Università degli Studi di Udine
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Informatica

Ph.D. Thesis

A Social Semantic Recommender System

Candidate:
Felice Ferrara

Supervisors:
Carlo Tasso
Antonina Dattolo

Author's e-mail: felice.ferrara@uniud.it

Author's address:

Dipartimento di Matematica e Informatica
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine
Italia

Abstract

The enormous mass of the resources published on the Web contains potentially the responses to the questions of people, companies and even governments: people can better perform their daily activities by enjoying information which can save their time and moneys, the activities of companies and governments can be improved by identifying information which are fundamental to develop business plans and services. The explosive quantity of resources on the Web is the aspect which makes the Web as a container able to satisfy the information needs of a such variegated set of users. However, larger is the set of available resources, harder is the task of finding the information which can satisfy a specific interest. This problem, usually referred as information overload, is tackled by recommender systems since they can filter the resources according to a specific information need. For reaching this aim, recommender systems have to model the interests of each user by monitoring her interactions with an information system: the feedback of the user is utilized to identify her interests and, consequently, to filter the resources.

Web 2.0 tools, on the other hand, innovate the way the users interact with the Web: people are today used to produce and share information which is in turn visited by other peers. This phenomenon exacerbates the information overload problem due to the increasing production of information, but this also opens new opportunities to realize more adaptive and personalized tools. In fact, by publishing new information, people leaves traces about their interests, knowledge and habits. In this work we focus specifically on a meaningful example of Web 2.0 tools named social tagging systems. By using social tagging systems people can share and classify resources by assigning terms (named tags) to resources. By labeling resources with tags, the users can define in a social way semantic relations involving tags and resources which we refer as social semantic relations.

The main aim of this thesis is to provide mechanisms which can use and empower the socially generated classifications in order to model the user interests and to adaptively filter the available resources. For this reason in this work we follow two main research directions. The first one is focused on the proposal of new algorithms aimed at using the social semantic relations defined in social tagging systems in order to model the interests of users and to filter the resources according to the identified interests. The second one is, on the other hand, aimed at empowering the semantic value of the socially defined classifications by suggesting meaningful

labels to the users of social tagging systems. In fact, by supporting the users during the manual classification of the resources it is possible increase the accuracy of the resulting classification.

Acknowledgments

Three years ago I left Naples for starting my PhD at the University of Udine. In order to follow my research interests I had to stay far from the most important people of my life, my family and my girlfriend Angela. We were far but they never stopped to follow my steps. Mom, dad, Angela, Rosalba, thank for your unconditional love!

But these three years were also filled by new interesting professional and educational experiences. I had the opportunity to interact and to work with people who enriched my bag of knowledge. My deepest thanks go to my supervisors, Prof. Carlo Tasso and Dr. Antonina Dattolo, who supported in different ways my activities. I am also very grateful to Andrea, Nirmala, Paolo and Gerhard who worked with me during these years. I would like also to make thanks to the other PhD students who shared with me both the hard and the happy times of this experience in Udine.

My special thanks go to the few good friends who spent with me pleasant free times during these three years. In the future I will remember with joy: the rests breaks at work with Paolo, Marco and Andrea; the dinners and the pizzas with Alexandru; the other pizzas I had with the little Teresa, Marialuisa and Giangiacomo; the coffees with Pasquale; the laughs with Davide and the other right/almost normal guys in the Convitto.

Finally, many thanks to Prof. Francesco Ricci and Prof. Eelco Herder who gave me many suggestions for improving the quality of this work.

Contents

List of Figures

List of Tables

1 Introduction

'*We don't have a choice on whether we do social media, the question is how well we do it*'. This is the main message given by Erik Qualman in [129] where the author emphasizes the growing impact of Web 2.0 applications on people, companies and even governments. The term Web 2.0 is used to classify the applications, such as social networks or social tagging systems, which innovate the paradigm adopted by the first generation of Web applications. The first generation of Web applications allowed people to access the information published by a restricted set of authors. On the other hand, the Web 2.0 introduces tools for supporting the participation of the users: these applications allow people to create, classify and share information. For this reason, the Web 2.0 is also referred as participatory Web (or social Web) and Web 2.0 tools are defined as social media (or social Web applications).

According to Qualman, the revolution introduced by the social media can be read in the numbers which describe the success of tools, such as Facebook [9], which are used by a growing population of users. In fact, it is estimated that the number of users of Facebook is today larger than the US population and, if Facebook were a country, it would have the 3-th largest population in the world (behind only India and China) [12]. The popularity and the success of social applications is also influencing the behavior of Web users. For example, the interactions with Facebook produce a Web traffic larger than the traffic generated by the requests submitted to the Google search engine [8]. This means that social applications are also changing the habits of Web users in accessing information. In fact, the active participation of users of social Web applications transforms these tools in pools where people can find the information they need. For instance, YouTube [25] receives more than 3 billion visit per day and 48 hours of videos are uploaded on the same YouTube each minute [15], the users of Flickr [19] uploaded more than 6 billions pictures [17] and the users of Wikipedia [22] created a multilingual encyclopedia containing more than 20 million articles written in 282 different languages [21]. These data are also more interesting if we consider that less than the 20% of Web users trust Web advertisements while more than the 80% the people consider as usefull the recommendations generated by other people [18]. Companies and governments are also interested in Web 2.0 tools since social applications are currently used for finding a job, for posting opinions about brands, for providing services and even identifying terroristic plans.

The enormous mass of linked data stored and uploaded on social systems makes these tools very attractive for people and companies. In fact, Web 2.0 applications do not introduce new technologies, but they only increase the participation of users who can create connections among documents, people and concepts. These connections are the real power of the social Web applications: people create relations which have a semantic value and these relations can be browsed in order to find new knowledge. Since these semantic connections are generated by the social collaboration (or collaborative intelligence) we refer them as social semantic relations. Using and empowering the social semantic relations for adaptively filtering the information in social systems is the main goal of this work and, more specifically, we will focus on a specific category of social media: the social tagging systems.

Social tagging systems allow people to upload, share and, above all, classify resources. The user generated classifications are the main characteristic of social tagging systems: each user can associate terms (named *tags*) to resources producing, in this way, her own classification. The union of all the classifications provided by the users produces a massive amount of relations involving users, resources and tags.

Social tagging is one of the main features of many social tools, such as Wikipedia and Flickr, where the growing production of information (which makes these systems attractive) generates also an information overload problem: the enormous mass of available information prevents an effective access to the knowledge. In this thesis we follow the idea that the social semantic relations are an expression of the human intelligence which can be used (but also enriched) in order to face the information overload problem. In order to reach this goal we work on two main directions. The first one is to define recommender systems which integrate the social semantic relations in order to both represent the interests of users and to filter the resources. The second direction, on the other hand, is focused on increasing the semantic value of the user generated classifications by suggesting to the users terms and multi-terms able to classify the resources.

1.1 Motivation and Goals

Social Web applications provide users with a set of tools for creating, sharing, and promoting new contents: users can easily leave the role of passive consumers of resources and become active producers (prosumers) of knowledge. This approach increases both the information on the Web and the number of available resources. Consequently, the growing number of resources prevents an effective access to them: a user needs to read the content of each resource for evaluating whether it is interesting for her.

An effective classification of the resources could greatly improve the access to knowledge. Although the manual process usually reaches high quality levels of classification for traditional document collections, it does not scale up to the enormous size of the Web in terms of cost, time, and expertise of the human personnel required.

In order to overcome this limitation, researchers proposed automatic classification tools based on ontologies, which add a semantic layer to the classification process. But, these tools are usually domain dependent due to the obvious difficulties to build and maintain universal ontologies covering all possible information needs [140].

A possible, cheap, and domain independent solution is provided by social tagging applications [108], which are not constrained to a specific informative domain and distribute the task of classifying document over the set of Web 2.0 users. While approaches based on ontologies use semantic information defined by knowledge engineers, in social tagging systems semantic relations emerge from the classification process exploited by Web 2.0 users, that generate folksonomies by tagging resources. This means that on one hand people can freely choose tags in order to classify resources, on the other hand meaningful relations (socially defined) between pairs of tags can be extracted by analyzing the aggregated mass of tagged content.

The tagging activity does not require significant efforts since users can associate tags to resources without following specific rules: each user applies her personal classification which then can be used by others to find resources of interest. For this reason, social tagging applications have both private and public aspects [71]: users may apply tags for personal aims (typically they associate labels to resources in order to find them again), or they can enjoy/exploit the classification applied by other users in order to browse related available documents.

However, due to the freedom of social tagging systems the classification process is not rigorous. This means that the classification proposed by a user may not be useful to other users and, for this reason, tools able to adapt and personalize the access to knowledge embedded in social tagging systems are fundamental to allow users to access information in a highly effective way.

