sets of tags with different scores. The sets are later merged, so the final score is a linear combination of the scores proposed by both schemes. The outcome of this recommender is a set of **user related tags**. Finally, resource related tags and user related tags are

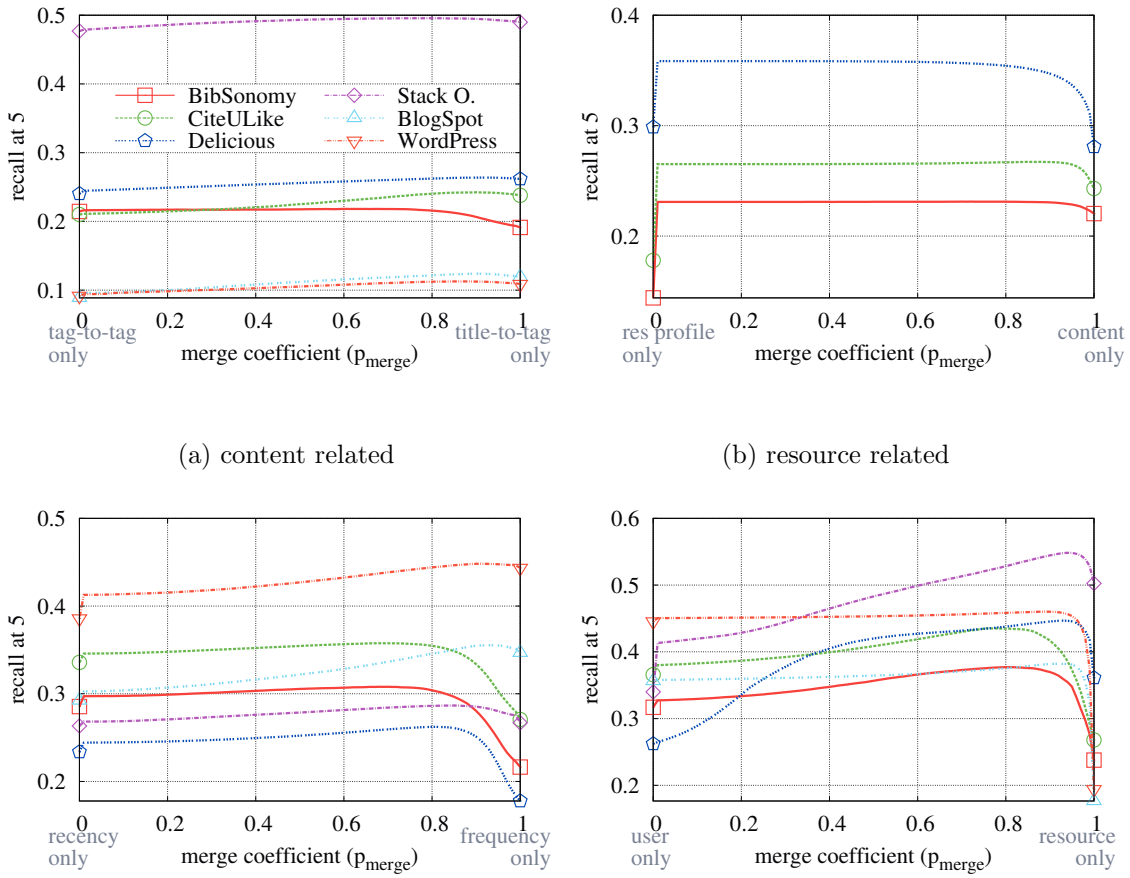
merged to produce the **final recommendation**.

4.1.2 Results Merging

Given the results of various processing steps it is crucial to combine them in a way that preserves the most promising tags in the top of the ranking of the output set. To achieve this we standardized the scores produced by all recommenders ($s \in [0, 1]$). The tag recommendation sets are combined by *mergers* which take two tag recommendation sets as input, linearly re-score the tags given the *merge coefficient* $p_{merge} \in [0, 1]$, representing the relative importance of both sources of tags, and merge the sets adding the scores of the tags that can be found in both input sets.

To understand how the choice of the merge coefficient influences the quality of the result set, we discretized the range of p_{merge} into 101 values ($p_{merge} = 1$ represents tags from the first input set only and $p_{merge} = 0$ represents tags from the second input set only). We used the 80% of the posts with the earliest time-stamps to build the folksonomy and then we iteratively, in timestamp order, added the remaining 20% of test posts to the repository calculating the average quality score (i.e., $recall@5$) for each value of p_{merge}. As a result we obtain the *merge quality curve*, which is the quality score (recall@5) as a function of the merge coefficient, for each merger (Fig. 4.2). The comparison of the curves shows that each merger has its specific characteristic, which also depends on the dataset. In many cases (e.g., the merger producing user related tags for the *Delicious* dataset), the optimal value of p_{merge} is closer to the input set with the lower quality, which is counterintuitive. In general, the optimal value of p_{merge} tends to be close, but not equal, to one of the extremes. The results of the experiment confirms the need of a parameter tuning algorithm that would be able to predict the optimal value of p_{merge} for each merger.

We decided to base the design of the parameter tuning algorithm on the characteristics of merge quality curves. It seems that the shape of the curve is constant for a given merger-dataset pair. As a result, two different subsets of posts taken from the same dataset should lead to the same optimal value of the merge coefficient. In addition, the shape of the curve is smooth, in the sense that a small change in p_{merge} is not able to make a dramatic change in the tag ranking, hence it cannot affect the score. Therefore, it is possible to choose a nearly optimal value of p_{merge} from a discrete number of choices. Given these two observations, parameter tuning becomes



(a) content related

(b) resource related

(c) user related

(d) resource and user related

Figure 4.2: Merge quality curves represent the quality score (*recall*@5) averaged over the set of test posts, as a function of the merge coefficient p_{merge}. The border points of each curve demonstrate the accuracy of a single input set (as marked by the label below). Each merger is presented on a separate plot. The comparison of the curves obtained for different datasets shows differences among their characteristics.

a simple optimization task, which can be solved by recording the average quality of the merger given a limited set of p_{merge} values and use the value that has the highest quality. We used this learning approach to tune the merge coefficient to a value that overall produces results with the highest average quality.

4.2 Scalable System Architecture

The large amount of processed data as well as the need of a short response time make the system architecture a crucial aspect of the design process. The main challenge in the implementation of the presented tag recommendation system is the representation of the co-occurrence graphs and the tag profiles (for resources and users). In both cases it is clear that, given the amount of data and open-ended vocabulary, they cannot be stored in operational memory. The system architecture must consider three phases of post processing: recommendation, indexing and training.

4.2.1 Recommendation

For performing a recommendation task the system needs to extract two tag profiles (for the resource and the user) and a series of references to the co-occurrence graphs. The number of these references is limited by the size of the content based recommendation set. To simplify the problem, the co-occurrence graph lookup can be reduced to the tag profile lookup task. A tag profile for a term represents all tags that co-occurred with it in any of the posts, while the frequency of co-occurrences can be used to calculate the weight of the connection. To extract a tag profile for a post element (i.e., user, resource, tag or content word) the system uses a text indexing engine (Apache Lucene[1]), which stores all previously processed posts. By accessing the Lucene index directly, the system is able to quickly retrieve a list of posts that contain a given element. As the extraction of posts is a much more time consuming task, we decided to limit the number of posts, based on which the profile is built, to the 1000 most recent posts that contain the element.

To reduce the number of references to the index, the system contains a layer of caches (Fig. 4.3). Each element type has a separate tag profile cache. If the system hits the profile in the cache, it does not have to refer to the index. In case of a miss,

[1] http://lucene.apache.org/

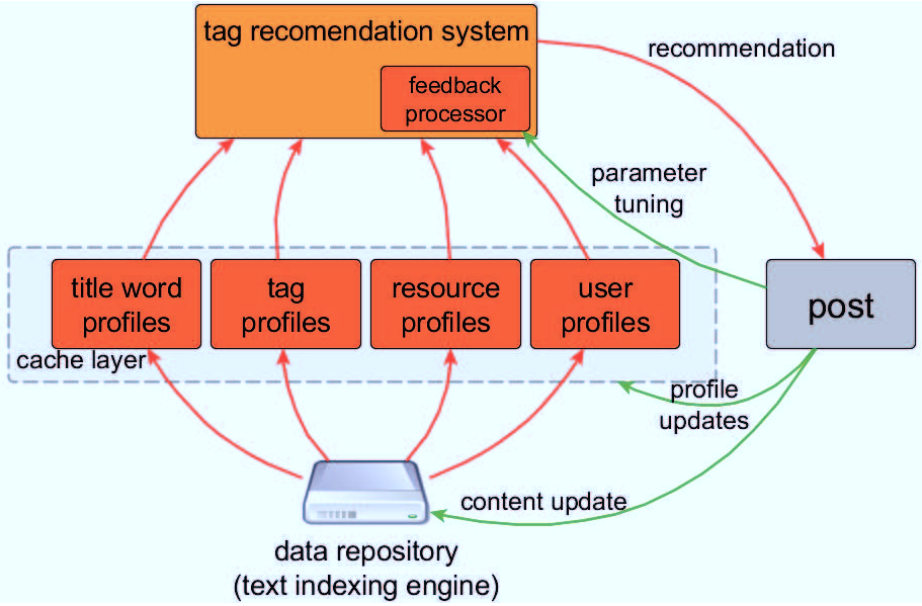


Figure 4.3: System architecture. The cache layer improves system efficiency. Utilization of the feedback loop allows online content adaptation.

the profile is built based on the information extracted from the index. If the element was used in over 20 posts, its profile is added to the cache replacing the profile with the lowest value of replacement function. We experimented with two basic replacement policies: Least Frequently Used and Least Recently Used and found their performance inconsistent. In the system we decided to use a combination of recency and frequency factors (Eq. 4.2), which in most cases is able to match or outperform the better of the two basic policies (see Section 5.3.1 for details).

$$rf(item) = \frac{frequency(item)}{currentTime - lastTimeUsed(item)} \tag{4.2}$$

4.2.2 Indexing

Whenever a new post is added to the system, it is stored in the text index. Each of the post elements is indexed separately. In addition, the tags entered by the user are used to update each of the relevant profiles in the cache layer (Fig. 4.3). This approach solves the cache synchronization problem without the need for additional information extraction from the index. To better control the memory usage, the system has a hard constraint on the maximal size of a profile that is stored in the cache layer. Tags, that are relatively rare in the profile or have not been entered into it recently,

are not likely to become prominent enough in the recommendation process to reach the top of the list that is finally presented to the user, so they can be omitted. To retain potentially useful tags, the profile itself is also implemented as a cache, using the same cache replacement policy (Eq. 4.2).

All the constraints put a hard limit on the memory allocated to the storage of the profiles in the cache layer; however, they can potentially decrease the quality of recommended tags as they limit the number of posts and tags used in the recommendation process. To investigate whether this is the case we removed the constraints from the system and ran it for the *BibSonomy* dataset (see Section 3.1 for dataset details). The difference in the quality of the results produced by the constrained and unconstrained version is negligible. Therefore, the constrains on posts and tags do not play a negative role in the recommendation process.

4.2.3 Parameter Tuning

User tags are also passed to the *feedback processing module*, which is responsible for tuning the mergers. The module stores the input sets used by all mergers while processing the post. Given user tags, it is able to reproduce the merging process for different values of merge coefficient and learn the optimal value online. The system has no parameters that would have to be defined by the administrator to tune its performance towards a specific underlying collaborative tagging system. In addition, as both indexing and parameter tuning is done every time a post is added to the system, there is no need for additional re-indexing or re-training steps, which greatly simplifies system maintenance.

Chapter 5

Evaluation

We evaluated the proposed tag recommendation system using datasets from six collaborative tagging systems described in details in Section 3.1. The datasets represent a wide variety of tagging systems in terms of type of folksonomy, its size, time-span of posts and character of posted resources. The system was evaluated from the perspectives of its effectiveness and efficiency. The effectiveness evaluation included the experiments, which tested the system's ability to tune its parameters to the characteristics of a specific collaborative tagging system and the quality of recommendations produced by the system and its processing stages. In comparative evaluation we used a sequence of datasets pruned by p-cores extraction to compare the system to state-of-the-art techniques used in parameter tuning and graph-based recommendation. The efficiency evaluation focused on the performance of the cache layer as well as throughput and response time of the system.

5.1 Effectiveness Evaluation

While evaluating the effectiveness of the system we focused on three aspects of the recommendation process. First, we present the experimental results for parameter tuning module, which objective is to adapt the parameters of mergers to the processed posts. We show that the system is able to tune its parameters to specific characteristics of processed dataset. Second, we discuss the impact of the online content adaptation feature that is available in our system. The fact that the system is able to instantly use the incoming posts to update its data repository has significant impact on the accuracy of recommended tags. Finally, we compare the final results of the system to the results of processing stages to show the importance of utilization and merging of tags from different sources.

In Section 5.2, we present the comparative evaluation of the system, using two

state-of-the-art systems. The performance of the parameter learning module is compared to SVMrank [37] a classification system that is adapted to the ranking regression problem. The overall efficiency of the system is compared to Pairwise Interactive Tensor Factorization (PITF) method [59] and its modification based on Factorization Machines [56]. Both of them can be considered as an adaptation of the collaborative filtering approach to the graph-based tag recommendation task.

5.1.1 Discussion on the Evaluation Methods in Tag Recommendation

Despite the large number of publications on tag recommendation problem, little has been done on the unification of the evaluation methods. In fact, most of the systems are evaluated in a unique way proposed by their authors. In some cases the evaluation methodology follows the specific application of the system and it is unlikely that we can find a "one-fits-all" evaluation approach for all tag recommendation systems. The objective of this section is to present the main features of tag recommendation system design and what follows the evaluation methods presented in the literature. We focus on off-line recommendation task in which the recommendations are evaluated based on the tags entered to the system without the use of the recommendation system. While evaluating the system, it is assumed that all and only correct tags were provided by the users. We decided to adopt this approach because of its simplicity and popularity, although it has certain limitations [20]. An alternative approach, in which the set of recommended tags is assessed by judges is used in a great minority of publications [63, 65]. Asking humans to help with the evaluation of recommendation results is cost-prohibitive for large scale evaluation. Based on the features we classify various evaluation approaches, we discuss their practical usefulness and provide rationale for the design decisions of the evaluation process used in this work.

Recommendation recipient A basic classification of tag recommendation systems can be done based on the target recipient of the recommendation. The fact who is going to use the recommendation determines the way in which the quality of the recommendation is evaluated. We can determine two main recommendation recipient types:

User (per post) The majority of the tag recommendation systems considers the

user who is entering a post to the system as the main beneficiary of the recommendation. From the user's perspective the recommendation system should propose tags that are tailored to personal interests and tagging style. They should also contain user specific tags (e.g., *mythesis*, *toread*) In such case, to perform an off-line evaluation it is assumed that tags entered by the user in a specific post represent the gold standard of recommendation. This way we refer to this evaluation approach as *per post* evaluation.

Community (per resource) Some tag recommendation are designed keeping in mind the community that would benefit from highly descriptive tags assigned to a resource by a group of users [4, 43, 64]. The objective of the tag recommendation system is then to support the collaborative knowledge formulation process. The recommendation system should avoid the personal tags and focus more on highly descriptive tags related to the tagged resource. In such case, the recommendation is done *per resource* and it is often assumed that the gold standard are the most popular tags from resource profile. Recommending tags to a community of users creates an important question about the specific user or users who are supposed to review and accept the recommendations. If the recommendations are accepted by the author of the post then we should take into account the personal preferences and the evaluation based on the most commonly used tags does not seem to be representative. On the other hand, it is not clear how we can select other users that represent the interests of the community. Therefore, it can be more reasonable to consider this task as an automatic tagging approach [51].

Prior knowledge assumptions The authors of tag recommendation systems often make assumptions about the access to prior knowledge about the posts or resources. These assumptions have impact on the practical usefulness of the proposed systems. Among the most frequently made assumptions we can find:

Rich textual content of resources This assumption is necessary for content-based recommendation methods; however, we have to remember that the access to the textual information is strongly dependent on the character or resources stored in a tagging systems. In systems storing references to scientific publications, (e.g., BibSonomy or CiteULike) the textual content in the form of publication

abstracts is sparsely available [45]. In social bookmarking systems (e.g., Delicious) the access to the textual content of a web-page may require additional processing steps [73]. Finally, some resources do not contain rich textual content by their nature (e.g., photos in Flickr). Therefore, dependence on the textual content makes the recommendation system specific to a dataset for which it is designed.