In particular, in [52] the authors argue that, in order to simplify the access to information in folksonomies, the following set of recommendation tasks should be addressed:

- Finding similar people. Given a user, find a community of people with similar interests.

- Finding similar resources. Given a resource, find items in the same topic.

- Finding experts. Given a resource or a set of tags, find people who classify and share relevant information in a specific topic. These people can help a user to find relevant resources.

- Supporting browsing. Suggest tags for refining the search of contents for a given information need.

- Tag recommendation. Given a resource, find a set of tags which classifies the resource in a personalized or not personalized way.

- Content recommendation. Given a user, filter resources according to her interests.

In this thesis we focus specifically on two of these tasks: the content recommendation tasks and the tag recommendation task.

In fact, by following the vision that shortcomings of Semantic Web technologies can be faced by using socially defined classifications, and vice versa, user generated classifications can be supported by Semantic Web tools in order to produce more meaningful classifications we propose to:

- infer social semantic relations from social tagging systems in order to identify the interests of the users and, consequently, to filter meaningful information according to the personal needs of each user.

- enrich and empower the user generated classifications by extracting terms and multi-terms from textual resources in order to have a more semantic description of resources and user interests.

The idea of combining the social and the semantic perspectives is also the main motivation of the larger PIRATES (Personalized Intelligent Recommender and Annotator TEStbed for text-based content retrieval and classification) project [64], an ongoing project developed at the Artificial Intelligence Laboratory of the Department of Mathematics and Computer Science of the University of Udine.

This thesis is aimed at defining some modules of PIRATES which is a general framework aimed at providing support in many different scenarios: in PIM (Personal Information Management), for supporting the identification of relevant Web contents in a personalized way; in E-Learning for supporting the tutor and teacher activities for monitoring (in a personalized fashion) student performance, behavior, and participation; in knowledge management contexts (including for example scholarly publication repositories and, more in general, digital libraries) for supporting document filtering and classification and for alerting users in a personalized way about new posts or document uploads relevant to their individual interests; in online marketing for monitoring and analyzing the blogosphere where word-of-mouth and viral marketing are nowadays more and more expanding and where consumer opinions can be listen.

Figure 1.1 shows the general architecture of the PIRATES framework. PIRATES uses a set of software agents for crawling Web resources as well as other meaningful information provided by Web 2.0 users. Web resources are then classified/labeled by means of a set of tools:

- the IEM (Information Extraction Module), based on the GATE platform, extracts named entities, adjectives, proper names, etc. from input documents;

- the KPEM (Key-Phrases Extraction Module) extracts meaningful keyphrases which summarize each input document;

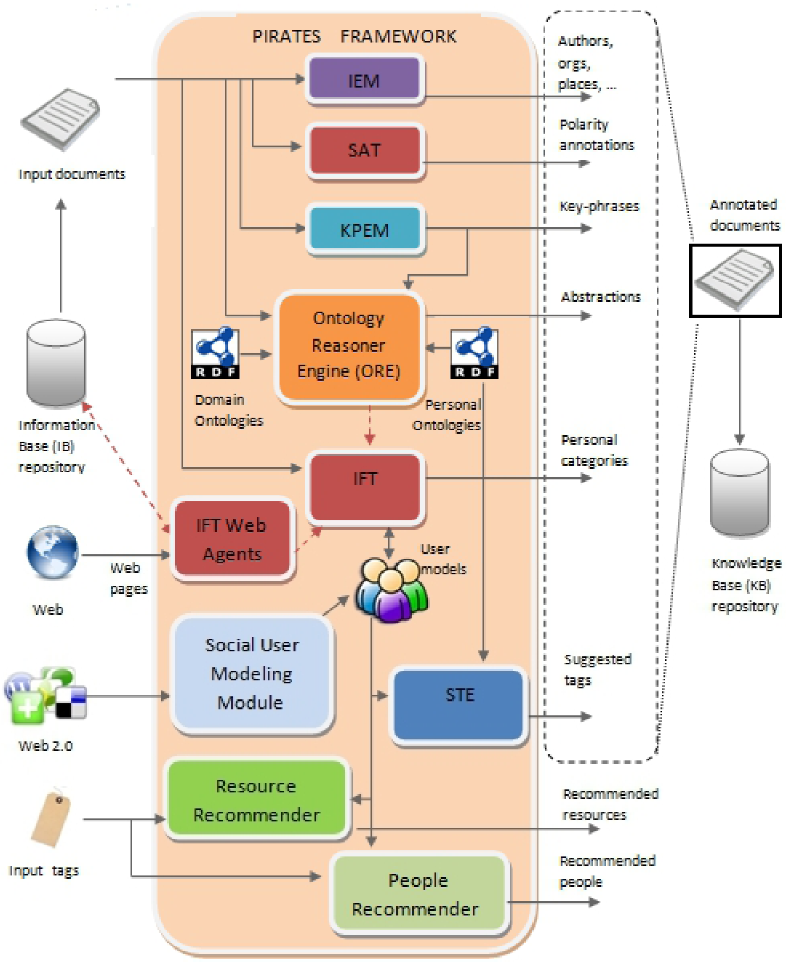


Figure 1.1: The general architecture of PIRATES

- the IFT (Information Filtering Tool) [113] evaluates the relevance (in the sense of topicality) of a document according to a specific personalized model of user interests represented with semantic (co-occurrence) networks;

- the STE (Social Tagger Engine) suggests new annotations for a document relying on the tags generated by Web 2.0 users: social applications (such as delicious, BibSonomy, etc.) are also monitored in order to model the behavior of Web 2.0 users. The personal interests of each user are inferred by taking into account the set of resources that he/she tagged;

- the ORE (Ontology Reasoner Engine) [128] suggests more abstract concepts by browsing through ontologies, classification schemata, thesauri, lexicon (such as WordNet) and by using information extracted by the IEM, KPEM, IFT, and STE modules;

- the SAT (Sentiment Analysis Tool) [50] is a specific plug-in for personalized sentiment analysis that is capable of mining consumer opinions in the blogosphere.

PIRATES is also designed for recommending new potentially relevant contents and for identifying people with interests similar to a given user. For this purpose, PIRATES includes:

- the Resource Recommender module which filters resources according to an analysis of tags and resources interesting to the users;

- the People Recommender module, which identifies experts in a specific domain.

In this thesis we focus specifically on the development of approaches for implementing the Resource Recommender module and the KPEM module.

1.2 Contribution

The contribution of thesis can be summarized in the following points:

- **Survey of the tasks**. A description of the characteristics of social tagging systems and a classification of recommender systems are provided in Chapter 2 by specifying the terminology, the characteristics of most popular systems and the rationale of the main models defined in literature. In Chapter 3 we survey the methodologies for inferring social semantic relations from social tagging systems. The formalization of the techniques used in literature for extracting social semantic relations is then used in both Chapter 3 and Chapter 4 for describing the state of the art of recommending contents and tags in social tagging systems.

- **Collaborative filtering algorithms for recommending resources in social tagging systems**. One of the main goals of this work is to provide new algorithms for modeling the user interests and for filtering resources in social tagging systems. More specifically, we propose two approaches for using the social semantic relations (extracted from social tagging systems) in order to identify the Topic of Interests (ToIs) of each user and to produce personalized recommendations.

  In the proposed approaches, a ToI of a user is defined by means of two sets: a set of tags with a shared semantic meaning, the set of resources that the user labeled with the tags in the topic. Both the approaches define a ToI by taking into account also the resources tagged by a user by following the idea that the labeled documents have information relevant to describe the user interests. In fact tags may be very general and, for this reason, they are sometimes not enough for representing the interests. On the other hand, the resources contain the knowledge that the user is interested to access: the detailed knowledge about the user interests is embedded in the resources.

  The main difference among the proposed approaches is the mechanism used to identify the semantic relations among tags. More specifically, the approaches face the issue of identifying the social semantic relations from two distinct perspectives: a collaborative perspective and a personal perspective. The collaborative perspective is based on the idea that social semantic relations among tags emerge from the collaborative work of the users: the agreement among the users about the meaning of two tags is estimated in order to identify tags with a similar meaning. On the other hand, the personal perspective follows the idea that each user may have personal classification schema which are not respected by a large community of people. For this reason, it makes sense to infer the semantic relations by focusing specifically on the the approach adopted by the user for combining the tags.

- **Approaches for extracting keyphrases from scientific papers and Web pages**. We propose to extract meaningful terms and multi-terms (named *keyphrases*) from the resources in order to enrich the set of tags provided by the users: by extracting keyphrases from resources, we want to more toward a more semantic representation of user interests and resources. More specifically, we focus on the the extraction of keyphrases from scientific papers and Web pages.