Previously assigned tags Some recommendation systems aim to extend the tag-based description that is already present in the system [63, 53, 4]. In this case the system assumes that each resource was previously assigned a set of tags. Such systems can serve as interactive recommendation methods in which the set of recommendations is modified based on the tagging decisions made online by the user. The automatic evaluation of these systems is done by hiding a subset of tags assigned to a resource, the objective of the system is then to re-discover the hidden tags.

Dense folksonomy graph This assumption is made for all graph-based tag recommendation methods [21, 35, 57, 59, 66]. These systems are evaluated using the dense core of the folksonomy graph obtained by p-cores extraction. As we discussed in Section 3.7, this approach focuses on a very small subset of real tag assignments which reduces the practical usefulness of the recommendation results.

Constrained tag vocabulary The low-frequency tags are the hardest to recommend, due to lack of information that can be used in the recommendation models. It is sometimes assumed that these tags should be considered as noise, which results in a constraint vocabulary of tags. Therefore, during the evaluation the tags that cannot be found in the training set frequently enough are disregarded. Alternatively, in open tag recommendation it is assumed that all posts and all tags assigned in them are used to train and test the system.

Selection of test instances Separating the test instances from training instances is an important factor that can strongly bias the evaluation results and undermine their validity. Here we discuss the main test sample selection methods used in the literature:

Random The basic method of splitting training and test instances is a random split of the dataset. This method is especially useful if we plan to run cross-validation tests [66]. The random split has two important disadvantages in *per post* recommendation task. First, it breaks the temporal relations between posts, giving the system access to information about future actions of users, which would not be available in practical conditions. Second, the approach causes a great risk of biasing the results for datasets, in which the imported posts were not properly removed. Splitting the large number of imported posts between training and test data would cause the system to be trained and evaluated on a small but frequently occurring group of artificial tags.

Time-based The problem of training the system on instances which are more recent than the test posts is resolved by the time-based selection of the test posts [64, 66]. Assuming that we have access to the time-stamps of all posts we can arbitrarily select a set of the most recent posts as test instances. This approach to some degree resolves the problem of bias caused by imported posts; however, as noticed in [45] it introduces another bias related to the representation of user profiles. Users tend to add their posts in burst, therefore it is possible that a great majority of user's post will be placed in either training or test set. As a result, the system is trained on posts of users who are not longer active in the system for posts of users which have not yet become active.

User-based This selection of test posts is a form of *leave-one-out* approach [28] and is popular among graph-based recommender systems [35, 59]. Its main advantage is that it creates diversified and balanced test set from the perspective of user representation, even if the number of instances in the entire dataset is low. The method was proposed by Jaeschke et al [35], who randomly selected a single post for each user present in the dataset. The recommendation experiment was repeated for each of the selected posts. In each run, the post was kept as the test instance and all other posts were used as a training set. A modification of this approach was proposed by Rendle et al. [59]. They extracted a single post for each user present in the dataset and used all these posts in the test set. The choice of the posts was done randomly, to the extend that the remaining training set would still preserve the p-cores characteristics. We can easily think

of a combination of both sampling approaches and select for the evaluation the most recent post of each user. The usefulness of this sampling approach is however is questionable for the open tag recommendation problem, because of the highly skewed distribution of user activity. The user-based sampling method would over represent the users from the long-tail of the user distribution (a large number of users with very few posts). At the same time, users from the heavy tail (few users with a large number of posts), would be underrepresented.

Evaluation metrics In off-line tag recommendation evaluation a ranked list of recommended tags is compared with the set of real tags provided by the user. The task is analogous to standard information retrieval tasks, hence the basic IR measures are used to evaluate tag recommenders.

Precision@N and recall@N Both precision and recall are among the most commonly used information retrieval evaluation metrics [28]. In the context of tag recommendation problem precision is calculated for a single processed post (or resource) as the number of correct tags recommended divided by the total number of recommended tags. Recall is calculated as the number of correct tags retrieved divided by the total number of correct tags. The reported scores are an average over all tested posts (or resources). In off-line tag recommendation it is assumed that the correct tags are the tags assigned to the resource by a user or a community of users. As the set of recommended items can be arbitrary large, usually it is limited to the top N tags that have the highest recommendation score. If the scoring scheme used by a recommendation system works properly both precision@N and recall@N should be monotonic with N. As the number of recommended tags grow, the precision should decrease as tags with low score are less likely to be correct. Recall@N is constantly growing with N, because the denominator in recall formula is independent of the number of returned tags and larger number of tags increases the chance of a match between the result set and the set of correct tags. However, the increase in the value of recall@N should be largest for low N.

F1-score Recall and precision are often combined into a single score — F1@N, which is the harmonic mean of both recall and precision calculated for N results (Eq. 5.1). F1-score is a commonly used in the evaluation of tag recommendation

systems [35, 51, 57, 59, 64]. In addition, this metric was the main evaluation criterion for both ECML/PKDD Discovery Challenges. Unlike, precision and recall F1@N is not monotonic. In fact, the maximal value of the score seems to be related to the number of correct tag assignments. It was shown that limiting the number of produced tags M to the estimated number of correct tags can significantly increase the F1@N, where $M < N$ [58]. Although limiting the number of recommended tags seems to be an easy way to improve the performance of the system it is not practical in tagging systems, which aim to provide a constant number of recommended tags in each case. Most importantly, focusing on two or three most accurate tags does not seem to be beneficial for the user, which suggests that F1-score does not represent the practical value of the recommended tags.

$$F1@N = \frac{2 \times P@N \times R@N}{P@N + R@N} \tag{5.1}$$

Mean average precision (MAP) Another measure used to evaluate tag recommendation systems is mean average precision. Average precision score is an extension of the precision that takes into account the ranking of the recommended elements. The score is described in Eq. 5.2, where S is the complete set of ranked recommended tags and $isCorrect$ function returns 1, whenever the tag at position n is correct. The $|correctTags|$ value is the total number of correct tags. It is used to keep the value of AP score in $[0, 1]$ range. Mean average precision is the AP score averaged over all test samples. This metric is often used in *per resource* recommendation tasks [4, 43]. However, the practical usefulness of taking the tag ranking into consideration in *per post* is questionable. Despite the fact that tag recommendation systems return a ranking of recommended elements, *per post* tag recommendation seems to be a binary task. Most of the tagging systems present the recommended tags in alphabetical order to make them easier to comprehend for users. At the same time, the number of recommendations presented to the user is limited to a small set (usually five to ten tags).

$$AP = \frac{\sum_{n=1}^{|S|} P@n \times isCorrect(n)}{|correctTags|} \tag{5.2}$$

Precision/recall plot This approach is a popular visualization method, which provides an overview of a performance of a recommendation system [21, 35, 66]. The plot presents a sequence of precision@N and recall@N pairs for the growing value of N. This characteristic informs us whether the system focuses on a small set of tags that are very likely to be correct (steep precision/recall curve) or a broad set of tags with low precision of each single tag but high recall of the set (flat precision/recall curve). Precision/recall plots can be used to compare the general properties of recommendation systems, but not the overall performance as they do not provide a single performance evaluation metric.

5.1.2 Methodology and Measures

We based the evaluation of our system on the methodology used in ECML/PKDD Discovery Challenge 2009. The detailed evaluation of the system was based on the evaluation procedure used in two challenge tasks: content-based task and online tasks. Both of them can be classified as open, time-based evaluation method. In the content-based task the organizers provided a full dump of the BibSonomy system data up to April 1st 2009, the posts entered in the following two months were released later as a test set. The online task was run for approximately three months after the release of the test. In this setting the contestants could use both training and test data to train a system that was serving real users in the system. The system had access to the feedback information, namely the real tags entered to by the user after the evaluation. Therefore, the recommendation system could perform an online adaptation of recommendation models and parameters. Due to constraints of the systems used in the comparative evaluation, in this setting we had to follow the evaluation procedure used in the third task of the ECML/PKDD Discovery Challenge 2009 — graph-based task. The evaluation procedure in this task followed the procedure from content-based task, with the exception that the training set contained only 2-cores of the original training set and the test set contain only users, resources and tags that could be found in the training set.

In preparation to the evaluation, we sorted the posts chronologically and separated roughly the latest 20% of posts to test the system; the 20% of the posts that precede them were used to tune the system parameters. The division was made just to clarify the presentation of the process. In the system in operation, indexing, parameter

tuning and recommendation are done simultaneously in a single run.

In our opinion, a tag recommendation system should provide the user the maximal possible number of correct tags, given a hard constraint on the number of tags recommended (usually five). If the limit of tags presented to the user is five, a system that gives the user five tags, the last three of which are correct, is better than a system, which proposes two tags only, even if both of them are correct. This is why we decided to adopt *recall*@5 as the main quality criterion. We also use this measure to tune the parameters of the tag recommendation system.

5.1.3 Learning the Merge Coefficients

One of the most important aspects of the proposed hybrid tag recommendation process is the combination of results from the basic recommenders. In our system we decided to use a processing stream, in which at each step the system combines two tag recommendation sets. We also proposed a parameter tuning method based on the merge quality curve. The optimal value of a merge coefficient is greatly dependent on the type of input recommenders and data characteristics, the value for a single merger used for a specific dataset seems to remain relatively constant. Thanks to rich user feedback (i.e., real tags entered in each post) the system is able to learn the value of merge coefficient that would produce the best average result over large number of posts. To asses if the system is able to predict the optimal value of the merge coefficient we ran the following experiment.

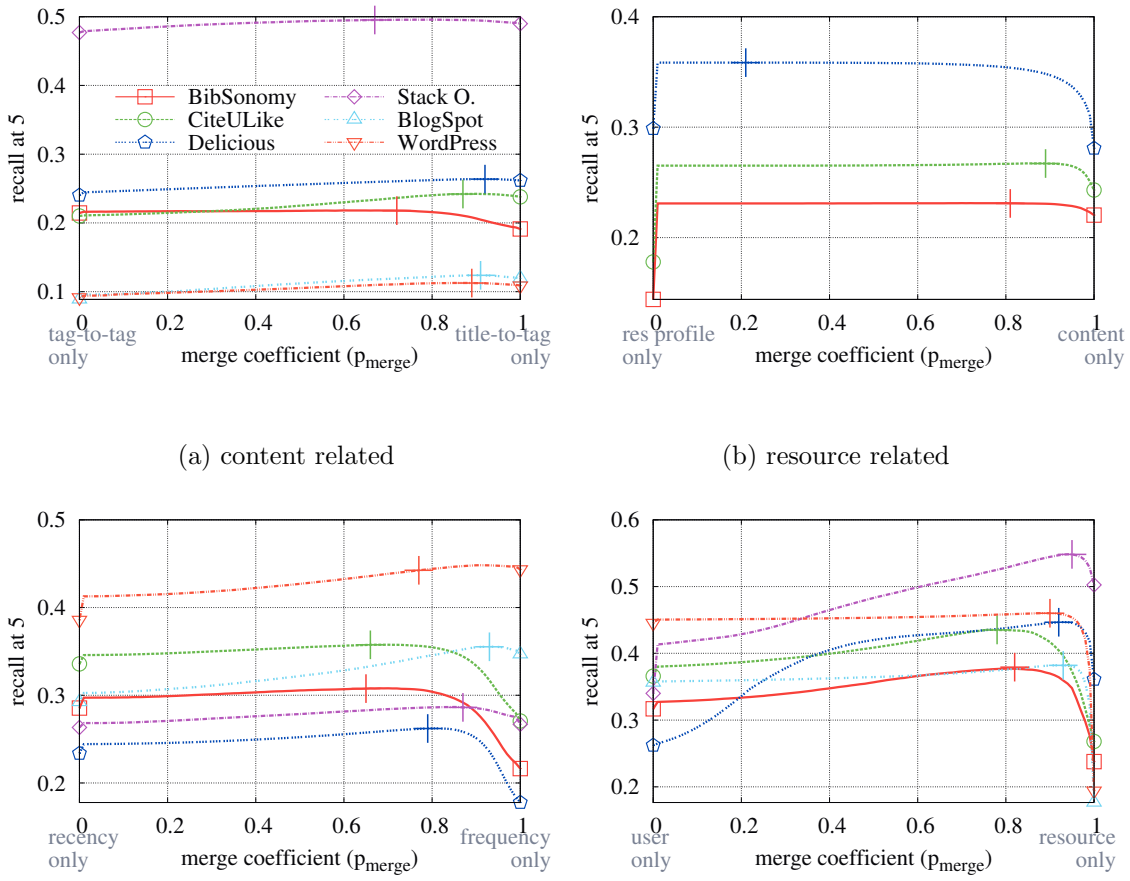
We used the set of last 20% of posts in each dataset as the test set. The parameter learning was done on the 20% of the posts that precede the test posts. We produced the merge quality qurves for training posts and test posts and observed if the optimal value of merge coefficient estimated for the training posts matches the value calculated for the test posts. The experiment confirmed that the learning method is able to discover nearly optimal values of p_{merge} in almost all cases. On the plots, the learned value of each merge coefficient and *recall*@5 obtained for this value are represented with a cross (Fig. 5.1). The only case in which the learning approach failed to predict the correct value of p_{merge} is the merger producing user related tags for the *WordPress* dataset. Most likely it is caused by the presence of a large group of users with different tagging patterns in the test set. To mitigate such problems we experimented with online adaptation of the merge coefficient based on a sliding window over the most

recent posts. However, we found this approach less computionally efficient and slightly less effective for most of the merger-dataset pairs.

5.1.4 Online Content Adaptation

To observe the impact of online content adaptation on the results and provide a baseline for the system we ran a series of experiments in which this feature was turned off. The parameters of the system were re-trained to tune it to the new conditions. The adaptation improves the results of the recommendations for all tested datasets (Table 5.1). The statistical significance of the difference was confirmed by a Wilcoxon signed-rank test ($P < 0.001$). For the three broad folksonomies, online content adaptation has a clear impact on the relative importance of different tag sources. We present the plots of recall and precision for each stage of the recommendation process, without and with adaptation, to show how they contributed to the final result (Fig. 5.2). For all datasets the largest improvement is noticed for the user related tags. Adaptation allows the system to extend the repository of user related tags by the tags that describe user's recent interests. The system is also able to gather information about new coming users, from the moment they start to use the system. It is especially important for the *BibSonomy* and *CiteULike* datasets, for which we observed a large number of users who started to use the system in the test period. For these two datasets user related tags become the richest and most accurate source of tags. This is not the case for the *Delicious* dataset where the improvement of user related tags is comparable to resource related tags. It seems that the availability of a large number of newly added posts allows resource profiles to overcome the problem of cold start — the noisiness of profiles of infrequently posted resources [43]. Finally, the adaptation seems to have little or no impact on the content related tags extracted from the co-occurrence graphs. The associations between tags are well established at the time of the evaluation and they are not changed by the adapted content. In this case the adaptation is likely to be useful in the early stage of folksonomy formulation only.

Online content adaptation has lesser impact on the narrow folksonomies. The relative importance of processing stages remains the same independently of the use of online content adaptation. For the *Stack Overflow* dataset this is caused by the fact that the recommendations are mostly based on the content related tags. The



(a) content related

(b) resource related

(c) user related

(d) resource and user related

Figure 5.1: Merge quality curves represent the quality score ($recall@5$) averaged over the set of test posts, as a function of the merge coefficient p_{merge}. The border points of each curve demonstrate the accuracy of a single input set (as marked by the label below). Each merger is presented on a separate plot. The proposed parameter learning method is able to discover the optimal value of p_{merge} for almost all of the merger-dataset pairs. The learned value of p_{merge} and $recall@5$ obtained for it are represented with crosses.

Table 5.1: Recall@5 for the final recommendation. For all data online content adaptation gives statistically significant improvement ($P < 0.001$, Wilcoxon test).

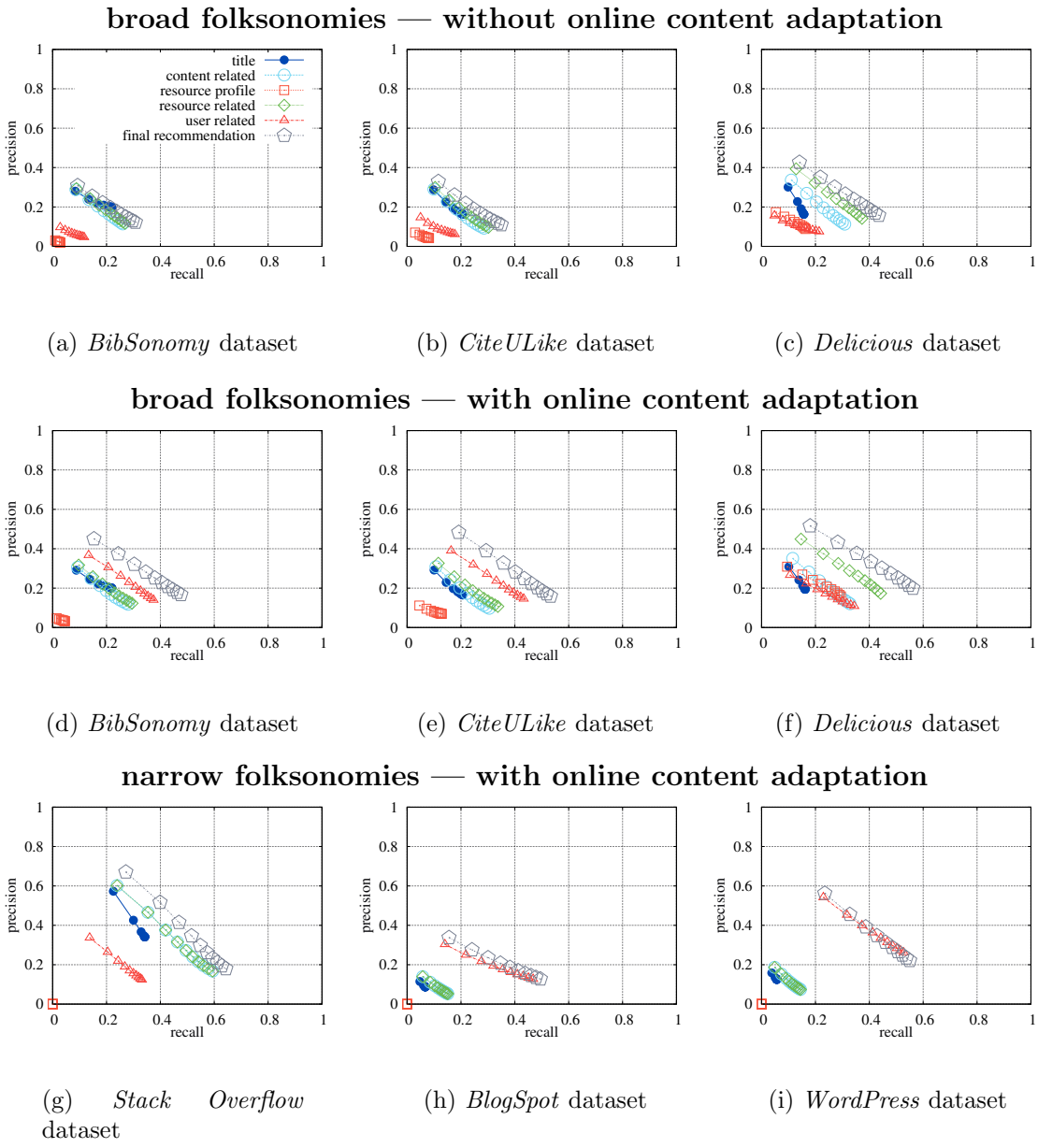
dataset	**no** adaptation	**with** adaptation	**increase**
BibSonomy	0.238	0.379	0.141 (60%)
CiteULike	0.272	0.435	0.163 (59%)
Delicious	0.343	0.447	0.104 (30%)
Stack Overflow	0.498	0.548	0.050 (10%)
BlogSpot	0.355	0.382	0.027 (8%)
WordPress	0.427	0.460	0.033 (8%)

most likely reason of this outcome for blog datasets is the short time-span of posts and consistency of users in the choice of the most frequently used tags.

5.1.5 Results of Processing Stages

Comparison of the precision/recall plots for the final recommendation and intermediate processing stages reveals that the importance of tag sources for recommendation strongly depends on the characteristics of the collaborative tagging systems (Table 5.2 and Fig. 5.2). Among the broad folksonomies the results for two similar systems *BibSonomy* and *CiteULike* are much alike, while at the same time being different from the *Delicious* results. The first two systems are used to gather resources related to research interests of the users. Well defined interests of the users result in higher precision and recall of recommendation based on user profiles. In comparison, *Delicious* gathers bookmarks which represent general interests of users, hence the user based recommendation is noisier. On the other hand, the size of *Delicious* results in higher percentage of non-unique resources and allows the recommendation system to take advantage of the resource profiles. Low performance of the resource profiles for *BibSonomy* and *CiteULike* reveals an important characteristic of these datasets — uniqueness of posted resources. The percentage of posts that contain a resource that can be found in only 5 or less posts of the overall dataset is 95% and 80% respectively. This implies that unless the repository contains millions of posts, as in case of the *Delicious* dataset, resource profile is not a practical source of tag recommendations.

Among the narrow folksonomies we can observe a clear distinction between *Stack*



(a) *BibSonomy* dataset (b) *CiteULike* dataset (c) *Delicious* dataset

(d) *BibSonomy* dataset (e) *CiteULike* dataset (f) *Delicious* dataset

(g) *Stack Overflow* dataset (h) *BlogSpot* dataset (i) *WordPress* dataset

Figure 5.2: Precision/recall plots for $k \in [1, 10]$ top tags produced at each stage of tag recommendation process. Comparison between the system without and with online content adaptation, for broad folksonomy datasets, shows the positive impact of adaptation on the quality of user related tags, resource related tags and the final recommendation.

Overflow and two blog datasets (*BlogSpot* and *WordPress*). The authors of programming questions posted in *Stack Overflow* try to design informative titles so they are easier to find by others, which makes the title a rich source of potential tags. Conversely, the authors of blogs tend to make their titles attractive to catch the attention of the readers. Hence, the overlap between attractive title words and informative tags is low. In addition, the authors of blogs use the tags as a personal classification scheme for their posts which results in higher precision and recall of user based tags. Despite the differences among datasets, our system is able to utilize the most accurate tag sources and combine them in the final recommendation. In all cases *recall*@5 of the final recommendation is higher than the best basic recommender used as a baseline (Table 5.2). The difference is statistically significant ($P < 0.001$, Wilcoxon test). We used the Wilcoxon signed-rank test because we deal with repeated measurements om the same set of samples (i.e., the accuracy scores calculated for a set of posts processed by the baseline and the hybrid recommendations system) and we are not able to make any assumptions about the underlying distribution of sample scores. Given the differences in the characteristics of various datasets we decided to apply the statistical test to each dataset independently.

Table 5.2: Recall@5 for all recommendation stages (baselines) and the final recommendation. In all cases the final recommendation result is better than baselines (statistically significant, $P < 0.001$, Wilcoxon test).

dataset	title	content related	resource profile	resource related	user related	final recommendation
BibSonomy	0.205	0.218	0.039	0.231	**0.308**	**0.379**
CiteULike	0.197	0.242	0.109	0.267	**0.357**	**0.435**
Delicious	0.160	0.264	0.242	**0.358**	0.262	**0.447**
Stack Overflow	0.341	0.495	0.000	**0.495**	0.286	**0.548**
BlogSpot	0.067	0.124	0.000	0.124	**0.355**	**0.382**
WordPress	0.056	0.113	0.000	0.113	**0.443**	**0.460**

5.1.6 Impact of Hybrid Components

The examination of the results of the different processing stages revealed the importance of the two final tag recommendation sets (resource related tags and user related tags). To gain deeper insight into the system's performance we turned off some of its components. We focused on the generation of two tag recommendation sets: content

Table 5.3: The impact of removing specific system components on the final recommendation. Two tag recommendation sets are considered – content related tags and user related tags. The boldface values point out the base recommender which results with the highest accuracy when not removed.

	content related tags			user related tags		
dataset	title-to-tag graph only	tag-to-tag graph only	no spreading activation	frequency scheme only	recency scheme only	full system
BibSonomy	0.366 (-3.4%)	**0.375** (-1.1%)	0.366 (-3.4%)	0.319 (-15.8%)	**0.362** (-4.5%)	**0.379**
CiteULike	**0.429** (-1.4%)	0.424 (-2.5%)	0.407 (-6.4%)	0.373 (-14.3%)	**0.419** (-3.7%)	**0.435**
Delicious	0.445 (-0.4%)	**0.447** (0.0%)	0.439 (-1.8%)	0.406 (-9.2%)	**0.437** (-2.2%)	**0.447**
Stack O.	**0.540** (-1.5%)	0.538 (-1.8%)	0.489 (-10.8%)	**0.544** (-0.7%)	0.538 (-1.8%)	**0.548**
BlogSpot	**0.378** (-1.0%)	0.375 (-1.8%)	0.371 (-2.9%)	**0.375** (-1.8%)	0.319 (-16.5%)	**0.382**
WordPress	**0.456** (-0.9%)	0.455 (-1.1%)	0.448 (-2.6%)	**0.461** (0.2%)	0.405 (-12.0%)	**0.460**

related tags and user related tags (Table 5.3). The content related tags are a combination of results of the spreading activation algorithm on two term co-occurrence graphs: title-to-tag graph and tag-to-tag graph. Discarding the results of one of the graphs has a slight negative impact on the final recommendation results. Although the tags extracted from the two graphs are not identical, there is some overlap between them. Therefore, if needed, one of them can be removed to increase the efficiency of the system. Discarding the results from both graphs causes a significant drop in system performance. The value of *recall*@5 for the final recommendation set drops up to 11% for the *Stack Overflow* dataset. All other datasets are mostly dependent on the user related tags. The scores in the user related tags set are a combination of two scoring schemes, based on frequency or recency of tag use. Discarding one of the schemes reveals interesting differences between datasets. The recency factor has great importance for broad folksonomies. Possibly, the overall frequency of tag use does not reflect the current interests of the user. The opposite behaviour can be observed for blog datasets. Here, the users have a constant set of categories that represent their intrests and they switch between them often decreasing the importance of the recency scheme, up to the point where, for the *WordPress* dataset, removing the recency scheme increases the quality of recommended tags. This unexpected outcome is the result of a poorly trained parameter, as discussed in Section 5.1.3.

5.2 Comparative Evaluation in Graph-based Recommendation Task

Considering the comparative evaluation of the overall system performance we have to refer to the results of the ECML/PKDD Discovery Challenge 2009. The challenge provided an opportunity to test various tag recommendation systems on equal conditions in three diversified tasks. The predecessor of our system won two out of three challenge tasks, including the online recommendation task, in which the evaluation was performed in the real condition of the tag recommendation system usage. Since that time both effectiveness and efficiency of the system improved significantly. To the best of our knowledge, there is no tag recommendation system with capacities to compete with our system in the open tag recommendation task. Therefore, we decided to run a comparative evaluation in graph-based recommendation tasks. This is the only setting in which the predecessor of the system did not take the first place in the ECML/PKDD Discovery Challenge 2009. We compare our system with the winner of the graph-based recommendation task and its modification. The basic version of the system is based on Pairwise Interaction Tensor Factorization ($PITF$) algorithm [59], its modification utilizes Factorization Machines (FM), which can be considered as a generalization of the tensor factorization approach [56]. During the evaluation process we downloaded the code of tag recommendation system based on both techniques from author's web-site[1] and ran the experiments on our datasets.

One of the most crucial elements of the presented system is the merging of results of the basic tag recommenders. In our system this task is solved by pairwise tag recommendation set merging and the merge parameter is optimized using a heuristic based on the merge quality curve. To evaluate this approach we compare it with *SVMrank* [37], a fast ranking method based on Support Vector Machines regression [14, 27]. Parameter learning in hybrid tag recommendation task is analogous to a well-studied problem of learning retrieval functions in meta-search engines [36]. The task there is to combine a ranking of documents retrieved by several search engines into a single meta-ranking. In tag recommendation problem, each basic tag recommender can be considered as a single search engine and the tag recommendation scores can be mapped to document relevance scores. Unlike our system, SVMrank model is based on a support vector, which assigns a single parameter to each input

[1] http://cms.uni-konstanz.de/informatik/rendle/software/tag-recommender/

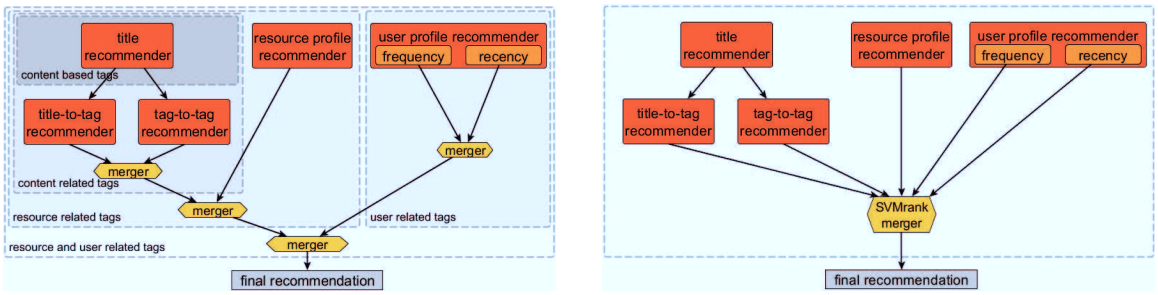


Figure 5.3: The basic design of the system based on pairwise merging of tag recommendation sets in four steps (left) and the system design based on one step merging using SVMrank (right).

ranking. The input rankings are linearly combined based on the parameters in a single step (Fig. 5.3). To evaluate the performance of SVMrank we used the five basic tag recommenders which rankings are combined to produce the final recommendation scores in our system, namely title-to-tag recommender, tag-to-tag recommender, resource profile recommender and user profile recommender with frequency and recency ranking schemes. This way, SVMrank fully replaced our pairwise merging approach. Thanks to an alternative representation of the SVM optimization problem [37, 38], SVMrank is able to efficiently train its model in linear time with a constant number of iterations, that depends only on the number of input rankings, not the number of training instances. Nevertheless, to iterate over the instances set efficiently, all of them have to be stored in operational memory. For this reason, we could not perform fair comparison of the two learning approaches in the open evaluation setting, because a large number of training instances. One alternative would be to randomly sample the training set; however, a more meaningful sampling is already available through the p-cores sampling approach. Therefore, we decided to use the graph-based experimental setting for the comparison of all recommendation techniques: our system with and without online content adaptation, PITF, FM and SVMrank. In case of SVMrank we used the rankings generated from the basic tag recommenders of our system with the online content adaptation feature. During the experiments we used the code of SVMrank system downloaded from author's website[2].

[2]http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

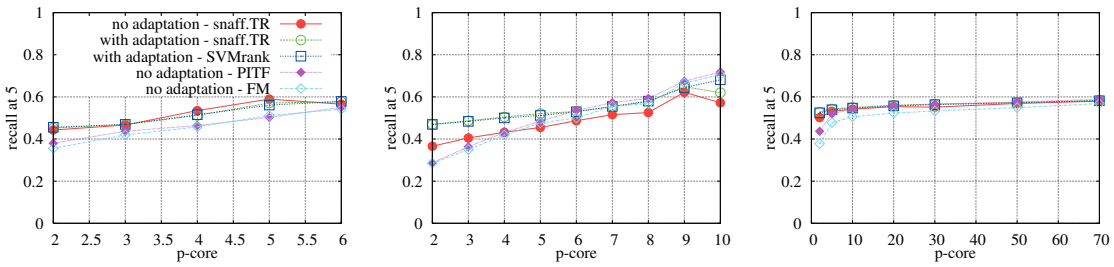
5.2.1 Evaluation Methodology

To create a uniform evaluation setting for all tested algorithms we adapted the test procedure described in Section 5.1.2. In particular, we used the same time-based 60/20/20 split to separate posts that are used for (1) pre-indexing, (2) training and (3) testing respectively. To assure that all users, resources and tags were used at least p times in the training set of the PITF algorithm and at least p times in the indexed posts prior the training procedure of our algorithm and SVMrank, we run the p-core extraction algorithm, as described in Section 3.7, on the posts in set (1). After p-cores were extracted from set (1), sets (2) and (3) were pruned, so they contained only users, resources and tags which could be found in the p-cores of set (1). As the PITF algorithm does not require a special set of posts used for pre-indexing, we combined the sets (1) and (2) together as a training set for PITF.