  The proposed approaches can be used into two main possible scenarios. The first one is the scenario of the live suggestion of tags: the keyphrases can be suggested to the user when she labels the resources. By supporting the users during the classification task, we can avoid typing errors, prevent poor classifications and reduce the efforts required to the users for classifying the resources. On the other hand, the proposed mechanisms can be used to extend the classification generated by a user by adding a new semantic layer composed by the keyphrases. This semantic layer can better describe the resources tagged by the users and, as a consequence, it can improve the description of the interests of the user.

Our approaches are domain independent and they differ from other state of the art mechanisms since they take into account the way the authors of scientific papers and Web pages organize their content. In fact, the proposed approach is based on the identification of a set of features which take into account the organization of the contents of scientific papers and Web pages. Moreover, we also worked for increasing the coherence of the extracted keyphrases by integrating the information stored in Wikipedia [22] in the computations.

Also if at first sight recommender systems and keyphrase extraction systems may appear weakly correlated, there are some meaningful aspects that link these two research areas. In fact, as described in Chapter 2, recommender systems can use a description of a resource in order to both identify the interests of the users and filter new resources. Keyphrases can be then used to have a short summary of the resources which, in turn, can be utilized in order to classify the interests of the users. In fact, keyphrases can catch the main relevant information in the documents and, consequently, this information can be used to provide a description of the interests of the users. Moreover, in order to information interesting for a user, the description a the specific resource (given by a set of keyphrases) can be analyzed/compared to the representation of the user interests.

In the scenario of social tagging systems, where the tags associated by the users to the resources can be used to have both the description of the user interests and the description of the contents of a resource, the extraction of keyphrases opens also new interesting perspectives. In fact, the social semantic layer introduced by the taggers can be semantically enriched by extracting keyphrases from the resources: keyphrase extraction system can empower the semantic layer generated by the collaborative work of the users. For this reason, we believe that by integrating keyphrase extraction system into social tagging systems we can collect a significant set of information useful to model user interests and, consequently, to improve the accuracy of a recommender system.

1.3 Outline

In this chapter we introduced the main motivations and goals of this thesis. We showed some statistics about social media which evidence:

- the success of Web 2.0 systems and the growing interest of people and companies for these tools;

- the necessity of providing mechanisms able to prevent the information overload problem caused by the increasing amount of uploaded resources.

Moreover, by focusing on the specific scenario of social tagging systems we presented the main goals of the thesis, i.e. to define new methods (to be integrated in the PIRATES project) for recommending resources in social tagging systems and extracting/suggesting keyphrases from scientific paper and Web pages.

Chapter 2 surveys the main characteristics of social tagging systems and provides a classification of recommender systems. The advantages and drawbacks of the socially generated classifications are described by taking into account the characteristics of the tags provided by the users and the features of the most popular social tagging applications. Models, techniques, limitations and benefits of Collaborative Filtering, Content-Based, Knowledge-Based and Hybrid recommender systems are then illustrated.

In Chapter 3 the proposed Collaborative Filtering algorithms are described. The description of the techniques which can be used in order to infer the social semantic relations from socially annotated data are surveyed in order to simplify the description of both the state of the art mechanism and the proposed approaches. In particular the description of the state of the art mechanisms is provided for emphasizing the novelties of the described and evaluated approaches.

The task of extracting keyphrases from scientific papers and Web pages is the subject of the Chapter 4. The approaches for extracting social semantic relations provided in Chapter 3 are still used for describing the works in literature about the recommendations of tags. Since the proposed approach wants to provide more semantic suggestion by extracting keyphrases from the text we also describe the main concept of the Information Extraction (IE) field and current approaches for extracting keyphrases from textual documents. Then, a detailed description of both the proposed mechanisms and the used evaluation methodologies are also given.

Finally, the Conclusions and an appendix including the list of own publications.

2 Related Work

This chapter provides a general introduction to both social tagging systems and recommender systems. More specifically, in Section 2.1 social tagging is discussed by showing the basic concepts, the terminology, the characteristics of the user generated classifications and the features of the most popular social tagging systems.

A survey of recommender systems is then proposed in Section 2.2 by offering a description of the different strategies used in Collaborative Filtering, Content-Based and Knowledge-Based recommender systems in order to model the users interests and, consequently, to adaptively filter the resources. The methodologies for combining different recommender systems are also presented in an overview of the approaches used in literature for building Hybrid recommender systems.

2.1 Social Tagging Systems

Social tagging, referred in literature also as collaborative tagging, folk classification, distributed classification, social classification, open tagging, and free tagging, allows the Web users to classify resources. People do not have to classify a specific set of resources since they classify only the resources they upload. Moreover, each user of a social tagging system does not have to respect specific rules: the user upload a resource and then she only assigns to it one or more terms named *tags*.

A tag is a metadata defined by a user for describing a resource or a property of a resource. There are not vocabularies or schema which restrict the set of the tags which can be applied by a user to a resource. The user is the only one who decides what are the meaningful tags for classifying a document. There are not predefined hierarchies or relations among tags and this simplifies the classification process since the user never wrongs. Tags are also the main mechanism used to browse and to search new resources in social tagging systems. The collection of all the tag assignments performed by a user constitutes her *personomy*. In other words a personomy is defined by the resources uploaded by the specific user and the tags she associated to them. Figure 2.1 shows the personomy of a user (named $User_2$) who uploaded and classified two resources: the resource R_2 labeled with the tags 'Java' and 'Applet'; the resource R_3 tagged with the tags 'Semantic' and 'Web'.

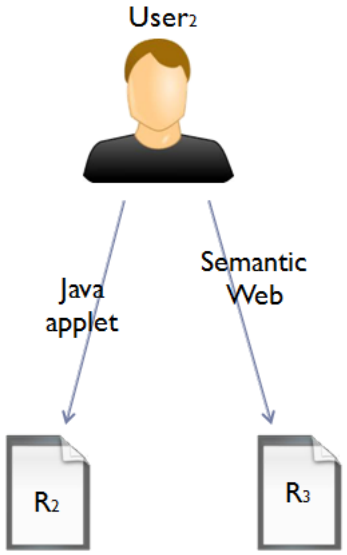


Figure 2.1: An example of a personomy

On the other hand, the union of all the personomies of each user of the specific system is called *folksonomy*. The term folksonomy was coined by Vander Wal [10] by merging the terms folk and taxonomy in order to emphasize that a folksonomy is the result of the classification performed by a community of users. A simple example of a folksonomy is given in Figure 2.2 and, by looking at the Figure, some meaningful aspects can be noticed:

- the classifications provided by each user is merged with the other classifications generated by other people. In the example, the personomy of the $User_2$ represented in Figure 2.1 is merged with the personomies of the users $User_1$ and $User_3$. Each resource is classified at least by one user, but it is possible that the same resource has been labeled by many users.

- There is not agreement or coordination among the users about: the number of tags which must be used for classifying a resource; the choice of the tags which are attached to the resources. In the example, the user $User_2$ assigned two tags to her resources but we cannot assume that she will do the same also in her future assignments. Moreover the users $User_1$ and $User_2$ share the resource R_2 but they used a completely different set of tags for labeling it.

Folksonomies represent an alternative to traditional hierarchical taxonomies which involve selected sets of experts in order to categorize resources according to a strict predefined schema. The manual work of experts produce an index which aims at reducing the potential ambiguities by means of a reference schema such as an ontology. As shown in [98], the indexing process is based on two main pipelined phases,

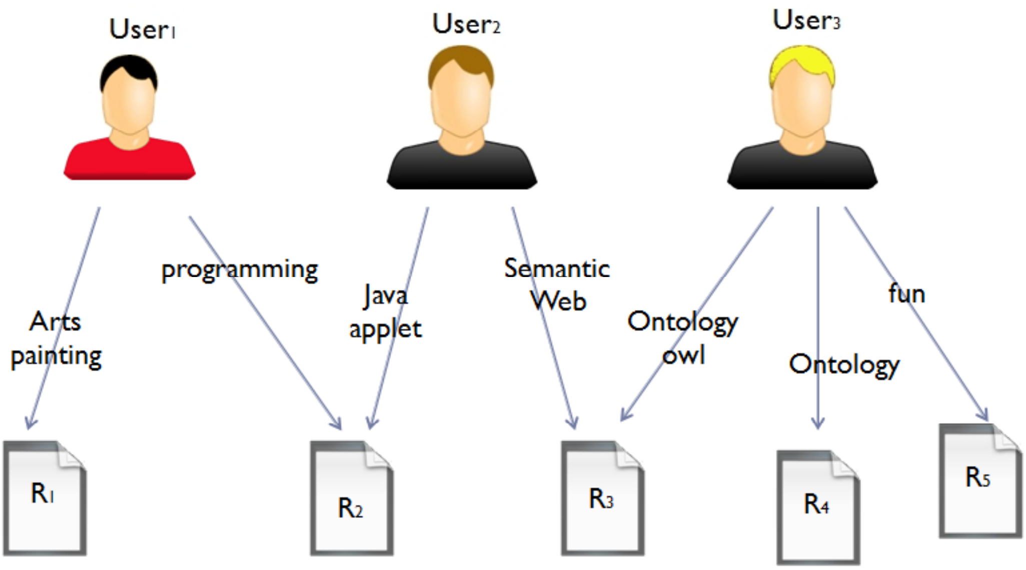


Figure 2.2: An example of a folksonomy

namely, the conceptual analysis and the translation. The conceptual analysis is the process where the human classifier needs to access the knowledge contained in the specific resource in order to detect the corresponding topics or characteristics. Then, the translation is executed in order to map the detected features into the concepts defined in the adopted reference classification schema.