The pruning procedure was used to extract p-core training and test samples for the three broad folksonomies used in the evaluation of our system. Depending on the size of the dataset we extracted a set of p-cores for a growing number of p to observe the performance of the algorithms for while the densification of the dataset is increasing.

5.2.2 Comparison with PITF and FM

Following the previous sections we use *recall*@5 as the main performance factor. Figure 5.4 presents the results of all systems with the growing value of p, which determines the p-core level. Comparing the results of our system and two factorization based methods we can observe a common pattern across all tested datasets. The relative performance of our system is the highest for p-cores with low p and decreases with its growth. It is important to notice that by increasing the value of p we remove more tag assignments which makes the problem less practical. In all cases the performance of the factorization based methods is lower than the performance of our system for $p < 6$. For BibSonomy, six is the highest value for which a non-empty p-core can be extracted. As the value of p grows over this threshold the performance of the systems is different depending on the dataset. For CiteULike, factorization based methods begin to achieve better performance than our system. However, the number of training and test instances at this point is very low and as we discussed in Section 3.7 the p-cores sample for this dataset are highly disturbed by the pruning. For



(a) BibSonomy (b) CiteULike (c) Delicious

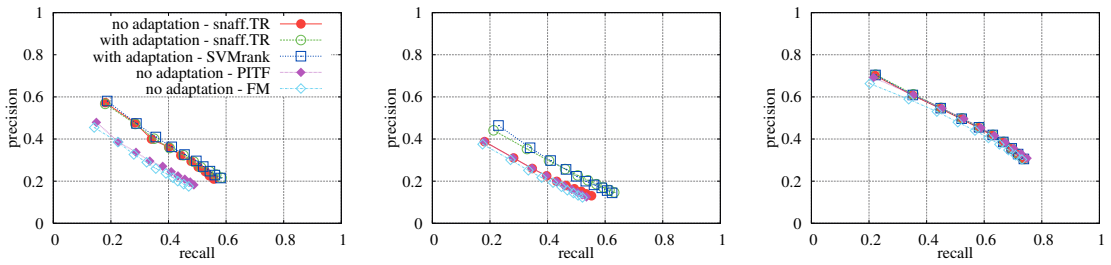
Figure 5.4: Recall@5 for p-cores of an increasing number of p. Our system outperforms state-of-the-art approach (PITF) for low values of p, which represent the most practical cases.

Delicious, at some point all systems begin to produce results of very similar accuracy and the accuracy seems to stabilize for high values of p. This result is unexpected given the properties of the Delicious dataset (i.e., non-empty cores for high values of p and the correlation between tag-based and resource-based similarity) that makes it most applicable to the factorization based approaches among all broad folksonomies.

In comparison to the PITF and FM methods the performance of our system is much less affected by the sparsity of data. We can observe it specifically for Delicious dataset, where the results of the factorization methods for low cores are much worse comparing to denser datasets. Therefore, we can conclude that or system much better handles the cold-start problem. At the same time, we can observe that for high values of p the performance of all systems stabilizes, it suggests that there is an upper limit of the accuracy of recommendation caused by the unpredictability of users decisions.

5.2.3 Comparison with SVMrank

For all datasets and p-cores the performance of the pairwise merging approach and SVMrank is nearly identical. The only difference can be observed for the highest cores of BibSonomy and CiteULike, where the number of training and test posts is lower than 100. In these cases the performance of our system decreases in comparison to lower values of p. It suggests that the approach based on the merge quality curve is incapable of selecting the proper value of merge parameter with a small number of training instances. This issue, however, has low practical importance as in most cases we can expect a large number of training instances. However, our system has



(a) BibSonomy, 2-core (b) CiteULike, 4-core (c) Delicious, 70-core

Figure 5.5: Precision-recall curves of five tag recommendation systems for chosen p-cores. The characteristics of all systems are very similar. The difference between our system with and without online content adaptation is much lower than in case of the open recommendation task. In comparison to our system SVMrank focuses more on the quality of the top recommended tag.

a number of practical advantages over the SVMrank approach. It has much lower memory use thanks to stream processing instead of batch processing approach. It can be updated iteratively if the condition of the recommendation are changing. For example, the system can be trained offline based on the database dump and then fine-tuned while it is operating online.

To gain more insight into the results of all systems we plotted the full precision/recall curves for chosen values of p for each dataset. Comparison of the results of our system and SVMrank shows slight difference in the performance of both systems which is caused by a different optimization criterion. While our system optimizes $recall@5$. The SVMrank algorithm can be seen as the optimization of ROC-Area [37]. As a result, SVMrank achieves slightly higher $precision@1$ score and slightly lower $recall@k$, for high k. It is mostly noticeable for CiteULike dataset (Fig. 5.5(b)). This fact can be seen as an advantage of our system which is more flexible in a sense that it can be used to optimize any performance factor.

5.3 Efficiency Evaluation

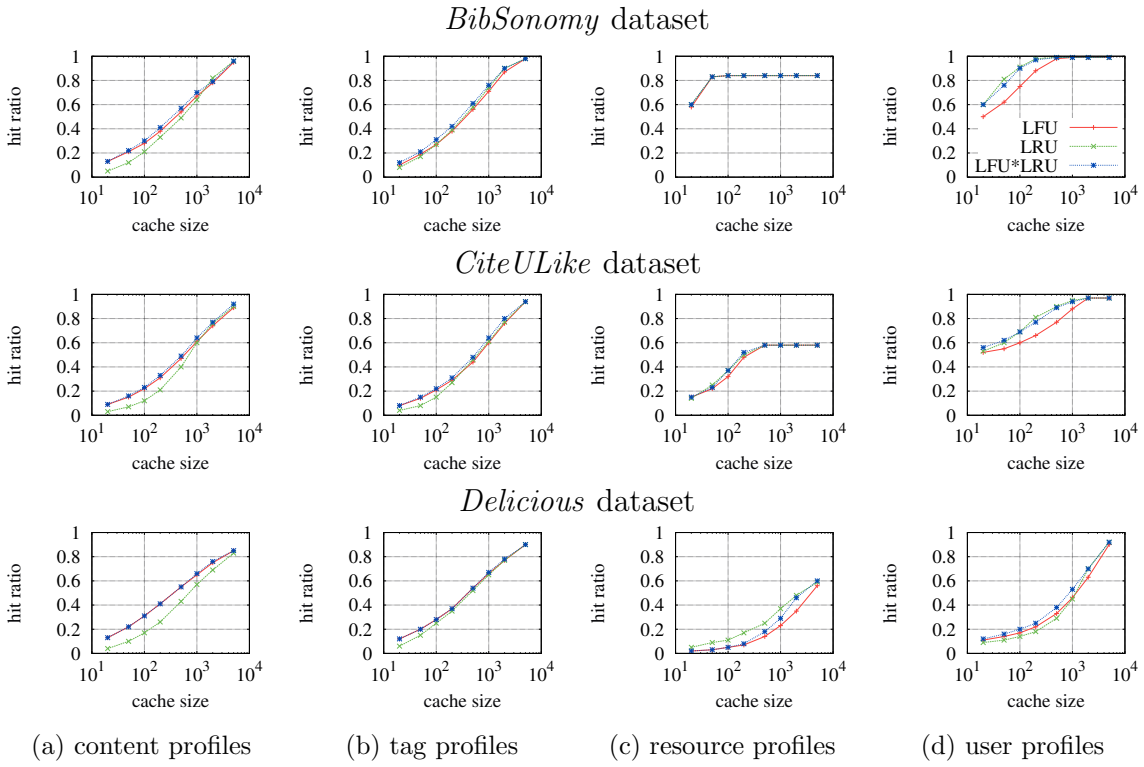
In the evaluation of the system efficiency we were interested in three characteristics: (a) cache hit ratio which defines the performance of cache layer, (b) system throughput (the number of posts processed per minute), which defines the ability of the system to serve large number of requests and (c) response time, which is the time needed

to generate a recommendation. We based the evaluation of system throughput and processing time solely on the *Delicious* dataset, because of its size. We also decided not to use a high-performance server machine, as it is hard to control the impact of its configuration (e.g., RAID, hard-drive cache) on the processing efficiency. The tests were performed on a personal computer with a 32-bit, dual-core, 1.73 GHz CPU and 7200 RPM, 16 MB Cache, SATA (3 GB/s) hard drive and with the system restricted to 1.5 GB of memory. We iteratively added the posts to the index in chronological order, restarting the system twice: after indexing 60% of posts (over 5 million) and 80% of posts (over 7 million). The objective of restarts was to observe potential scalability issues. After the restart, full recommendation was run for 8 hours.

5.3.1 Cache Hit Ratio

To determine the usefulness of cache layer we ran a sequence of experiments on three broad folksonomy datasets (*BibSonomy*, *CiteULike* and *Delicious*). We measured the ratio of cache hits in relation to the cache size for various profile types. In all experiments we used $100,000$ posts from the training and testing sets. Three cache replacement strategies: Least Frequently Used (LFU), Least Recently Used (LRU) and combination of both were compared. The objective of the experiment was to determine which replacement strategy has the highest hit ratio. In addition, we wanted to gain better insights about the trade-off between performance and memory requirements of the cache layer for each post element (i.e., content words, tags, resources and users).

The comparison of the results for different datasets and post elements reveals their specific characteristics (Fig. 5.6). The plots of the cache performance for the profiles of content words (Fig. 5.6(a)) and tags (Fig. 5.6(b)) show that the cache hit ratio grows linearly with the logarithm of the cache size. Such behaviour is expected as content words and tags produce heavy-tailed frequency distributions (Fig. 3.1). As a result, small caches are able to handle majority of tag-related requests. For example considering the BibSonomy dataset, the cache that is able to handle profiles of 4% of all distinct tags ($5,000$ cache size to $111,343$ distinct tags cf Table 3.1) gives 0.96 cache hit ratio. Similar results were obtained for tag profiles of content words. Unlike tags and content words the use of user profiles seems to clutter in time. Users tend to add their posts in bursts which results with higher cache hit ratio for



(a) content profiles (b) tag profiles (c) resource profiles (d) user profiles

Figure 5.6: Cache hit ratio — the number of times the element was found in the cache divided by the total number of times it was used.

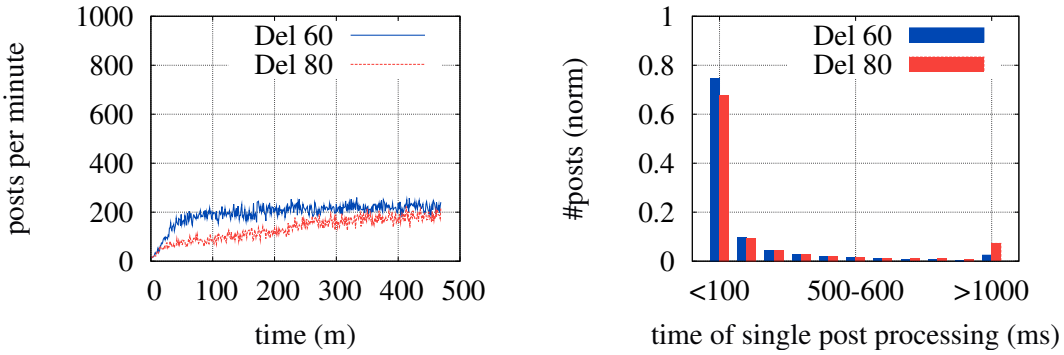
LRU strategy. In addition, large number of user profiles becomes inactive after some time. It can be observed for BibSonomy and CiteULike datasets. For BibSonomy dataset, despite the fact that the dataset contains posts from over $5,000$ users, most of them were inactive during the tested time frame, hence the cache of 500 profiles was sufficient to store all used profiles. For Delicious dataset, in which the number of active users is much higher, increase of the cache size results in similar performance improvement as for the tag profiles cache. The lowest performance of the cache layer can be observed for resource profiles. High sparsity of resources results in low cache hit ratio independently of their size (e.g., CiteULike dataset).

Although the cache performance differs depending on the dataset and element type, the cache of 5000 profiles is enough to obtain over 80% cache hit ratio, with the exception of the resource profiles cache. However, in this case the sparsity of resources makes the cache layer less important, as the resource profiles are small (if not empty) and can be easily retrieved from the underlaying database. Therefore, we conclude that the use of cache layer has a practical potential of improving the

recommendation time. This fact, was confirmed in the next experiment, in which we measured the system throughput. In this experiment, the system was initialized with empty caches. They were filled in with profiles needed for currently processed posts. It dramatically decreased system's performace in terms of the number of processed posts per minute before the caches filled up.

5.3.2 System Throughput

During the experiments we logged the number of posts processed per minute (Fig. 5.7(a)). To confirm their impact on efficiency, the caches were not pre-fetched. With 5 million posts in the index the system needed around 50 minutes to fill the caches with useful profiles. After that it demonstrated stable throughput of around 200 posts processed per minute (which adds up to 288,000 posts per day). The comparison with the test performed with 7 million posts in the index shows that the time needed to recover to stable performance is increasing with the increasing number of indexed posts. The cold-start problem can be mitigated with cache pre-fetching. After a restart the caches can be filled with profiles of the most frequent elements. By the cost of longer initialization the system is then able to produce recommendation in acceptable time from the start.



(a) Number of posts processed per minute (includes indexing).

(b) Histogram of recommendation times per single post.

Figure 5.7: Processing efficiency. Both post throughput (nearly 300K posts per day) and recommendation times (majority of posts processed within 100ms) are well above usability levels for practical tag recommendation for systems like BibSonomy, CiteULike or Stack Overflow (confront Table 3.1).

5.3.3 Response Time

Together with the number of posts processed per minute we recorded the recommendation time for each processed post. For both restarts we disregarded the first $10,000$ posts processed to reduce the impact of system recovery stage. The histogram of single post processing times shows that the majority of posts was processed in less than 100 ms (Fig. 5.7(b)). The ratio of posts processed longer than one second is around 2-3%. This ratio is much higher for the second restart, which is caused by a longer recovery stage.

Apart from the restart recovery problem the system displays stable performance. In the range of 9 million posts from *Delicious* dataset we were not able to observe substantial scaling problems other than a longer recovery stage without cache prefetching. It shows that the system would be practically usable for most of the collaborative tagging systems available; however, it is certainly limited by the capability of the underlying text indexing system.