Obviously, such a process is expensive because it requires a systematic work of experts which must be recruited and properly trained. In fact, the experts have the proper knowledge for exploiting the conceptual analysis phase, but they also must be able to follow a well-defined set of procedures and rules required to execute the translation step. The work of experts usually reaches high quality levels of classification for traditional document collections, but it is very expansive and time-consuming. For this reason, it is not a feasible solution for very large collections of resources and, more specifically, it does not scale up to the enormous size of the Web.

On the other hand, users contributing to a folksonomy are free to add tags without using predefined vocabularies or rules. For this reason, two users can define the same concept using two completely different representations, i.e. two different sets of tags. Even a specific user can represent the same concept or property by utilizing distinct set of tags.

Compared to the indexing process executed by a set of experts, the tagging process is cheap since it distributes the work among the Web 2.0 users which spontaneously participate to the classification task. The spontaneous participation of the users erases the costs for recruiting the experts, the lack of fixed rules deletes the need of training the experts, the large number of participant reduces the time requested for classifying the resource.

However, the freedom associated to folksonomies causes some limitations, which may hinder an effective classification of resources:

- Due to the absence of guidelines, constraints, and control, users can exploit the same tag in different ways: for example, acronyms are a potential cause of ambiguity, or the same tag may be written using different lexical forms (e.g. ‘*photo*’, ‘*photos*’, ‘*web20*’, ‘*web2*’, ‘*Web-2.0*’).
- It is frequent to find synonymy, i.e. different words which describe, more or less, the same concept, or polysemy, i.e. single words associated to various different meanings.
- Users classify documents using different levels of expertise and specificity. Since relations among tags are not defined, it is difficult to understand when distinct tags are referring the same concept.

Nevertheless, tags contain rich and potentially very useful social/semantic information, and more specifically two studies ([71] and [151]) focused on this issue by classifying the informative nature of tags. By summarizing the results of these two works we can conclude that tags are commonly used to:

- **describe the content of a resource**. Tags may be used for summarizing the content of a resource.
- **Describe the type of the document**. Some users utilize tags for identifying the kind of document. A document may be classified according to its MIME type (as, for example, ‘pdf’ or ‘doc’) or taking into account the publication form (as, for example, ‘*article*’, ‘*blog*’, ‘*book*’, ‘*journal*’).
- **Describe features and qualities**. Adjectives (such as ‘*interesting*’, ‘*good*’, and so on) may be used for expressing opinions, emotions, or qualitative judges.
- **Associate people to documents**. Tags can report the authors of a document or people involved in a particular task or event. Moreover, tags such as ‘*my*’, ‘*mycomments*’, ‘*mystuff*’, and so on are used to define a relationship between the resources and the user.
- **Associate events to documents**. Locations, dates, conferences acronyms are widely used for associating an event to a document.
- **Associate tasks to documents**. Some tags, such as ‘*mypaper*’, ‘*toread*’, ‘*job-search*’ reveal personal matters or engagements.

These possible motivations should be considered together with the following two further factors:

1. **Heterogeneity of users**. Taggers have different levels of expertise and goals. This has several consequences: classifications exploited by some user may be not understandable (or acceptable) for other users; different users may describe the content of a resource using distinct vocabularies; different users may have different opinions about a topic; users may not have knowledge about people, events, or tasks associated to a resource by other users.

2. **Temporal changes**. Users' knowledge, motivations, and opinions may change over time. A tag used today for describing an item can be useless in the future: emotions and opinions of people may change; reputation of people evolves; a topic may be not any more interesting to the user.

2.1.1 Social Tagging Applications

The galaxy of social tagging applications is very large and some of these applications are going to be ever more popular while some other applications had not success. This galaxy includes applications which have different goals as showed by the following list which reports some of the main popular systems in the field:

- Delicious [6] is a social bookmarking Web application. Founded by Joshua Schachter in 2003, it was acquired by Yahoo! in 2005. It allows the users to share, classify and store URL on a Web platform extending, in this way, the traditional bookmark manager provided Web browsers. In 2008, the service claimed more than 5.3 million users and 180 million unique bookmarked URL [48].

- CiteUlike [3] is aimed at supporting the researchers in sharing scientific references. Developed in its first version by Richard Cameron, it is today sponsored by the Springer Science+Business Media a publishing company which publishes books, e-books, peer-reviewed journals in science, compute science, engineering and medicine.

- BibSonomy [1], developed by a team of students and researchers from the Institute of Knowledge and Data Engineering of the University of Kassel, is a Web application which allows the users to share, store and classify both scientific references (in the bibtex format) and URL.

- Connotea [4], similarly to the CiteUlike system, supports researchers in managing and sharing bibliographic references. In September 2005, Connotea won the Association of Learned and Professional Society Publishers Award for Publishing Innovation, and in November 2005 was shortlisted for the International Information Industry awards in the Best Scientific, Technical and Medical (STM) Product category.

- Wikipedia [22] is a freely available collaborative encyclopedia, supported by the non-profit Wikimedia Foundation. It contains more than 20 million articles (3.79 million of these are in English) written and classified (by means of tags) in a collaborative way. Launched in 2001 by Jimmy Wales and Larry Sanger, Wikipedia is a multilingual encyclopedia: since the July 2011 there are editions of Wikipedia in 282 languages. It is estimated that Wikipedia receives 2.7 billion monthly page views from the United States alone [21].

- Groupme! [20] is a social bookmarking Web site. It was developed by some researches of the L3S Lab in Hannover. Similarly to Delicious it can be used to associate tags to Web resources, but it also allows the users to group resources into sets of objects.

Obviously, the mission of the specific social tagging application has a strong impact on the way the users interact with the application and on the motivation for tagging [160] [107]. These motivations can be divided into two main groups: organizational motivations and social motivations. The first ones refer to the usage of tags as an alternative way of structuring contents and resources: the users are interested in having a repository of personal classified resources which can be further extended by taking into account the classifications provided by other users. The organizational motivations are:

- **Future retrieval**. Tags are commonly used in social tagging applications, such as Delicious, in order to have an index on previously visited resources. Such an index simplifies future retrievals.

- **Empowering the resource sharing**. Social tagging systems can be used as a platform for empowering the resource sharing in a community of people.

On the other hand, the social motivations come from the facilities that social tagging systems provide to the users for communicating. More specifically, social motivations in using social tagging systems are:

- **Expressing opinions**. Opinions, emotions, or qualitative judges are used to usually associated to resources. A social tagging system can be used as a blog where: the contents are imported by uploading an URL; the user polarizes the content by describing her feelings with tags.

- **Attract the attention**. Tags can be used in order to promoting people and/or resources. Tags can be associated to the resources in order to make the resources more visible to the other peers: users can increase the probability that other people will visit a resource by associating popular tags to it. Similarly, tags are also used to put the attention of users on specific people. In fact, tags can be used to associate people to relevant resources with the scope of increasing the visibility of a person or a set of people. Self-presentation is one of the emerging aspects of social tagging applications, such as Flickr, where users are interested to mark resources with a sign aimed at promoting their personal abilities. This strategy can be also used by users interested in proposing themself as an authority for the specific domain.

- **Play and competition**. There are also few applications, such as the ESP game [7], which use the social tagging mechanism for engaging users is competitions. In the case of the ESP game, once a user is logged in, she is automatically matched with a random partner for playing. The two participants cannot communicate and they do not know each other's identity. During the game the same image is shown to the players and their goal is to agree on a tag that would be an appropriate label for the image. They both enter possible tags, and once they enter the same tag (not necessarily at the same time) the system shows them a new picture. The players has two and a half minutes to label a maximum of 15 images. The competition is an expedient to identify meaningful metadata by using the human intelligence expressed by tags. The original version of the ESP game was bought by Google which used it in order to improve the accuracy of Web searches. A new version of the ESP game which is provided in the gwap Web application [11].