Chapter 6

Additional Aspects of Tag Recommendation

In this chapter we present the results of a set of additional experiments on the proposed tag recommendation system. The contribution of the experiments is two-fold. First, they allowed us to gain more insight into the performance of the system and the characteristics of the processed datasets. Second, they evaluated potential ways of improving the system's accuracy. In particular, we were interested if the system can be improved using the ideas and methods proposed in other streams of tag recommendation research. The chapter is composed of three independent set of experiments on three aspects of the tag recommendation problem:

- performance of the system for frequent tags (Section 6.1)
- utilization of textual content for tag recommendation (Section 6.2)
- usefulness of various patterns of user tagging behaviour (Section 6.3)

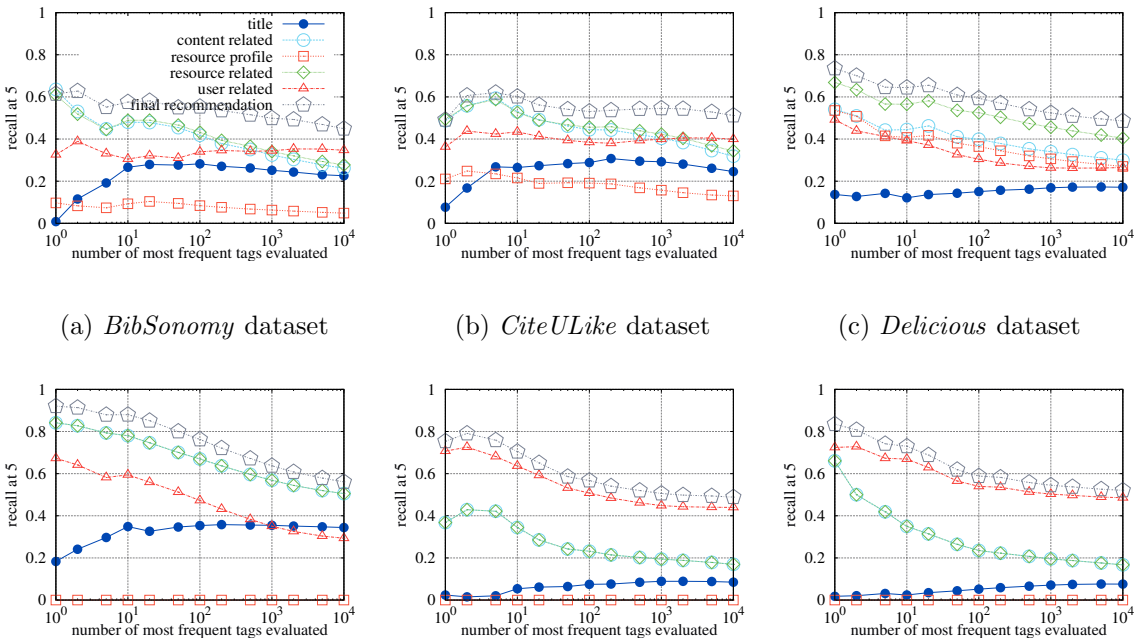
Both graph-based recommendation and some content-based techniques recommend tags from a fixed vocabulary. In addition, a tag should be used frequently enough to provide information about its usage. On the contrary, our system utilizes title words and newly added user tags which give it constant access to unique tags. The use of an open-ended vocabulary of tags creates a risk of underestimation of frequent tags. In such case, a possible extension of the system would be to utilize these recommendation methods designed for frequently used tags as an additional post-processing step. To determine if such extension is potentially useful, we re-evaluated the outcome of our system considering the most frequently used tags only. Another interesting aspect is the utilization of textual content of the resource (other than the title). This issue seems to be especially important for blogs, where the quality of tags extracted from the title is particularly low. To address this problem we turned our attention towards key-phrase extracting technique used for topic indexing. We demonstrate how this approach can be used for the extraction of low

frequency tags. Finally, we looked into the concept of recommendation personalization and more generally various patterns of user tagging behaviour. First, we tried to capture the individual tagging patterns of users by personalized learning of merging coefficients. Second, we proposed a tagging pattern processing module which is able to extract a set of patterns that represent various settings of the tag recommendation system. Unfortunately, due to the lack of proper features that characterize the patterns, the potential level of accuracy improvement is not reached because of poor post-to-pattern assignment.

6.1 Frequent Tags

Unlike graph-based and content-based recommendation techniques our system does not explicitly focus on frequently used tags, which creates a potential area of improvement. If the system failed to recommend frequent tags with high accuracy, its results could be combined with the results of a system that focuses explicitly on these tags. To test if such extension is needed, we re-evaluated the results of the system considering the top $N \in [1, 10000]$ tags, sorted by the frequency of occurrence in all posts. Posts which contained no tags from the set of the most frequent tags were removed from the evaluation process. It is important to notice that we did not prune the list of recommended tags by removing the low frequency tags. Although, it would certainly improve the accuracy of the system, it would defeat one of the purposes of the experiment, which was to determine if the system needs an additional module to increase the rank of frequently used tags among all recommended tags.

The results of the experiment show that the system achieves much higher recall@5 score considering the most frequent tags only (Fig. 6.1), comparing to the results of the system evaluated for all tags (Table 5.2). In most cases the largest improvement is noticed for the top few tags. The accuracy of recommendation decreases with the increasing size of the most frequent tags set, which is an expected behaviour, given that less frequent tags would become harder to recommend. The same pattern can be observed for user related tags, which show that the recency ranking scheme is not impairing the quality of recommendation for high frequency tags. The results for title based tags show that the title is not a good source of the most frequently used tags, which seem to be too general to be used as title words. However, this limitation is overcome by the use of spreading activation algorithm, which confirmed its ability to



(a) *BibSonomy* dataset (b) *CiteULike* dataset (c) *Delicious* dataset

(d) *Stack Overflow* dataset (e) *BlogSpot* dataset (f) *WordPress* dataset

Figure 6.1: Recall@5 for each processing stage considering $N \in [1, 10000]$ most frequent tags only. Despite the use of open-ended vocabulary the system is able to recommend frequent tags with high accuracy.

generalize the content based tags. In fact, for the most frequent tags, the content related tags have the highest accuracy (excluding the blog datasets).

When focusing on the differences between datasets, we observe a very low performance of the system for the most frequent tag in the *CiteULike* dataset (i.e., *review*). *Review* is an example of *task organizing* tags [22], which are generally hard to recommend. There is no concrete context in which the tag occurs, therefore it is not well captured in the term co-occurrence graphs. Our observations show that such tags are in a great minority among all tags, which are generally used to describe a resource. Therefore, we did not design the system to focus on this type of tags. When the context of a tag is well-defined, the co-occurrence graphs are able to recommend tags with very high accuracy. The example of such behaviour can be observed in the *Stack Overflow* dataset, where the most frequently used tags are the names of programming languages (Table 6.1). In this case, the spreading activation algorithm, ran on term co-occurrence graphs, successfully replaces a classification algorithm assigning a post to a related category (programming language). A similar situation is observed for

Table 6.1: The list of five most frequently used tags and the number of their occurrences for each dataset.

BibSonomy		CiteULike		Delicious		Stack O.		BlogSpot		WordPress	
web2.0	(8K)	review	(27K)	design	(400K)	c#	(98K)	politics	(19K)	politics	(67K)
software	(7K)	evolution	(18K)	software	(329K)	java	(60K)	john-mccain	(10K)	life	(57K)
tools	(6K)	model	(13K)	web	(310K)	php	(51K)	barack-obama	(10K)	news	(49K)
web	(5K)	theory	(12K)	blog	(302K)	.net	(48K)	sarah-palin	(9K)	music	(33K)
blog	(5K)	network	(12K)	music	(288K)	javascript	(44K)	music	(8K)	family	(30K)

the *WordPress* dataset, where the top tags represent the general categories of posts. On the contrary, the *BlogSpot* dataset is strongly skewed towards a single topic — presidential election in the U.S. (the data was collected in fall 2008). The top four tags are related to each other. The most frequent tag *politics* is much more general than the other three tags (names of candidates). In addition, the names of candidates are often used together, which creates strong co-occurrence scores between them. As a result, they are recommended together, downgrading the more general tag *politics* in the recommendation ranking, lowering the accuracy of the system for this tag. The *BlogSpot* dataset seems to be the only case where a technique tailored for frequent tags could be beneficial.

6.2 Tags Based on Textual Content

The textual content of the resource is a valuable source of tags as it potentially provides access to the low frequency tags, which otherwise are very hard to find. On the other hand, the resource content can be costly to retrieve or process. Therefore in the basic version of the system we decided to use the resource title as the only content related source of tags and expand it using the information that is already present in the tagging systems (i.e., tag co-occurrence graphs). Nevertheless, in some systems the use of the full textual content of the resource may be beneficial. To investigate this issue we experimented with the textual content of the three narrow folksonomies. First, we looked at the potential usefulness of the content based tags for blog posts. Second, we evaluated the benefits of using a key-phrase extraction method in the tag recommendation problem.

6.2.1 Usefulness of Content-based Tags

The comparison of the system performance on various datasets revealed the high variability of the accuracy of tags extracted from the resource title. For some types of resources (i.e., blog posts) the title is a poor source of content based tags. In such cases, the keywords extracted from the textual content of the resource (i.e., post text) could possibly be a better choice. To examine this possibility we replaced the resource titles from both datasets by a set of words extracted from the resource content. For clarity and consistency with related work we refer to the blog post text as a *document*. For a fair comparison to the title words we limited the number of words extracted from the content to five. Two ranking scores were used. Term frequency—inverted document frequency score (Tf-Idf, Eq. 6.1) promotes words with high number of occurrences n_w in the document and low relative number of documents d_w in the dataset D where it occurred. To capture the additional information about tag usage we modified the Tf-Idf score replacing the Idf part by the logarithm of the total number of occurrences of a word as tag — tag frequency t_w (Eq. 6.2). Tf-Idf score looks for specific terms that are unique for the document, whereas Tf-tf promotes general terms that are likely to be used as tags.

$$tf\_idf(w) = n_w \log \frac{D}{d_w} \tag{6.1}$$

$$tf\_tf(w) = n_w \log(t_w) \tag{6.2}$$

Surprisingly, despite the difference in the underlying assumption, both scores produced tags of similar quality (Table 6.2). Tf-Idf accuracy was slightly below, and Tf-tf slightly above, the accuracy of title words. Although the accuracy of content related tags based on Tf-tf tags is higher, it does not result in the improvement of the final recommendation. It suggests that the low quality of title based tags is in fact not caused by the title itself, but by the character of tags used for blog posts. Blog datasets tend to have very rich tag vocabulary. For example the *BlogSpot* dataset, despite being smaller, contains over ten times more distinct tags than the *Stack Overflow* dataset (Table 3.1). Over 60% of these tags were used only once. Most of them are tags composed of two or more terms, many of these are names. These characteristics suggest that utilization of resource content as a source of tags is possible,

Table 6.2: Recall@5 for three processing stages that utilize content information. Replacing the title with terms extracted from the resource content is not able to improve the results for blog posts.

dataset	title	content related	final recommendation
BlogSpot (title)	0.067	0.123	0.380
BlogSpot (content Tf-Idf)	0.063	0.125	0.375
BlogSpot (content Tf-tf)	0.077	0.136	0.375
WordPress (title)	0.056	0.112	0.458
WordPress (content Tf-Idf)	0.047	0.099	0.456
WordPress (content Tf-tf)	0.073	0.124	0.457

but would require an information extraction approach based on natural language processing techniques (e.g., key-phrase extraction).

6.2.2 Key-phrase Extraction for Tag Recommendation

To test the performance of content-based tags we focused on statistical key-phrase extraction methods used in *Maui* [51], an extension of a well-known key-phrase extraction system — *Kea* [18]. Kea maps the keyphrase extraction problem to a binary classification task. Each term or term phrase used in the content is classified as a potential keyphrase or a noise word. The scores provided by the classifier are used to rank the produced tags. Kea uses the Naive Bayes algorithm to classify phrases as good and bad candidates for keywords describing the document. Naive Bayes algorithm seems to be a suitable choice for content-based tag recommendation as well. Thanks to the simplicity of the models it generates, the system can be easily adapted to newly added posts. Kea's classifier is based on two features: tf-idf score of a phrase and the distance between the beginning of the document and the first occurrence of the phrase. Unlike the classification features, Kea's preprocessing steps are not applicable to the tag recommendation problem. Prior to classification, Kea removes proper nouns and creates phrases up to three words long. Proper nouns are often used as tags, so they should be considered as recommendations. On the other hand, despite multi-term phrases being sometimes used as tags (e.g., "'information_retrieval"') our preliminary experiments revealed low precision of this type of tag, so multi-term phrases have to be handled with special attention. An important additional feature

of Kea is the optional controlled vocabulary pruning. The user can provide a manually designed list of acceptable keyphrases. In such case only the keyphrases that can be found in the list are returned to the user. This extension was claimed to improve the accuracy of key-phrase extraction. However, one of the objectives of the content-based tag recommendation step in our system is to gain access to previously unseen tags, hence, we prefer to avoid this type of constraint. The basic Kea system was recently extended by Medelyan et al. [52, 51]. The first extension [52] uses the Wikipedia corpus as the source of controlled vocabulary. The system first filters the candidate phrases, accepting only ones that can be found as titles of Wikipedia articles. Wikipedia is also used as a source of additional phrase features. For example, one of the newly added features is *keyphraseness* — the ratio of occurrences of the phrase as a link anchor among all occurrences of the phrase in Wikipedia corpus. The underlying assumption is that descriptive phrases are likely to be used as links. The second extension [51] is an adaptation of the keyphrase extraction algorithm to an automatic tagging problem. The proposed algorithm *Maui* extends the list of Wikipedia based features and focuses on single-word terms. However, the underlying assumptions remain the same. The system was evaluated based on the frequently used tags only, hence its usefulness in the general tag recommendation problem is still unclear. The use of Wikipedia as a way to control the tagging vocabulary is an interesting approach; however, it has certain limitations. This approach is language dependent. In addition, despite the large number and broad scope of Wikipedia articles there is no guarantee that Wikipedia would cover the vocabulary of specific collaborative tagging systems. Given this constraints we decided to evaluate the accuracy of tags generated by Maui on the three narrow folksonomy datasets. In addition, we tested the performance of Maui as a component of our hybrid tag recommendation system.

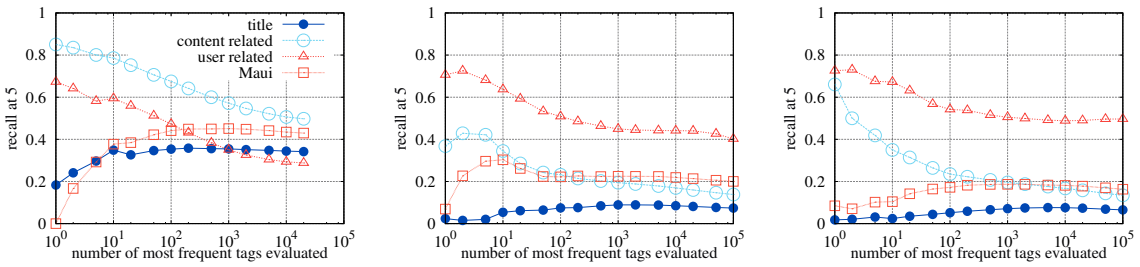
The recommendation accuracy results (Table 6.3) confirms that Maui is able to produce high quality recommendations that exceed the accuracy of title based tags. Nevertheless, Maui in all cases Maui was not able to outperform the strongest base recommender used in our system. For Stack Overflow dataset, the tags extracted from the full text of a post provide some improvement over the title-based tags, but Maui was not able to outperform the content-related tags based on the co-occurrence graphs. For the two blog datasets Maui was able to beat both title-based and content-related tags but was still outperformed by user-related tags. Although for all datasets

the results generated by Maui were worse by the strongest basic recommender used in our system, the addition of Maui as a component of our system significantly improved its performance.

Table 6.3: Recall@5 for the components of the system without and with Maui extensions. Boldface values point out the tag recommendation set with the highest accuracy. Although, in any case, Maui is not able to beat the top component of the basic system, adding Maui tags to the resource related tags improves the overall performance of the system (statistically significant, $P < 0.001$, Wilcoxon test).

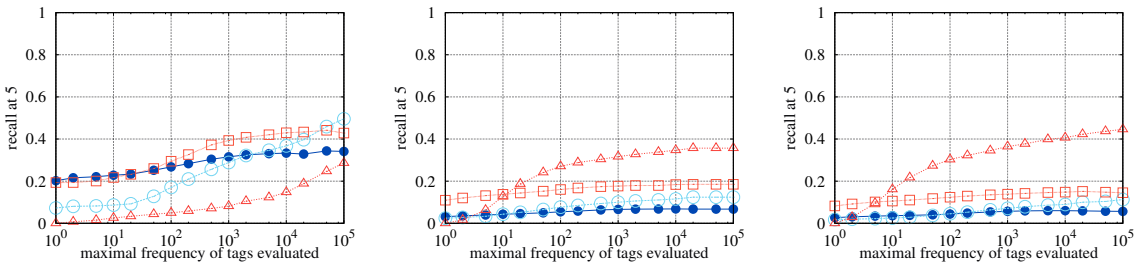
dataset	title	content related	user related	Maui	final rec. (no Maui)	final rec. (with Maui)
Stack Overflow	0.341	**0.495**	0.286	0.428	0.548	**0.587**
BlogSpot	0.067	0.124	**0.355**	0.185	0.382	**0.418**
WordPress	0.056	0.113	**0.443**	0.145	0.460	**0.474**

To gain more insights about the performance of Maui we repeated the experiment described in Section 6.1, looking at the relative performance of this system in comparison to the components of our system for the most frequently used tags. The comparison of the results for the few most frequent tags for all datasets shows that unlike content-related tags, Maui underestimates the importance of the frequent tags (Fig. 6.2). This problem is strengthen in Stack Overflow dataset (Fig. 6.2(a)) where a number of the most frequent tags (e.g., *c#*) are described in Wikipedia under different names. On the other hand the example of BlogSpot data (Fig. 6.2(b)) shows that Maui deals well with names of politicians and celebrities discussed in blogs (see Table 6.1).