A classification of the social tagging systems can be defined by taking into account the way these applications manage and support the interaction with the users. This issue has been analyzed in [107] where the authors propose a taxonomy of social tagging applications by identifying the following seven dimensions:

1. **Tagging Rights**. Social tagging systems define policies about who is allowed to assign tags to resources. Some social applications, such as Youtube, Wikipedia or Technorati, restrict the tagging rights to the resource creator (i.e. the user who uploaded or created the resource). In this case, following the definition provided by Vander Wal, the resulting folksonomy is a *narrow folksonomy*. On the other hand, the social classifications obtained in systems such as Delicious, BibSonomy or CiteULike, where the resources can be tagged by everyone signed in the system, constitute the so-called *broad folksonomies*. Obviously, other possible configurations can be realized. For instance, a social tagging system could allow the users to define personalized policies by granting different permissions for different set of people (friends, relatives, colleagues, and so on).

2. **Tagging Support**. The way the user is supported during the tagging process has a strong impact on the resulting classification. There are basically three possible approaches to support the users during the tagging process, namely, the *blind approach*, the *viewable approach*, and the *suggestive approach*. The

suggestive approach is the one which better supports the user during the tagging process: given a resource, it suggests to the user a set of possible tags for the specific item. Many possible strategies (deeper discussed in Chapter 4) can be used to identify meaningful tags for a resource by taking into account the tags applied by the specific user, the tags applied by the other users in the folksonomy, or the textual content of the resource. A weaker support is provided by the viewable approach because, in this case, the system suggests only the tags applied on the specific resource by the other users. However, this approach cannot provide useful indications if the user is the first to classify the resource. Finally, the weakest support is given by the blind approach where each user cannot view the tags applied by the other users. Usually, the users of systems which adopt the blind approach are supported, at least, by an auto-completing function. This function supports the users when they edit the tags by showing similar tags they used in the past.

3. **Tag Aggregation**. By aggregating the tags assigned to a specific resource we can obtain two possible results: the *set aggregation model* and the *bag aggregation model*. The set aggregation model characterizes narrow folksonomies where by aggregating the tags assigned to a specific resource we obtain a collection of unique tags. The set model prevents duplicate tags for a resource and broad folksonomies can follow this model only if the tagging system requires an agreement among the different users who classify the specific resource. On the other hand, the typical aggregation model which characterizes the broad folksonomies is also referred as bag-model. In the bag aggregation model the same tag can be associated several times to a resource. This is due to the fact that there is not a consensus among the users who classify the resource.

4. **Type of object**. The type of objects which can be tagged in the social application is a relevant aspect which must be considered. Actually, this detail depends on the specific application: photos can be tagged in Flickr, video can be tagged in Youtube, bibtex items are tagged in BibSonomy, Web pages are classified in Delicious, and so on. However, according to the specific class of objects which can be tagged different services and kind of support can be provided to the users. For instance, the suggestion of tags can use the textual content of resources if the object type is a textual object (Web pages or scientific articles), but this cannot be implemented for the photos published on Flickr.

5. **Source of the material**. The resources can be uploaded by the users and/or by the system. The resources uploaded by the users can be resources existing on the Web, such as the URL uploaded by the users of social bookmarking applications or a completely new resource, such as the pictures uploaded in Flickr by photographs.

6. **Resource connectivity**. Resources can be connected each other also independently from the tags applied by the users. Two possible ways of connecting resources can be identified: by linking resources and by grouping resources. Web pages are, for instance, connected by means of links. In the case of Web pages the links among the HTML pages are defined by the author of the hypertextual document. However, it is possible to add links among the resources by using automatic systems. For example, in Delicious the set of tags associated by a user to her bookmarks is automatically extended with other tags which describe the type of the file (such as, filetype:pdf or media:document). Other systems, such as Group.me!, allow the users to group resources. Obviously, also in the case of the grouping strategy, an automatic approach can be used to group resources by using a clustering approach. The connectivity among resources has a relevant impact on several tasks such as the analysis of the relevance of the resources, the tasks of inferring relations among tags, and the task of suggesting tags.

7. **Social connectivity**. As the resources can be connected each other, the users also can be connected by social relations. Several social tagging systems have social network where people explicitly create social links among them. In this scenario, an interesting feature is to allow the user to classify the social relations by labels, such as '*friend*', '*relative*', and so on. This feature is also referred as grouping approach since each user can group her social contacts into sets, in opposition to the unstructured linking approach.

2.2 Recommender Systems

Statistics evidence the explosive growth of the Web during the time: according to a 2001 analysis [106], it was composed by more than 550 billion documents but, in 2008, Google announced that their search engine indexed more that one trillion unique URLs [118]. These numbers are not actually strict bounds on the real dimension of the Web (since they do not take in account some hidden parameters, such as, duplicate or near duplicate Web pages [106]), but they show that the WWW is the largest and fast growing information repository.

Since the 1990s, this scenario introduced new appealing challenges to Information Retrieval (IR), a research field born around 1950 for supporting users to find documents according to their informative needs. The evolution of Web search engines simplified the access to information available on the Web changing the way people accessed the information. This aspect is proved by some statistics which show us that people in 1990s preferred to get information from other people, but in 2004 around 92% of Internet users considered the Web as a good place to find everyday information [61].

The effectiveness of Web search engines depends basically on the function used to state the relevance of resources. This function calculates the relevance of the available resources by using as input the query submitted by the user. However, this schema has some limitations:

- A query is the expression of the perception of the user and, typically, the user does not take in account the methodology used by the search engine to rank resources. Users typically submit queries composed by few keywords without taking in account the ambiguities of the natural language or technical aspects such as the existence of a stop word list.

- Distinct people can have distinct relevance criteria [115] which are traditionally not recognized by the search engine.

These limitations are strongly dependent on the fact that Web search engines are designed for fulfilling a one-time information need without dealing with long term information needs or modeling user interests, which are instead issues addressed by Information Filtering (IF) tools.

Belkin and Croft defined IF as a branch of IR due to the common aim of selecting relevant information [37]. Hanani [76], instead, evidenced some specific aspects which distinguish IF from IR. In fact, IF systems, differently from IR systems, have modules for building and updating a user profile (a machine readable representation of the user interests) which is used to detect information needs and filter resources according to these needs.

Recommender systems are IF tools aimed to support users while they interact with large information spaces, directing them toward the information they need [26]. In order to reach this aim, recommender systems model interests, goals, knowledge and preferences of each user. Then, according to the identified information need the recommender system filters only the set of resources which can more effectively satisfy the user needs.

In the following subsections a general architecture of recommender systems and a classifications of recommender systems are provided.

2.2.1 General Architecture of a Recommender System

Several technologies and algorithms have been proposed to monitor the user behavior,to model user interests and to filter relevant resources, but the underlying architecture of recommender systems never changes. In fact, a recommender system is defined on three main modules: the Profiler, the Matcher and the Advisor.

In order to support the active user (the target user, i.e the user which is going to receive the recommendations) a model of her interests must be built: this task is performed by the Profiler module. More specifically, the Profiler generates the profiles of users and resources by gathering information about the users and the resources which is translated into a proper machine readable representation [111]. More specifically, the machine readable representation of the characteristics of a

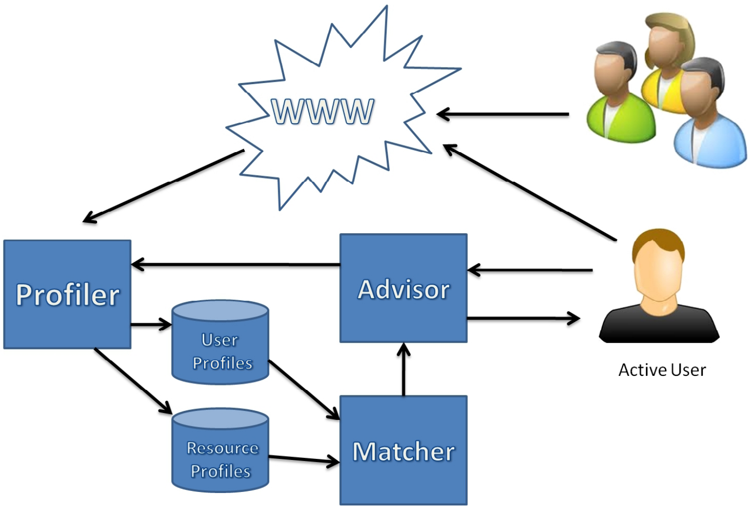


Figure 2.3: The general architecture of recommender systems

user is named user profile: it stores information relevant to personalize the recommendation process by modeling features such as, demographic data (for instance age, sex, nationality), habits, preferences, goals, knowledge on specific domains, and interests.

Several technologies have been used to gather information about the user which can provide both explicit and implicit feedback. When the user returns explicit feedback she explicitly provide her preferences: for instance, the MovieLens system allows users to explicitly provide feedback by voting/rating an item. On the other hand, if when a system collects implicit feedback, then, it infers the user preferences in an unobtrusive way: the system observes the user behavior and then it infers the user preferences according to some criteria. For example, a system can estimate in an unobtrusive way the interest of the user for an item according to the number of times she visits it.

The simplest approach to gather information about the user is to allow the user to actively describe her interests and needs. This approach is the best way to collect stable characteristics, such as demographical data. However, the approach of collecting explicit feedback is not the best choice when users are required to provide a rich and formal description of their interests. In fact, the choice of gathering information about users explicitly or implicitly is linked to a tradeoff between accuracy and costs. By asking people to express judgments on items, the system can obtain more precise details about the user preferences, but this process strictly depends on how much the user works. On the other hand, a system which uses implicit feedback does not add work to the user but it cannot be very precise because the calculation of ratings is based on assumptions which might be false. In order to clarify this concept, the reader can imagine a system which collects implicitly user preferences by observing how much time the user spends in reading a Web page. In this case, it is reasonable to assume that if the user stays on the Web page for a long time then she is very interested in the contents of that specific page. However, the system cannot control if the user is reading the Web page or if she is far from the monitor since she is having a coffee.

Whatever the system collects information about the user, it needs to translate this information into a machine readable format. Ratings are the main objects to reach this aim. In particular, a rating describes/quantifies a relation between a user and a resource (often referred as item) or a feature. Ratings can be unary, binary or scalar values. Unary values are often used when the ratings are collected implicitly. For example unary ratings can be used to indicate that the user visited a Web page or purchased an object: in this case we have information about what the user likes (the visited Web page or the purchased object) but we do not have information about what the user thinks about other items. Unary ratings can be also used when users can explicitly signal only that they like a resource, but they are not allowed to mark what they dislike. When users can signal what they like and dislike, then binary values can be used. Finally, scalar ratings can be used to define a more precise representation of the user interests: the interest for a resource or for a feature can be described by a larger set of values such as a 5-likert scale (the 1-5 stars provided for instance in MovieLens).

The Profiler also generates, implicitly or explicitly, a representation of resources (referred in this paper as *resource profile*). For example, the votes that the users associate to resources by an explicit evaluation describe resources according to the opinions of people. On the other hand, approaches which extract features from resources for identifying the user interests associate some features to the resources too. Profiles of users and resources are analyzed by the Matcher module which is the real core of the recommender system.

Recommender systems have also been described as the play of matching since they evaluate the relevance of a resource for a user by comparing the respective user profile and resource profile. This task is executed by the Matcher module by computing a score for the resource by using of the techniques which will be described in the following sections. The scores assigned to resources are then used by the Matcher module to filter relevant resources which constitute the set of recommendations. Several strategies are used to define the cardinality of the set of recommendations. The simplest approach can order resources according to the score associated to them and then it can filter the top K scored resources. More sophisticated approaches use thresholds to filter only the resources which are really significant to the active user (i.e. the resources with a relevance higher than a certain threshold).

The Advisor module deals with the task of showing to the active user the recommendations computed by the Matcher. The recommendations can be transmitted to the active user in both an pro-active way (for example sending a mail to the active user when the system finds relevant resources) and on-demand way (for example, producing a list of recommendations when the active user submits a query).

When resources are displayed by a Web based application, the system can structure the recommended information in order to maximize the user satisfaction. Researchers involved in Human Computer Interaction highlight that the way the recommended resources are presented strongly influences the perceived accuracy, i.e. the degree to which users feel that recommendations match their interests and preferences [147]. This, in turn, can increase the user confidence in the recommender system creating a virtuous cycle where users are spontaneously interested in providing their preferences in order to have more accurate recommendations [126]. For example, by showing explanations about the criteria which led to recommend a specific resource the user can better understand what are the meaningful elements considered by the system to generate recommendations and, eventually, she can provide more details about her preferences.

The Advisor module may also allow the users to evaluate the received recommendations. This feedback is forwarded to the Profiler module which can use this feedback for refining the profile of the active user in order to produce more accurate future recommendations.

This general architecture is shared by many different implementations of recommender systems. Several classifications of recommender systems have been proposed in literature according to the features used to model users (e.g. demographic recommender systems), the data structure used to represent the user profile (e.g. graph-based recommender systems), the approach used to gather information about the users (e.g. behavioral-based, preference-based) or the recommended objects (e.g. news recommender systems). However, the most accepted classification focuses on two aspects:

1. the representation built by the Profiler and in particular the set of information used to model users and resources
2. the filtering techniques applied by the Matcher module.

According to this classification, recommender systems can be organized into three classes: collaborative filtering, content-based and hybrid recommender systems [104]. These classes will be discussed in the following sections by focusing specifically on: the process of modeling users and resources; the techniques used to filter resources.

2.2.2 Collaborative Filtering Recommender Systems

Collaborative Filtering (CF) is the process of filtering resources according to opinions of people. In this way, CF recommender systems simulate the behavior of humans since people are used to share opinions with their friends or ask suggestions to experts when they need to take a decision [135].

The term 'Collaborative Filtering' was coined by Goldberg et alt. [70] to describe Tapestry, the first attempt to handle an incoming stream of documents by

	Item1	Item2	Item3	Item4	Item5
User1	1	3	2	4	-
User2	4	1	5	1	5
User3	1	4	2	4	1
User4	3	2	3	-	5
User5	1	3	4	5	1

Table 2.1: The User-Item matrix of a Collaborative Filtering recommender system

means of user actions. In fact, the Tapestry system stored not only metadata associated to documents (such as date or author), but it also allowed users to annotate resources. The Tapestry approach was also the first example of *pull model* recommender system, i.e the first system where users had to explicitly manifest their interest in receiving recommendations by submitting a query. Tapestry showed the great potential of approaches based on the analysis of feedback of users and opened a long series of experimentations which also produced new paradigms.

Maltz and Ehrlich [105], for example, proposed the *push active* model where each user can forward an item to people which may be interested in it. However, this model has a drawback: the system needs a massive participation of users which should be also able to recognize the interests of other peers.

This limitation is faced by *Automated Collaborative Filtering* (ACF) the latest and, arguably, the most popular paradigm for collaborative recommender systems since they free users from trusting people, from evaluating opinions of other users and from forwarding documents to their peers. In fact, these systems build a profile for each user in order to take in account only opinions relevant for the specific user. This approach was quickly adopted by several applications in distinct domains (GroupLens and Bellcore Video recommender for recommending movies, Usenet for news article and Ringo to suggest music or artists) due to the fully automation of the word-of-mouth process: users do not have to find people with similar interests because the system will address this task.

Modeling users and resources in CF systems

A CF system can model opinions of people by a *User-Item* (UI) matrix where:

- there is a row for each user
- there is a column for each item
- the cell $UI(i, j)$ has the rating that quantifies the interest of the user i for the item j

The Table 2.1 shows a simple example of the matrix UI built by a CF system by using scalar ratings for modeling users and resources. The reader can imagine that the ratings represent the votes that users assigned to resources (in the range $[0, 5]$ where the user assigned 0 to signal that the resource was completely not interesting while 5 was used to signal that the resource was very interesting).

The matrix describes users and items and more specifically:

- the i-th row $[UI(i,1),\ldots,UI(i,n)]$ of the matrix UI (where n is the number of the items in the system) is the vector which models the user $User_i$ according to her opinions about the resources $\{r_1,\ldots,r_n\}$;
- the j-th column $[UI(1,j),\ldots,UI(m,j)]$ of the matrix UI (where m is the number of users in the system) is the vector which describes the item $Item_j$ according to the feedback of users.

This example also shows that the profiles of users and resources can are merged in one representation (the matrix UI).

The reader can notice that some values of the matrix are not defined (in the example the cells $UI(1,5)$ and $UI(4,4)$) since users usually do not rate each resource in the system. On the contrary, in real life applications, each user provides feedback for a small part of the available set of resources and, for this reason, the matrix UI is usually sparse.