(a) *Stack Overflow* dataset (b) *BlogSpot* dataset (c) *WordPress* dataset

Figure 6.2: Recall@5 for title, content related, user related and Maui tags for $N \in [1, 100000]$ most frequent tags only.



(a) *Stack Overflow* dataset (b) *BlogSpot* dataset (c) *WordPress* dataset

Figure 6.3: Recall@5 for title, content related, user related and Maui for tags with overall frequency at most $N \in [1, 100000]$.

Nevertheless, the rationale for using key-phrase extraction method in tag recommendation problem is gaining access to low frequency tags, for which we are unable to build recommendation models based on the prior tag use information. To evaluate the ability of Maui to access potentially useful low frequency tags, we modified the experiment observing the performance of the system for tags with at most N occurrences in the overall dataset. It means, that for $N = 1$ we tested the accuracy of the system in recommending tags that overall were used only once. For Stack Overflow dataset Maui is not able to outperform title-based tags (Fig. 6.3(a)). It suggests that users of Stack Overflow wants to make sure that the specific problem they are addressing is stated both in the title and among tags. However, for both blog datasets Maui is clearly the most accurate source of low frequency tags. In fact, unlike the components of our system, its performance does not seem to be correlated with the overall frequency of recommended tags.

The ability of Maui to recommend tags of very low frequency makes it a useful addition to our tag recommender for tagging systems in which the textual content of the resource is easily available. Thanks to relatively simple process of feature extraction and the learning model Maui is also able to scale to datasets of practical size. The only potential obstacle are features based on the Wikipedia information; however, these features can be precomputed off-line prior the recommendation process.

6.3 Parameter Learning Based on Tagging Patterns

The basic version of our system uses a single set of merge coefficients for all tested posts. The parameters are tuned so, on the average, they produce the most accurate results. We refer to this approach as *global tuning*. The main drawback of the global tuning approach is the underlying assumption that all posts in a particular collaborative tagging system would require the same settings of merge coefficients. On the contrary, it is likely that each processed post has a specific range of merge coefficient values, for which the most accurate combination of two input sets can be produced. The accuracy is likely to be increased if, instead of using a global set of merge coefficients, the system would be able to predict their optimal values for a particular post. Clearly, inferring the optimal values of merge coefficients for each post is not likely to be possible or practical. A more realistic scenario is creating a set of *tagging patterns* that would represent a group of posts that share some characteristics and should be processed using the same set of merge coefficients. The posts that belong to the same pattern can be used to train pattern specific values of merge coefficients. The most intuitive extension of parameter learning is utilization of the personal character of tagging. As users tend to re-use the same set of tags they could also have specific tagging patterns that would favour one of the input sets of a given merger (e.g., resource related tags over user related tags in the system's final merger). Therefore, a basic attempt to utilize tagging patterns can be personalized parameter tuning. At the same time, we can assume that the personal patterns of users can be generalized into user type patterns. This intuition is supported by a study by Körner et al. [41]. The study shows that there are two types of taggers, categorizers and describers, who prefer different recommendation techniques. Finally, we can look for specific tagging patterns within the tagging actions of a single user. For example a user can have two different patterns for tagging her references to scientific literature. References related strictly to her thesis topic are tagged with a small set of personal tags so they are well organized. On the other hand, references to papers that she generally finds interesting but has no time to read now are tagged with a large set of unique descriptive tags so they can be easily searched for later. It is clear that the two types of resources would require different sets of merge coefficients to generate recommended tags. The distinction between them can be done based on the topic of the resources as well as the temporal characteristics of posts. The

Table 6.4: Recall@5 for the final recommendation with two parameter learning approaches. Global tuning uses one set of parameters for all posts. Personal tuning learns parameters for each user individually. Personalization is able to increase the accuracy of recommendation, yet the improvement is not satisfactory in comparison to the upper bound (perfect prediction).

dataset	global tuning	personal tuning	**increase**	perfect prediction	**increase**
BibSonomy	0.379	0.393	0.014 (3.7%)	0.486	0.107 (28.2%)
CiteULike	0.435	0.449	0.014 (3.2%)	0.539	0.104 (23.9%)
Delicious	0.447	0.463	0.016 (3.6%)	0.546	0.099 (22.1%)
Stack Overflow	0.548	0.551	0.003 (0.5%)	0.624	0.076 (13.9%)
BlogSpot	0.382	0.387	0.005 (1.3%)	0.427	0.045 (11.8%)
WordPress	0.460	0.467	0.007 (1.5%)	0.500	0.040 (8.7%)

latter is supported by Yin et al. [72], who demonstrated that utilizing the session-like behaviour of tagging system users can improve the accuracy of tag recommendation.

To evaluate the usefulness of tagging patterns in tag recommendation problem we designed a tagging pattern mining module which objectives were to distinguish various tagging patterns, map the incoming post to one of the patterns and use the pattern specific set of parameters to process the post. To keep the simplicity of the system we assumed that the module should work in completely unsupervised manner. The recognition of different patterns and post-to-pattern assignment should be done based on the information that is already present in the system (previous tagging history, attributes of the incoming post), with no need for additional interaction with the user.

6.3.1 Potential Usefulness of Tagging Patterns

Given the real tags, we were able to simulate the ideal scenario of choosing the values of merge coefficients that produce the most accurate tags for each post. We re-ran the experiments for each dataset assuming the *perfect prediction* of merge coefficients for each processed post. The quality of results ("perfect prediction" in Table 6.4) show a wide range of potential improvement, compared to the global tuning approach. Unfortunately, choosing the set of optimal merge coefficients for each particular post is not practically feasible. A simple and intuitive approach to include tagging patterns into the learning process is personalization. We can create a separate tagging pattern for each individual user.

We modified the system to train the merge coefficients separately for each user (given that the number of processed user's posts exceeded the threshold of 10). In this case the feedback processor tuned the parameters for both train and test posts, to make personalization possible for new users. For each new user found in the test set we first used the global parameters and gradually switched to the personalized parameters as additional posts of the user became available. As discussed in Section 4.2, the system is designed for online learning so the only extension needed is the storage of personal parameters. Personalized parameter training was able to improve the quality of recommended tags. However, the improvement was lower than expected. We found three potential explanations of this outcome. First, the tagging pattern of a user is to some extend already embedded in the recommendation process. For example, a user who often re-uses personal tags increases the scores of user profile recommender results, giving it an advantage over resource related tags when these two sets are merged. Second, user tagging patterns are likely to depend on other factors (e.g., type of a resource), hence more complex methods are needed to extract and represent users' tagging patterns. We find evidence for the second hypothesis observing the merge quality curves for specific users. In most cases, they have the same characteristics as the global curve. In addition, the comparison of the number of correct tags found among resource related and user related tags revealed that, even when a user prefers one of the tag sources, the other also contains a significant amount of correct tags. Third, constraining the training process to the posts of a single user results with a large number of tagging patterns for which we do not have enough training data. As a result, personal tagging patterns may not generalize well to future user's posts.

6.3.2 Tagging Pattern Processing Module

To examine further the problem of tagging patterns in tag recommendation we designed a tagging pattern processing module. In the previous experiment (personalization) we picked the tagging patterns based on the common feature shared between posts (user ID). The evaluation of this approach revealed that it does not create the expected improvement, because the tagging patterns in fact resulted with similar parameter tuning settings. To avoid this pitfall, the design objective of the tagging pattern module was to first come up with a small set of abstract patterns which

gather posts of different tagging characteristics and later try to infer the common features of the posts gathered in a single pattern. The process can be divided into three sub-problems: discovery of tagging patterns (clustering task); representation of a tagging pattern (feature selection task); finding the pattern based on post features (classification task). The module replaces the merging component of the original system. Instead of using a single set of parameters the module trains independent set of parameters for each of N tagging patterns. For each incoming post it uses the parameters tuned for a single tagging pattern, given the assignment of the post to the pattern (Fig. 6.4). The assignment is done by matching the features of the incoming post with the features of posts for which a specific pattern had top accuracy among all patterns. This task can be solved using classification or regression algorithms.

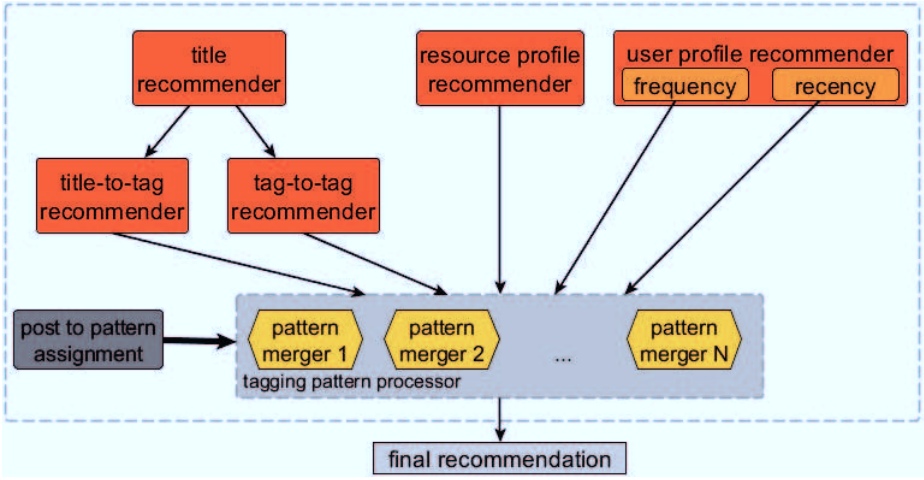


Figure 6.4: Tagging pattern processing module in the tag recommendation system scheme. The module trains merging parameters for N tagging patterns and processes the post with one of the trained sets. The choice of the pattern is done based on the post features.

Discovery of tagging patterns Each post is likely to have its own set of parameter values that produces the optimal recommendation given the tags from various sources. To be useful in a learning process these posts have to be generalized to a tagging pattern (i.e., grouped into sets of posts that need similar parameter values). This task can be considered as a clustering problem. To solve it we used the adaptation of Lloyd's algorithm, which is the commonly used approach for k-means problem. The input of the algorithm is the set of training posts and the expected number of tagging patterns N. The algorithm can be represented with the following steps:

1. Given a set of training posts, randomly assign them to N patterns.
2. Train the merging parameters for each pattern.
3. Evaluate the performance of each pattern on the training posts.
4. Reassign each post to the pattern with highest accuracy score.

Steps 2-4 are iteratively repeated until convergence. To limit the processing time we assumed that the algorithm converges when less than 5% of posts change its pattern. Following the previous experiments we used recall@5 as the accuracy score. However, when calculated for a single post recall@5 has a small discrete set of maximally six possible values. As a result, many patterns can produce tag recommendation sets of identical accuracy. To smoothen the range of the results we combined recall@5 with average precision score. We used a linear combination, with parameter β, which controlled the impact of the AP factor. The parameter value was set to 0.01, so AP value was effectively taken into consideration only if the value of recall@5 was identical for two or more patterns. The output of the algorithm is a set of N tagging patterns which are optimized for a subset of the training posts. Just like in k-means algorithm, the posts of similar tagging characteristics are likely to gather together in a single pattern.

Representation of a tagging pattern We decided to represent a tagging pattern with a set of features extracted from the posts assigned to it in the pattern discovery step. This way we can easily match the incoming posts with the patterns to decide which pattern should be used to process a given post. The features used in the system can be divided into four categories:

cat/desc We used a set of features proposed by Körner et al. [41] to discriminate between two types of users — categorizers and describers, who prefer different recommendation techniques. Categorizers tend to re-use a coherent set of tags for many resources, keeping their tag vocabulary small. The repositories made this way are more suitable for tag-based browsing. Describers attach a set of rare but very descriptive tags to a resource, creating verbose vocabularies. This tagging approach is more suitable for tag-based search. The distinction between categorizers and describers can be easily represented in our tag recommendation

system. For example a user who tends to re-use the same set of tags can have a specific tagging pattern that would favour user related tags over resource related tags in the system's final merger. We used four features proposed by by Körner et al. to differentiate categorizers and describers:

- Tag/resource ratio (trr), which is the number of unique tags in user's tag vocabulary divided by the total number of resources posted by the user.
- Orphaned tag ratio (otr), which captures the ratio of tags in the vocabulary which are not re-used (orphans). A tag is considered an orphan if it was used only once or it was used less than 100 times comparing to the most frequently used tag in the user's vocabulary.
- Overlap factor (of), although the formulation of the overlap factor score is more complex, in principle it represents the average number of tags per post, scaled so it is always in $[0, 1]$ range. As there is no need for scaling in our application, we decided to use the simpler representation.
- Title/tag intersection (ttr), which is the ratio of all terms that were used in the titles of all resources posted by the user and were used as tags among all the title words. The score can be expressed with a formula: $ttr = \frac{|V_{title} \cap V_{tags}|}{|V_{tags}|}$, where V_{title} is the vocabulary of terms that occurred in the title of user's resources and V_{tags} is the vocabulary of user's tags.

For all the features we can hypothesize that a typical categorizer would have a lower score than a typical describer.

session To capture the session behaviour of users we used two features: the time difference and the overlap between the title terms:

- Time difference is the difference (counted in seconds) between the currently processed post and the last posts of the same user.
- Title overlap is the total number of common terms from the title of the currently posted resource and the title of the last resource posted by the user.

Low time difference and high title overlap may suggest that the posts were added in the same session.

scores There are certain factors that can impact the choice of the tagging patterns that are related to the tag sources available for the given post rather than the behaviour of the user. Examples of such factors can be poorly formulated title that contains only terms with low usefulness as tags or profile of a sparsely tagged resource in which the top tag has a perfect score, but its reliability is low as it was calculated based on a small number of posts. To represent such factors we designed a set of features that are calculated based on the scores of the tags returned by each of the base recommenders, namely the title recommender, title-to-tag and tag-to-tag recommenders as well as the recommenders based on profiles of the resource and user. We calculated three features for each of the recommenders:

- Top score, which is the score of the top ranked tag from the source.
- Score at 5, which is the score of the tag at the fifth position in the ranking
- Standard deviation, which is the standard deviation of the scores of the first five tags from the ranking.

previous Finally, we can directly represent the accuracy of the tagging pattern for a given user or resource by aggregating the information about the previous performance of the pattern. We used a set of features to represent the average score of the pattern for all previous posts of a user who is creating the post or resource that is currently posted. Therefore, the number of features is equal $2N$, where N is the number of tagging patterns. If any of the patterns is obtaining higher scores for the posts of a given user or resource it will be represented by the higher value of its feature.

Finding the pattern based on the post features Each tagging pattern can be represented as an aggregation of training posts, for which the pattern obtained the most accurate recommendation results. This setting can be mapped to a classification problem in which posts are labelled instances and class labels represent tagging patterns. The objective of the system is then to assign an incoming post to one of the classes (patterns). An alternative representation is to consider the post-to-pattern assignment as an regression problem. Given the performance of all the training posts for a given pattern we can build a separate regression model for each pattern. This

way we can try to predict the performance of each pattern for the incoming post and choose the pattern with the highest predicted accuracy. We decided to experiment with both approaches.

6.3.3 Tagging Patterns — Evaluation

In the evaluation of the tagging pattern processing module we were interested in three main factors: (1) the ability of the pattern discovery algorithm to separate posts into patterns of different characteristics, (2) the performance of the classification or regression algorithms used to assign a post to a pattern and (3) the importance of feature categories in the assignment process.

Experiment design Some of the features used in the experiment rely on the previous information about the resource in the incoming post. Therefore, to be able to evaluate all features and reduce the impact of data sparsity we decided to use only the broad folksonomies pruned to p-cores. To balance the sparsity and the number of posts, we used 2-cores pruning for BibSonomy, 4-cores for CiteULike and 70-cores for Delicious. The pruning approach and the separation between training and test instances followed the one used in previous experiments (see Section 5.2 for details). The tagging pattern module allow the use of any parameter tuning approach. In the experiments, we used the parameter tuning module based on SVMrank algorithm. This way we wanted to make the results more generalizable, as SVMrank was successfully evaluated in other tasks [36, 4]. The evaluation of the tagging pattern processing module can be divided into three parts, here we introduce the experimental design for each of the parts:

- **Discovery of tagging patterns.** The pattern discovery algorithm takes as an input the number of result patterns. To evaluate how the choice of the number of patterns impacts the performance of the system we re-run the algorithm for a range of pattern numbers. The patterns were extracted using all the posts from the training set. The evaluation was performed on the test posts, assuming the perfect assignment of a post to a pattern. To achieve it, given a post, we evaluated all the patterns and picked the pattern with the best accuracy of recommended tags. Based on the evaluation we picked a constant number of tagging patterns ($N = 4$) that was used in all following experiments.