Matrix factorization mechanisms have been used to model the interactions among users and items according to a set of latent factors [94]: both items and users can be modeled by means of vectors over the space of a set of latent features which can both quantify the utility of a latent feature for a user and measure the extent to which the item has those factors. For example, in movie recommender system where the items are the movies, the latent features might be the genre, the duration as well as uninterpretable characteristics. The identification of the features can be implemented by adapting some well-known techniques used in the Information Retrieval field such as the Singular Vector Decomposition (SVD) technique [72]. Given the UI matrix, the SVD method identifies three matrices U, Σ and I such that:

- U is the unitary matrix which describes each users according to the latent factors. The matrix has a row/vector for each user and each component of a vector describes the utility of a specific factor for the considered user.
- Σ is a diagonal matrix which has a row for each identified latent factor.
- I is the unitary matrix which has a row for each item. Each row of this matrix describes an item according to the extent to which each factor characterizes the specific item.
- the matrix UI is given by the product $U\Sigma I^T$

By using this decomposition, it is possible to estimate the utility of a latent features for each user and the relevance of a latent feature for describing an item.

Filtering techniques for CF systems

CF algorithms are commonly divided in two classes: *memory-based* and *model-based* approaches.

Memory-based approaches consider the complete history of ratings to generate recommendations by deferring all computational efforts to the request time and, for this reason, they are also referred as lazy recommendation algorithms.

Two variants of the memory-based approach have been proposed in literature: user-based approaches and item-based approaches.

User-based approaches produce recommendations by considering opinions of a subset of people (technically known as neighborhood) 'similar' to the active user, i.e. people who share meaningful characteristics, knowledge and goals or users who provided similar feedback. The key idea is that users with similar interests, knowledge, tastes and goals are interested in accessing the same information and resources. On the other hand, item-based approaches look for resources similar to that the user liked in the past: given the set of resources the active user liked, these approaches search new items that the community rated similarly.

Both user-based and item-based approaches use algorithms to identify similar users or items by taking into account respectively rows and columns of the UI matrix. Given two rows (in user-based approaches) or two columns (in item-based approaches) of the matrix UI, several metrics can be used to compute the similarity among these two vectors. One of the most popular metric for comparing two users or items is the cosine similarity

$$cosine(a, b) = \frac{a \cdot b}{|a| * |b|}$$

where a and b are the two compared rating vectors (two rows or columns of the matrix UI) and missing values in the matrix are set to zero.

In user-based approaches the clustering algorithm filters a set of users similar to the active one. This set of users constitutes the neighborhood for the active user: the resources relevant to the neighborhood are filtered and suggested. In order to recommend only the most relevant items, the preferences of the active user are used in order to predict the ratings for the available ratings: the items can be ordered according to the predicted ratings for suggesting to the active the items which appear most useful to her.

Several strategies can be implemented in order to predict the ratings. For example, a CF system can assign higher ratings to the resources that neighbors more frequently associated to the highest ratings, alternatively, it can use a weighted schema where ratings expressed by the most similar users are more relevant than others.

In item-based filtering approaches a rating for a resource is computed by taking into account the similarity of the specific resource with the set of resources the user liked in the past.

Both user-based and item-based approaches based on matrix decomposition techniques can compute similarities among items and users by taking into account the latent feature which describe users and items [57].

Differently from memory-based approaches, model-based strategies generate a model aimed at predicting future ratings of users. In these approaches, the UI matrix is processed in order to learn a model of the preferences of users which defines a mapping between ratings and resources. For instance, a probabilistic approach can estimate the probability that a user will assign a specific rating to a specific resource. Bayesian classifiers have been used to address such prediction task. Following the example in Table 2.1, the Bayes theorem can be used to assess the probability $P(Y|X)$ where Y is the event '$User_1$ assign a rating 1 to the resource 5' and it is conditioned by the set of ratings that $User_1$ assigned in the past, i.e $X = (Item_1 = 1, Item_2 = 3, Item_3 = 2, Item_4 = 4)$.

By applying the Bayes theorem we have that

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

By assuming that the ratings of users are conditionally independent we can compute the value of $P(Y|X)$ as follows

$$P(Y|X) = \frac{\prod_{i=1}^{d} P(X_i|Y) \times P(Y)}{P(X)}$$

where d is equal to number of items the active user rated (in our example d is equal to 4). To compute this value we can:

- omit $P(X)$ since it is a constant value;
- estimate the value of $P(Y)$ (i.e. the probability that $User_1$ assign a rating 1 to the resource 5) as the ratio between the number of times that the users assigned 1 to $Item_5$ and the number of all users who rated $Item_5$;
- compute $P(X_i|Y)$ as the ratio between: (a) the number of users who both rated the $i - th$ item as the active user and the number of users who rated $Item_5 = 1$; (b) the number of users who rated $Item_5 = 1$.

Some Observations on CF Approaches

The effectiveness of CF recommender systems depends on certain properties of the application domain. In particular, these approaches produce better results when there are many users, many items and many ratings for items. If this does not happen, it is more difficult to assess similarities among users or items. Four open issues come from this:

- **The cold-start issue**. The system cannot produce recommendations for a new user until she rates a certain number of resources and, moreover, the system cannot recommend a resource until it has been rated by a substantial number of users.

- **The sparsity issue**. Usually, the number of ratings provided by the users is not very large reducing the effectiveness of the system. Two users may share an interest but the system cannot recognize such similarity since they rated different resources.

- **Scalability issue**. Heavy computational efforts are required in order to manage large amount of data.

- **The gray sheep issue**. In small communities there are users which do not agree with the others: these users are also referred as *gray sheep*. They cannot receive accurate recommendations due to the fact that the systems cannot find neighbors for them.

CF systems are more effective when user interests do not change rapidly otherwise the past ratings are no more useful to assess similarities. In fact, due to the fact that the system does not take into account the objective features of the items for producing the recommendations, it cannot immediately recognize that changes in the user interests.

An appealing characteristic of CF systems, from a computational pint of view, is that they do not need to analyze the contents of each resource. This simplifies the implementation of the system which does not need to process the contents of resources. However, this characteristic also makes CF system suitable to information systems where items are homogeneous, i.e. they mainly differ in some subjective criteria: the system does not take into account that the feedback (the ratings) can refer to distinct interests.

The results of the NetFlix Prize [14], an open competition for the best collaborative filtering algorithm able to predict user ratings for movies, showed that matrix factorization mechanisms can be effectively used for generating accurate recommendations. Matrix factorization mechanisms are based on the identification of a set of latent features inferred from the UI matrix which allow to describe both user interests and the items. In such mechanisms, the features extracted for describing items and users do not have a clear semantic value since they are inferred from the ratings stored in the UI matrix. Due to the ability of discovering latent relations among user, items and interests, matrix factorization mechanisms are today the first choice for implementing CF [94], and this is particularly true when users provide explicit ratings. On the other hand, in this work we analyze implicit feedback provided by users which label resources in social tagging systems. In the case of social tagging systems, the UI matrix has only unary values which do not give a detailed description of the user interests. However, by using the social semantic relations, we can have a richer set of relations among users and items. In other words, the mechanisms presented in this work use a set of explicit semantic features (instead of the latent features used in the matrix factorization mechanisms) for identifying the interests of the users and the characteristics of the resources. The major advantage of such approaches is the fact that the users can explicitly request suggestions about one of the interests. In fact, as we will describe in the following chapter each user may have more than one interest and this condition has a strong impact on the accuracy of the generated recommendations. For example, by using explicit semantic features, a CF system can recognize short-term interests by taking into account the labels applied by the user. On the other hand, in this work we use the explicit semantic features for adaptively identifying people who share a specific interest with the user. In this way the proposed approaches can identify specific neighborhood for each interest of the user.

2.2.3 Content-based recommender systems

Differently from CF systems, content-based (CB) recommender systems use only the opinions of the target user. The opinions of the target user are utilized to train a classifier which identifies the relevant resources for the active user: the resources which have features relevant to the active user are suggested to her. For reaching this aim, CB systems model resources according to a certain set of features. This means that each resource is described according a well defined set of features which constitute the *resource profile*. The set of features used to model the resources is also used to build the classifier exploited to identify the relevant resources. In fact, when the active user rates (implicitly or explicitly) a set of resources, then the CB system takes in account the features associated to the rated resources for inferring a set of positive features (the features associated to resources rated positively) and a set of negative features (the features associated to resources rated negatively).

To compute the relevance of a new resource for the active user, the recommender system compares the resource profile to the user profile: the features associated to the specific item are used to evaluate if the active user likes or dislikes the resource.

The idea of comparing resource profiles to user profiles extends classical IR theories which constitute the background of CB recommender systems.

Modeling users and items in CB systems

A central question in CB systems regards the choice of features which must be used to build the profiles of users and items. This choice strictly depends on the characteristics of items and, more specifically, it depends on the fact that the items can be structured, semi-structured or unstructured data.