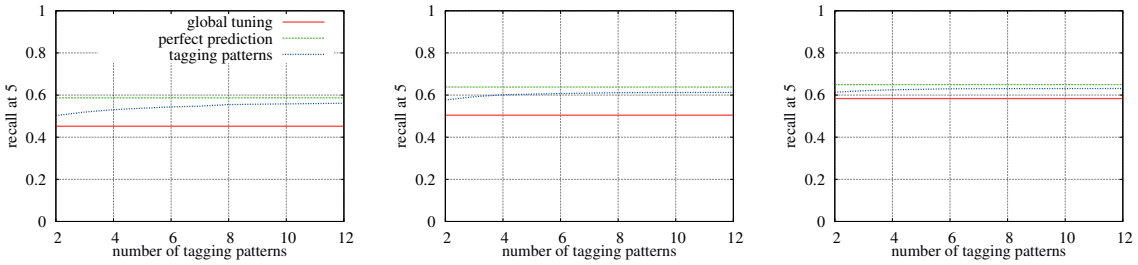
- **Post-to-pattern assignment methods.** As discussed above, the task of assigning an incoming post to one of the extracted tagging patterns can be mapped to the classification or regression problem. We tested a range of classification and regression approaches available in Weka toolkit [24]. We extracted the post features for all posts from the training and test set. In the classification setting, we labelled the posts from the training set with the ID of a pattern which is able to recommend the most accurate tags for it. Later we classified test posts assigning each post to a single pattern. In the regression setting, each post from the training set was assigned the accuracy score computed for each pattern. Later, a predicted accuracy score was calculated for each test post, pattern pair. The posts were assigned to the pattern with the highest predicted score. Given the post-to-pattern assignment we computed the aggregated accuracy score for all test posts considering only the recommendation outcomes from the assigned patterns.

- **The impact of the feature categories.** To evaluate the impact of the features on the post-to-pattern assignment we have repeated the previous experiment removing a single group of features at a time. The difference between the performance of the system with the full set of features and the system without specific features signifies the importance of these features in the post-to-pattern assignment process.

Each experiment was repeated ten times with different initialization of the pattern discovery algorithm. We report the mean recall@5 score calculated for all test posts and averaged over the ten runs.

Discovery of tagging patterns Assuming the perfect performance of the post-to-pattern assignment we can calculate the maximal potential gain in the accuracy using the parameter settings for N patterns instead of a single parameter set tunned for all posts (global tuning). As we can observe, the use of even two patterns gives visible increase of the accuracy (Fig. 6.5). The accuracy grows with the number of patterns but the scale of improvement decreases quickly and the performance seems to stabilize soon. A large number of tagging patterns increases the difficulty of the post-to-pattern assignment task. Therefore, we decided to keep the number of patterns low for the further experiments. We picked the number of patterns $N = 4$ for all

the following experiments. Although the number of patterns is relatively low, for all datasets, this choice makes the potential accuracy gain much closer to the perfect prediction score, which is the hard limit of the performance improvement, than the results of global or personal tuning.



(a) BibSonomy (b) CiteULike (c) Delicious

Figure 6.5: The potential accuracy improvement for increasing number of tagging patterns. The accuracy grows with the number of patterns but the scale of improvement decreases quickly and the performance seems to stabilize soon.

Post-to-pattern assignment methods In preliminary experiments we have compared a wide variety of classification and regression approaches and selected three with the top performance: Two classification algorithms: REPTree and Support Vector Machines (SVM) and linear regression. REPTree is a tree-based approach, a modification of a well known C4.5 algorithm. The selection of the features in REPTree is done based on information gain criterion and the tree is pruned using reduced-error pruning. As a baseline we used ZeroR classifier, which assigns all test instances into the most frequent class in the training set. Both classification approaches outperform the baseline and at the same time are outperformed by the linear regression (Table 6.5). Nevertheless, for all datasets, the results of linear regression are much worse than the optimal assignment. It suggests that none of the features represent well the distinction between the patterns.

The impact of the feature categories To look closer at the impact of the features on the post-to-pattern assignment we repeated the experiments removing particular categories of features and observed the decrease in the accuracy of the recommendation. The results of the experiment suggest that the impact of specific features depends on the dataset (Table 6.6). For BibSonomy, the largest decrease in the performance is observed after the removal of features that discriminate categorizers and

Table 6.5: Comparison of post-to-pattern assignment methods. Boldface value points out the most accurate method. For all datasets linear regression outperforms the classification based methods, nevertheless its performance is much worse than the optimal assignment of the posts.

dataset	ZeroR	REPTree	SVM	L. Reg.	**optimal**
BibSonomy (2-cores)	0.420	0.442	0.460	**0.465**	0.531
CiteULike (4-cores)	0.414	0.457	0.470	**0.502**	0.601
Delicious (70-cores)	0.535	0.574	0.578	**0.585**	0.625

describers. For CiteULike, only *session* features make visible difference. For Delicious, the features seem redundant as the removal of any of the categories of features make little impact on the results. In all cases, we can observe that the removal of *score* features does not change or even improve the performance of the system. We can conclude that these features are completely uncorrelated with the choice of the tagging pattern.

Table 6.6: The impact of feature selection on the accuracy of recommendation. Boldface value points out the features category which results in the highest accuracy of final recommendation. The selection of the most useful features depends on the dataset.

		without features:				**all**
dataset	ZeroR	cat/desc	session	scores	previous	**features**
BibSonomy (2-cores)	0.420	**0.458**	0.463	0.466	0.462	0.465
CiteULike (4-cores)	0.414	0.497	**0.487**	0.502	0.500	0.502
Delicious (70-cores)	0.535	0.584	0.584	0.586	**0.583**	0.585

Overall performance Finally, we have compared the results of the system with the tagging patterns module to the two basic approaches of global and personal parameter tuning (Table 6.7). It turns out that even though the discovered patterns have large potential in the improvement of the recommendation accuracy ("patterns optimal"), poor assignment of test posts to patterns makes their performance ("patterns regression") slightly worse than a much simpler approach based on personalized learning ("personal tuning"). All the tested features do not provide enough information to assign a post to the proper tagging patterns. Therefore, we should look

for more complex features that are likely to be related to specific characteristics of a tagging system or a specific user. However, in the latter case, it is possible that the system does not contain enough information to determine the different types of user behaviour and discriminate posts that are represented by the patterns.

Table 6.7: Comparison with global and personal parameter tuning approaches. Boldface value points out the approach with the highest accuracy. Even though tagging patterns have the potential to improve the results of recommendation (patterns optimal), poor results of post to tag assignment step makes this solution slightly worse than simpler personal tuning approach.

dataset	global tuning	personal tuning	patterns regression	patterns optimal	**perfect prediction**
BibSonomy (2-cores)	0.452	**0.467**	0.465	0.531	0.587
CiteULike (4-cores)	0.505	**0.508**	0.502	0.601	0.638
Delicious (70-cores)	0.583	**0.586**	0.585	0.625	0.649

Chapter 7

Conclusions and Future Work

Tag recommendation is an interesting and practical research problem, which has the potential to substantially increase the usefulness of collaborative tagging systems. The objective of a tag recommendation task, as seen today, is to predict a small set of tags that a user will find useful in describing a resource stored in a collaborative tagging system. Therefore, in a broader perspective, the design of a tag recommendation system should be based on a good understanding of how users perceive and use tags. In general, tag recommendation can potentially reveal many interesting properties of collaborative tagging systems. At the same time, every bit of information about why and how users tag has the potential to lead to better tag recommendation techniques. This relation between understanding of the data and utilization of its characteristics in the design of a tag recommendation system was the base point of our work. In our work, we analyzed in detail the characteristics of tagging datasets. Insights about the characteristics of the problem allowed us to determine a set of potential tag recommendation sources (e.g., resource title, resource and user profile) and tagging behaviour types (e.g., re-use of recent tags by users) as well as the potential pitfalls (e.g., data sparsity problem, low usefulness of collaborative tagging approach for tag recommendation and the disturbance in the dataset characteristics caused by p-cores pruning). The findings led to a simple but effective hybrid tag recommendation system. The quality of the system was confirmed in the ECML/PKDD Discovery Challenge 2009, where an early version of the system achieved top place in two tag recommendation tasks, including online evaluation, in which the system was evaluated based on real user feedback. The evaluation on six real-life datasets, presented in the thesis, confirmed the practicality of the proposed hybrid approach. The detailed experiments evaluation of the system's effectiveness allowed us to gain more insights about the characteristics of tagging systems. We found that users of broad folksonomies and blogging systems have strong interest in re-using their personal tags.

For broad folksonomies re-use of tags has a strong temporal component, whereas bloggers tend to repeatedly use a predefined set of abstract categories. Conversely, users of a programming forum (Stack Overflow) aim to make their tags descriptive and easily findable for the community. This results in a high overlap of tags with the content of their posts. Despite the differences, for all datasets the proposed hybrid tag recommendation approach was able to adapt to the tagging style of the users.

7.1 Main Contribution

The objective of our work was to determine the practical aspects of the tag recommendation problem both coming from the properties of the collaborative tagging data as well as the fact that tag recommendation is done with a close interaction with a user. Based on these we defined the six requirements of a tag recommendation system. The main contribution of the thesis is the conceptual design, architecture and detailed evaluation of a hybrid tag recommendation system that meets these six requirements of practical tag recommendation:

Data sparsity. Thanks to a combination of tags from various sources, including resource content and user profile, the system is able to recommend tags for virtually every post. Unlike many tag recommendation approaches, which focus only on the frequently used tags and resources, the system achieves satisfactory accuracy of recommendation without any constraints on the processed data.

Open-ended vocabulary. Our system puts no constraints on the set of tags considered during the recommendation process. Content based tags and user feedback give it constant access to previously unseen tags.

Generality. The evaluation on a broad range of datasets demonstrated that, despite different characteristics of collaborative tagging systems, our recommendation system is able to automatically determine and utilize the most accurate tag sources.

Adaptability. The use of simple but effective recommendation models allows the system to instantly adapt to newly added posts, which is especially important considering the tags retrieved from the user profile.

Efficiency. Thanks to an architecture based on a text indexing engine and overlaying cache layer the system is able to serve, in real time, collaborative tagging systems of practical size.

Low maintenance cost. The system utilizes the constant interaction with users to automatically train its parameters. No input parameters or periodic retraining of the models are necessary.

The development of a fully operational tag recommendation system which is able to serve a broad range of collaborative tagging systems allowed us to find answers to four research questions, which we mapped to the tag recommendation problem from the general area of recommender systems:

Research question 1 *What is the practical usefulness of various tag sources in the tag recommendation problem?*

Experiments on the characteristics of collaborative tagging system data allowed us to identify several tag sources that are potentially useful in tag recommendation process. We demonstrated the limitations in the usefulness of techniques based on collaborative filtering and we decided to focus on tag sources that are directly related to the post for which the tags are recommended. Among these sources we can point out the resource title, the tag profile of a resource and the tag profile of a user. We found that the words found in the title of a resource are likely to impact the tagging decisions of users. As the set of tags extracted from the title is limited, we proposed an extension method based on the tag co-occurrence graphs. The tag profiles of resources and users are characterized by heavy-tailed distribution, in which a small set of tags occurs much more frequently than others. These tags are likely to be re-used in the future, which makes them a potential source of tag recommendations. Observation of the dynamics of the formulation of resource and user profiles presented in the literature [71] suggests a strong temporal character of the tag use by users. To leverage this information we proposed a recency ranking scheme, which recommends the most recent tags of a user. Although some previous work suggests that a similar approach can be applicable to resource profiles due to collaborative agreement on the tags used for a given resource that is formed with time [22], we found no confirmation for this hypothesis. Considering specific characteristics of blog datasets we looked for

an additional source of tags that would leverage the textual content of the post. We demonstrated that tags extracted from the blog post text using key-phrase extraction techniques can be a valuable source of tags.

The evaluation of the tag recommendation system based on the proposed tag sources confirmed their usefulness in the recommendation process. At the same time, the comparison between datasets from various tagging systems revealed that the accuracy of a tag source is greatly dependent on the characteristics of the dataset. The resource title is most useful for systems with highly descriptive title and tags (e.g., Stack Overflow, BibSonomy, CiteULike). The resource profile has the potential of being the most accurate source of tags in broad folksonomy systems, given a sufficient number of previous posts with the resource stored in the system (e.g., Delicious). When the sparsity of posts is higher, the most accurate source is the user's profile (e.g., BibSonomy and CiteULike). In all broad folksonomy systems the recency of tag use has greater impact on the accuracy of tags retrieved from user profile than the frequency. This results suggest the temporal character of user interests. On the contrary, the frequency of tag use has much greater importance in blog datasets (e.g., BlogSpot, WordPress), where users seem to keep a constant set of topics they write about or use tags that are applicable only to the current post.

Research question 2 *What is the importance of heavy-tail and long-tail elements in the tag recommendation process?*

Most tag recommendation systems focus on the posts for which the information needed in the recommendation process can be easily found in the training dataset. In case of broad folksonomies a pruning approach based on p-cores extraction is used to densify the dataset and provide sufficient amount of information for all tested posts. We demonstrate that the use of p-cores pruning has great impact on the dataset as it leaves only a small fraction of tag assignments that are originally present in the dataset. In practice, it means that the recommendation system is able to serve only a small set of posts. Nevertheless, as p-cores pruning is a commonly used approach, we decided to run a comparative study with state-of-the-art approaches designed specifically to work with the frequently occurring elements on datasets with different levels of pruning. We demonstrated that our system is also able to adapt to the densified datasets and achieves comparable or better results for higher p, unless

the dataset used for training the system is very small. This is possible because three elements of the hybrid system (i.e., extension of title tags based on tag co-occurrence graphs and profile based recommenders) achieve the highest accuracy of recommendation for the most frequent tags. In such case, the real challenge is the recommendation of infrequently used tags, with insufficient information about the resource and user. The experiments for low values of p confirmed the robustness of our system, which for all datasets achieved significantly better results than competitors. It was possible thanks to the content-based tags; namely, the title-based recommender which is a default element of our system. The performance of the content-based tags can be improved using a tag recommendation approach based on key-phrase extraction (Maui [51]). We demonstrated that key-phrase extraction can be especially useful in blog datasets where large number of tags are proper names of people, places or events mentioned in the text of the post.

Research question 3 *Is a hybrid tag recommendation system that relies on several possible tag sources able to adapt to tagging style used in a specific tagging system?*

As mentioned earlier the accuracy of a source of recommended tags depends on the character of a tagging system. Nevertheless, the proposed parameter tuning approach, based on the merge quality curve, is able to combine the tags from various sources so the performance of the overall result is always significantly better than the best of the base recommenders. The proposed approach achieves similar results as state-of-the-art ranking system based on Support Vector Machines. However, it has some advantages over the competitor. It has much lower memory use thanks to stream processing instead of batch processing approach. It can be updated iteratively if the conditions of the recommendation change. Furthermore, it is easy to modify and extend.

Research question 4 *Can the feedback loop in the recommendation process be utilized to improve the quality of the recommended tags?*

The feedback loop gives the system access to the most recently used tags. In our system we used this feature to do online adaptation of the tag profiles content. In this way tag co-occurrence graphs and the tag profiles of resources and users always contain the information based on the most recent tagging activity. One of

the advantages of this approach is the simplification of system maintenance: the system needs no periodical retraining. Furthermore, as we demonstrated, online content adaptation significantly improves the quality of recommended tags for all tested datasets. It is especially visible for broad folksonomies that highly depend on the tags retrieved from the user profile (e.g., BibSonomy, CiteULike). In these systems, users often enter their posts in sessions re-using the tags within a short time span. Online content adaptation gives the recommendation system access to these tags.

7.2 Additional Contributions

The introduction of the tag recommendation problem and experimental results presented in the thesis offer insights related to previous research in tag recommendation. In this section we summarize some additional contributions of the thesis that complement the objectives of the work:

Overview of tag recommendation evaluation techniques. The concept of tag recommendation is used to describe a broad area of systems which objective is to retrieve a set of descriptive tags for a given resource. However, the specific objectives and features are viewed differently by different communities of researchers. These differences are easiest to notice while comparing the evaluation approaches. Therefore, to position our system in the broad scope of tag recommendation systems we classified and summarized the main features of tag recommendation system evaluation techniques. We discuss the advantages and disadvantages of the evaluation decisions proposed in the literature and based on that we propose an off-line evaluation approach that in our opinion represents the most realistic scenario considering the practical use of the tag recommendation systems.

Evaluation of PITF and FM methods for various graph density. The tag recommendation systems based on Pairwise Interaction Tensor Factorization and Factorization Machines are state-of-the-art approaches in graph-based tag recommendation. Their authors focused the evaluation on the comparison of the effectiveness and efficiency of their systems to other graph-based tag recommendation techniques. In our work, we extend the comparison to our hybrid

tag recommendation approach. In addition, we discuss the performance of the systems considering datasets of different size and density. The evaluation suggests that PITF and FM work best for small and dense datasets, which makes their practicality questionable. The size of the dataset seems to be especially important for the Factorization Machines approach which is significantly outperformed by PITF for the largest dataset used in the study — Delicious.

Evaluation of Maui for infrequent tags. The main objective of Maui as indicated by its authors is the automatic topic indexing of documents. As the system output is identical with the output of a tag recommendation system and it was evaluated on a tagging system — CiteULike, we can also consider Maui as a tag recommendation technique. Due to the fact that the authors focused on topic indexing task, the evaluation of the system was performed based on frequent tags which were assigned to a resource by many taggers. In our work, we focused on the evaluation of the system based on the infrequent tags used only a few times in the entire dataset. These tags do not generalize the content of the resource, therefore they cannot be considered as good topics. Nevertheless, we found that Maui is robust enough to successfully extract this type of tags from the content of the resource. Conversely, we found that it has more difficulties in extracting the most general tags in certain types of datasets, where the tags do not match article names in Wikipedia.

Evaluation of post features in tag recommendation task. Various researchers suggested that utilizing features like the type of a user [41] or session-like posting behaviour [72] can have positive feedback on the tag recommendation results. We have evaluated these features in the context of a hybrid tag recommendation system, where the features can be used to choose a specific parameter settings for a given post. The experiments with tagging patterns demonstrated that these features to some extent are able to discriminate between posts that should be processed with different parameter settings. However, the discrimination abilities of the system based or provided features are of low quality and they do not lead to a useful improvement in the recommendation accuracy. We find two explanations of this fact. First, these features are to some extend

already embedded in the scoring systems of our base recommenders. For example, in-session posts have short time distance, which results in higher scores of recency-based scoring scheme of user profile recommender. On the other hand, the good performance of the tagging patterns with the optimal post-to-pattern assignment suggests that there are important differences between posts, which are too subtle to be captured by the rather simple features used in our experiments. It creates an interesting task for future work on the system — the discovery of post features that would result in better post-to-pattern assignment.

7.3 Future Work

One of the main advantages of the tag recommendation problem is accessibility of large-scale, real-life data from a broad range of tagging systems. Provided with just the actions of users, we can train the recommendation system to predict their future tagging decisions. This concept of the tag recommendation process works under the assumption that the proper set of tags is provided by the user in the moment of entering the post to the system. This assumption is shared in the development and evaluation of a great majority of tagging systems presented in the literature. Tagging is a cumbersome process and users are likely to minimize the time spent on it, which hurts the quality of their tags. In addition, at the moment of tagging the users have limited abilities to predict which of the tags will be useful to the retrieve the resource in the future. As a result, tag recommenders trained on the previous user actions are not able to directly address the main purpose of tags which is to provide the access to resources through browsing, search or filtering. The objective of our future work is to change the paradigm of the task and focus on tag-based retrieval as the ultimate goal of tag recommendation. The general usefulness of a tag is likely to depend on the way it is used. Following the categorizer/describer classification proposed by Körner et al. [41] we can come up with two tagging approaches. The filtering task, in which the user sets up a set of interesting tags as the notification triggers to receive only interesting resources, would require a concise and well-formed tag vocabulary. The development of such a vocabulary can be observed in the community of Stack Overflow users. In this system, only the experienced users can propose new tags and the tagging vocabulary is a constant object of discussion of the community. A

tag recommendation system in such case should take into account the coherence of the tag vocabulary limiting the amount of tag synonyms and infrequent tags. To serve this purpose we plan to further exploit the tag co-occurrence graphs, currently used to extend the set of tags extracted from the resource title. The graphs provide contextual information about tags, which can be used for synonym detection following the techniques from text mining. On the other hand, the search task would require a verbose set of tags that provide a detailed description of the resource. In this case the tag recommendation system should come up with a large number of tags including synonyms and specific tags. An example of a system that would benefit from the use of such a tag recommendation system is Flickr, which allows users to store and share photographs and short videos. These resources do not contain rich textual content, so they are hard to access through keyword based search. In general, tags can help in the search of resources only if they do not overlap with the keywords that can already be found in the text of the resource. To get access to such tags we plan to explore the relations between resource features and specific types of tags assigned to resources. Examples of such features are: visual features (in Flickr), document features (in CiteULike), musical features (in Last.fm).

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17:734–749, June 2005.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 217–253. Springer, 2011.

[3] V. Batagelj and M. Zaveršnik. Generalized cores, 2002. cite arxiv:cs.DS/0202039.

[4] Fabiano Belém, Eder Martins, Tatiana Pontes, Jussara Almeida, and Marcos Gonçalves. Associative tag recommendation exploiting multiple textual features. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1033–1042, New York, NY, USA, 2011. ACM.

[5] M. Bender, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, R. Schenkel, and G. Weikum. Exploiting social relations for query expansion and result ranking. In *Data Engineering for Blogs, Social Media, and Web 2.0, ICDE 2008 Workshops*, pages 501–506, 2008.

[6] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. MOA: Massive online analysis, a framework for stream classification and clustering. In *Journal of Machine Learning Research (JMLR) Workshop and Conference Proceedings, Volume 11: Workshop on Applications of Pattern Analysis*, pages 44–50. Journal of Machine Learning Research, 2010.

[7] Dirk Bollen and Harry Halpin. The role of tag suggestions in folksonomies. In *HT '09: Proc. the 20th ACM Conference on Hypertext and Hypermedia*, pages 359–360. ACM, 2009.

[8] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Rev. Mod. Phys.*, 81(2):591–646, May 2009.

[9] Ciro Cattuto. Semiotic dynamics in online social communities. *The European Physical Journal C - Particles and Fields*, 46(0):33–37, August 2006.

[10] Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Semiotic dynamics and collaborative tagging. *Proc. the National Academy of Sciences (PNAS)*, 104(5):1461–1464, January 2007.

[11] Weiwei Cheng and Eyke Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2):211–225, 2009.

[12] Paul Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Siegfried Handschuh. P-tag: large scale automatic generation of personalized annotation tags for the web. In *WWW '07: Proc. the 16th International Conference on World Wide Web*, pages 845–854. ACM, 2007.

[13] Paul R. Cohen and Rick Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Inf. Process. Manage.*, 23(4):255–268, 1987.

[14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20:273–297, September 1995.

[15] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11:453–482, 1997.

[16] Klaas Dellschaft and Steffen Staab. An epistemic dynamic model for tagging systems. In *HT '08: Proc. the 19th ACM conference on Hypertext and Hypermedia*, pages 71–80. ACM, 2008.

[17] Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR-WS.org*, 2009.

[18] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[19] J. Gemmell, Th. Schimoler, M. Ramezani, L. Christiansen, and B. Mobasher. Improving folkrank with item-based collaborative filtering. In *ACM RecSys'09 Workshop on Recommender Systems and the Social Web*, pages 17–24, 2009.

[20] Jonathan Gemmell, Maryam Ramezani, Thomas Schimoler, Laura Christiansen, and Bamshad Mobasher. The impact of ambiguity and redundancy on tag recommendation in folksonomies. In *RecSys '09: Proc. the Third ACM Conference on Recommender Systems*, pages 45–52. ACM, 2009.

[21] Jonathan Gemmell, Thomas Schimoler, Bamshad Mobasher, and Robin Burke. Hybrid tag recommendation for social annotation systems. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 829–838, New York, NY, USA, 2010. ACM.

[22] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32(2):198–208, 2006.

[23] Robert Graham and James Caverlee. Exploring feedback models in interactive tagging. In *WI-IAT '08: Proc. the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 141–147, Washington, DC, USA, 2008. IEEE Computer Society.

[24] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[25] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proc. the 16th International Conference on World Wide Web*, pages 211–220. ACM, 2007.

[26] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), April 2005.

[27] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. In *International Conference on Artificial Neural Networks*, pages 97–102, 1999.

[28] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[29] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *WSDM '08: Proc. the International Conference on Web Search and Web Data Mining*, pages 195–206. ACM, 2008.

[30] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proc. the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[31] Graeme Hirst. Resolving lexical ambiguity computationally with spreading activation and polaroid words. In Steven L. Small, Garrison W. Cottrell, and Michael K. Tanenhaus, editors, *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*, pages 73–108. Morgan Kaufmann, San Mateo, CA, 1988.

[32] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. *Semantic Multimedia*, pages 56–70, 2006.

[33] Andreas Hotho, Beate Krause, Dominik Benz, , and Robert Jäschke. ECML PKDD Discovery Challenge 2008 (RSDC'08). http://ceur-ws.org/Vol-497, 2008.

[34] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, February 1912.

[35] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. *Knowledge Discovery in Databases: PKDD 2007*, pages 506–514, 2007.

[36] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.

[37] Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, New York, NY, USA, 2006. ACM.

[38] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural SVMs. *Mach. Learn.*, 77:27–59, October 2009.

[39] Sanghun Ju and Kyu-Baek Hwang. A weighting scheme for tag recommendation in social bookmarking systems. In *Proc. the ECML/PKDD 2009 Discovery Challenge Workshop*, pages 109–118, 2009.

[40] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Multilabel text classification for automated tag suggestion. In *Proc. the ECML/PKDD 2008 Discovery Challenge Workshop*, pages 75–83, 2008.

[41] Christian Körner, Roman Kern, Hans-Peter Grahsl, and Markus Strohmaier. Of categorizers and describers: an evaluation of quantitative measures for tagging motivation. In *HT '10: Proc. 21st ACM Conference on Hypertext and Hypermedia*, pages 157–166. ACM, 2010.

[42] Beate Krause, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Logsonomy - social information retrieval with logdata. In *HT '08: Proc. the 19th ACM Conference on Hypertext and Hypermedia*, pages 157–166. ACM, 2008.

[43] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent Dirichlet Allocation for tag recommendation. In *RecSys '09: Proc. the Third ACM Conference on Recommender Systems*, pages 61–68. ACM, 2009.

[44] Sigma On Kee Lee and Andy Hon Wai Chun. Automatic tag recommendation for the Web 2.0 blogosphere using collaborative tagging and hybrid ANN semantic structures. In *ACOS'07: Proc. the 6th Conf. on WSEAS Int. Conf. on Applied Computer Science*, pages 88–93, Stevens Point, Wisconsin, USA, 2007. WSEAS.

[45] Marek Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proc. the ECML/PKDD 2008 Discovery Challenge Workshop*, pages 84–95, 2008.

[46] Marek Lipczak, Yeming Hu, Yael Kollet, and Evangelos Milios. Tag sources for recommendation in collaborative tagging systems. In *Proc. the ECML/PKDD 2009 Discovery Challenge Workshop*, pages 157–172, 2009.

[47] Marek Lipczak and Evangelos Milios. The impact of resource title on tags in collaborative tagging systems. In *HT'10: Proc. the 21th ACM Conference on Hypertext and Hypermedia*, pages 179–188. ACM, 2010.

[48] Marek Lipczak and Evangelos Milios. Learning in efficient tag recommendation. In *RecSys '10: Proc. the 4th ACM Conference on Recommender Systems*, pages 167–174. ACM, 2010.

[49] Marek Lipczak and Evangelos Milios. Efficient tag recommendation for real-life data. *ACM Trans. Intell. Syst. Technol.*, 3(1):2:1–2:21, October 2011.

[50] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999.

[51] O. Medelyan, E. Frank, and I. H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Internat. Conference of Empirical Methods in Natural Language Processing, EMNLP-2009,*, 2009.

[52] Olena Medelyan, Ian H. Witten, and David Milne. Topic indexing with Wikipedia. In *Proc. the first AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.

[53] Guilherme Vale Menezes, Jussara M. Almeida, Fabiano Belém, Marcos André Gonçalves, Anísio Lacerda, Edleno Silva De Moura, Gisele L. Pappa, Adriano Veloso, and Nivio Ziviani. Demand-driven tag recommendation. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part II*, ECML PKDD'10, pages 402–417, Berlin, Heidelberg, 2010. Springer-Verlag.

[54] Cataldo Musto, Fedelucio Narducci, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. STaR: a social tag recommender system. In *Proc. the ECML/PKDD 2009 Discovery Challenge Workshop*, pages 215–227, 2009.

[55] Emilee Rader and Rick Wash. Influences on tag choices in del.icio.us. In *CSCW '08: Proc. the ACM 2008 Conference on Computer Supported Cooperative Work*, pages 239–248. ACM, 2008.

[56] Steffen Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society.

[57] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD '09: Proc. the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 727–736. ACM, 2009.

[58] Steffen Rendle and Lars Schmidt-Thieme. Factor models for tag recommendation in bibsonomy. In *Proc. the ECML/PKDD 2009 Discovery Challenge Workshop*, pages 235–242, 2009.

[59] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM*

international conference on Web search and data mining, WSDM '10, pages 81–90. ACM, 2010.

[60] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[61] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.

[62] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proc. the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, pages 181–190. ACM, 2006.

[63] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 327–336, New York, NY, USA, 2008. ACM.

[64] Yang Song, Lu Zhang, and C. Lee Giles. A sparse Gaussian processes classification framework for fast tag suggestions. In *CIKM '08: Proc. the 17th ACM Conference on Information and Knowledge Management*, pages 93–102, New York, NY, USA, 2008. ACM.

[65] S.C. Sood, K.J. Hammond, S.H. Owsley, and L. Birnbaum. TagAssist: Automatic tag suggestion for blog posts. In *Proc. the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[66] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys '08: Proc. the 2008 ACM Conference on Recommender Systems*, pages 43–50. ACM, 2008.

[67] M. Tatu, M. Srikanth, and T. D'Silva. RSDC'08: Tag recommendations using bookmark content. In *Proc. the ECML/PKDD 2008 Discovery Challenge Workshop*, pages 96–107, 2008.

[68] Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. Blog post: http://www.personalinfocloud.com/2005/02/, 2005.

[69] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine Learning*, 81:21–35, 2010.

[70] Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Mining Social Data (MSoDa) Workshop Proceedings*, pages 26–30, 2008.

[71] Robert Wetzker, Carsten Zimmermann, Christian Bauckhage, and Sahin Albayrak. I tag, you tag: Translating tags for advanced user models. In *WSDM '10: Proc. the Third ACM International Conference on Web Search and Data Mining*, pages 71–80, New York, NY, USA, 2010. ACM.

[72] Dawei Yin, Liangjie Hong, and Brian D. Davison. Exploiting session-like behaviors in tag prediction. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 167–168, New York, NY, USA, 2011. ACM.

[73] Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. Narrative text classification for automatic key phrase extraction in web document corpora. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, WIDM '05, pages 51–58, New York, NY, USA, 2005. ACM.

Appendix A

Copyright Permission

The following is an excerpt from ACM Copyright Policy, version 7 [1]:

> (...)
>
> 2.5 Rights Retained by Authors and Original Copyright Holders
>
> Under the ACM copyright transfer agreement, the original copyright holder retains:
>
> - all other proprietary rights to the work such as patent
> - the right to reuse any portion of the work, without fee, in future works of the author's own, including books, lectures and presentations in all media, provided that the ACM citation and notice of the Copyright are included
>
> (...)

The policy applies to three articles written by the author of the thesis, which were used in the preparation of the thesis (see Section 1.4 for details).

[1] Full policy available at: http://www.acm.org/publications/policies/copyright_policy