A catalog is a good example of structured data since it describes items according to a fixed list of features. A movie recommender system, for example, can use a

Title	Genre	Year	Main Actor	Director	Language
Frankenstein	Horror	1910	Phillips	Dawley	Silent
Night of the Living Dead	Horror	1968	Duane Jones	Romero	English
Parto col folle	Comedy	2010	Galifianakis	Phillips	Italian
Il buono, il brutto, il cattivo	Western	1966	Eastwood	Leone	Italian
Inglorious Bastards	War	2010	Pitt	Tarantino	English

Table 2.2: A movie catalog for a Content Based recommender system

database where each movie is associated to information about the genre, the year of production, the main actor, the director and the language, as showed in Table 2.2.

When the items can be represented by such structured representation, then the set of features used to represent the items is very useful to characterize also the user interests. When the user rates (explicitly or implicitly) a set of movies, the system can look at the characteristics of the rated movies to generate the user profile. For instance, following the example in Table 2.2, if the active user rates positively the first two movies in the catalog (Frankenstein and Night of the Living Dead), then the system can infer that the active user likes old fashioned horror movies.

In other scenarios, such as Web page recommender systems, the resources cannot be described according to a well structured schema. In fact, the HTML language is a markup language specifically focused on the presentation of contents and it does not classify the information reported in the corpus of the pages. For this reason, the set of features which describes an item must be inferred by analyzing the contents in the page. The text is the main example of unstructured data and both natural language processing and statistical analyses are usually used to extract meaningful features, i.e. the terms which accurately describe resources and interests.

A possible way to represent textual documents is to project the documents in a vectorial space model by using an encoding format such as the TFxIDF (term frequency - inverse document frequency). Given a document collection, the TFxIDF projects each document (in the document collection) in a multidimensional Euclidean space where the dimensions correspond to the terms in the document collection and the coordinates of a document in each dimension (for each term) are computed as the product of term frequency and inverse document frequency. Term Frequency (TF) describes how often a term appears in the document: higher is the TF, higher will be the term relevance. Inverse Document Frequency (IDF) reduces the relevance of terms which are very frequent in the entire document collection since these terms are not useful to discriminate documents. By using such representation of resources, also the user model can be defined by meaningful terms: a set of positive terms/keywords (terms in positively rated documents) and a set of negative terms/keywords (terms in negatively rated documents) can be extracted from the resources to describe what the user likes and dislikes.

However, this approach does not address the complexity of natural language and the context where keywords are used. The reader can consider the short textual description of a movie '*This is not a comic movie for children*' where the presence of the words comic and children would suggest that the movie is indeed good for children.

As shown in the previous example, traditional mechanisms (such as the TFxIDF) based on the usage of keywords are not appropriate for representing the content of documents and consequently the user interest. In order to contextualize words other approaches extract bi-grams and/or tri-grams from documents [63], build semantically richer profiles by extracting semantic networks from resources [149] or integrate Natural Language Processing (NLP) techniques. Following the previous example, we can for instance use more sophisticated NLP techniques to take into account that there is a '*not*' which changes the meaning of the n-grams '*comic movie*' and '*for children*'. In fact, a limited content analysis is the cause of a poor characterization of the user interests, which in turn can reduce the accuracy of a preference elicitation mechanism.

A better strategy should address semantic relations and the concepts reported in a document. However, it is quite obvious that all the possible relations among the concepts reported in a document cannot be extracted by considering only the keywords which appear in the given resource. For this reason, external knowledge sources have been integrated in some mechanisms to have a more accurate representation of items and interests [103]. More specifically, these approaches are based on the idea of extracting senses instead of terms by using the knowledge stored in ontologies built by knowledge engineer (such as Wordnet [23]) or in sources collaboratively populated by the work of the Web users (such as Wikipedia).

Ontologies describe a certain informative domain by reporting the vocabulary and all the essential concepts in the domain, their classification, their relations including all important hierarchies and constraints [128]. These tools simplify the task of tackling problems of polysemy or synonymy. To clarify this concept the reader can think to the fact that keyword-based methods represent the characteristics of the specific resource by a bag of terms without taking into account potential semantic relations among them. For this reason, if the term '*Java*' is one of the keywords used to model the document, the system is not able to understand if the document (the user) is describing a (is interested in) language programming or islands. On the other hand, by using the information stored in an ontology we can:

- discover that the term can be used for referring two different concepts. Following the previous example we can discover that the term '*Java*' has two possible meanings respectively in the area of computer science and geography;

- disambiguate the term by taking into account also the other terms in a given document. If two terms in a document are connected by 'significant' relations in the ontology than we can infer the real meanings of the terms. For instance, if we found in a document the terms '*Java*' and '*C++*' we may discover that both are programming languages and this allows us to discard the hypothesis that the term '*Java*' has been used to refer an island;

- identify other meaningful terms/concepts which are strongly related to the concepts expressed in the document. Also if the textual resource is describing the characteristic of some programming languages, it is not obvious that the bigram '*Programming Language*' is used in the resource. However, by navigating the ontology meaningful terms/concepts which are not necessarily reported in the document but are still very relevant.

In this way, ontologies such as Wordnet can be used in order to move from bag-of-word models, where documents and interests are defined by terms, to bag-of-sense models where meanings (senses) substitute keywords. Wordnet is a lexical ontology where terms are associated to *synsets* which define the possible senses of the terms. Synsets are used in the JIGSAW algorithm [35] for disambiguating the terms which appear in a textual resource. This algorithm uses both the distances among the terms in a document and the distance among the concepts in Wordnet to assign a sense to each term. Other ontologies are created by knowledge engineer in order to model information in a specific domain by defining specific concepts and relations which describe meaningful elements in a certain area. Taxonomies are ontologies where there is only one kind of relation (the '*is-a*' relation) for connecting concepts in a hierarchical way. These hierarchies can be integrated in a recommender system to model interests and items. For example, the ACM's Computing Classification System (CCS) taxonomy has been used for modeling scientific papers and the interests of the users of the Citeseer system [2]. In fact in [51], the authors use the CCS taxonomy to build both the document profiles and the user profiles as trees of concepts of CSS. Ontologies are something more complex than taxonomies since they can use many different kinds of relations and constraints for connecting concepts. In [41], for instance, an ontology written in OWL is used to represent TV programs and people. In this case, the ontology has information about the characteristics of TV programs, but it also stores some relations which connect the characteristics of broadcasts to the possible preferences of the users. By navigating these relations the authors can identify the preferences of the users: a spreading activation algorithm is used to identify the preferences starting from the characteristics of the liked broadcasts. However, ontologies are built to describe specific scenarios according to given abstraction of the informative domain and this means that: an ontology used to recommend TV broadcasts cannot be used into a different informative domain; two ontologies which describe the same informative domain may have different structures. In order to face these issues different ontologies can be used for obtaining a more comprehensive representation. This approach has been used in the news recommender systems described in [49] which uses 17 ontologies containing concepts from multiple domains (such as education, culture, politics, religion, science, technology, business, health, entertainment, sports). In this way items and user profiles are represented as vectors in the space of concepts defined in the different ontologies.

Ontologies are built by knowledge engineers which are experienced with specific domains and tools. On the other hand, Wikipedia is a knowledge source built by the collaborative work of the Web users. People populate this encyclopedia by adding articles which are also classified by tags. As reported by [103], there are meaningful characteristics which makes Wikipedia an attractive knowledge source for modeling users and interests. In fact, this freely accessible encyclopedia is not focused on a specific domain and moreover it is constantly enriched by the users. The concepts described by articles in several languages are daily added to the encyclopedia which, in this way, has an accuracy comparable to the Encyclopaedia Britannica [69]. The articles of the Wikipedia encyclopedia can be used to compute semantic relatedness among concept as proposed in [66] where the authors propose the Explicit Semantic Analysis (ESA) method. The ESA method measure the distance among concepts by:

- projecting the corresponding articles in a vectorial space (by using the TFxIDF metric);
- computing the distance among the concepts/articles by computing the cosine similarity among the articles.

The ESA method can be used to enrich the semantic value of the description of the user interests as well as the representation of the resources.

Filtering techniques for CB systems

When items are structured data, decision tree and rule induction approaches can be used to filter relevant resources. Decision tree learners, such as ID3 [130], build a decision tree by recursively partitioning items into subgroups until those subgroups contain only items associated to features relevant with respect to the user interests.

When there is a very detailed knowledge about the characteristics of the items (such as in the catalog showed in Table 2.2.3) the user can specify a set of requirements which can be used by the system in order to filter a specific solution. These CB systems are also referred as Knowledge Based (KB) systems which are in turn classified into constraint-based and case-based recommender systems.

Constraint-based recommender systems [156] use a set of well-defined recommendation rules to filter relevant items: the user can explicitly describe her preferences by defining a set of constraints which should be satisfied. Following the example showed in Table 2.2.3, the user could specify that she is interested in cameras with a price lower than 250 euros but with an optical zoom higher than 4x. Then, the recommender system can produce a set of recommendations by executing a query on the corresponding database. Usually, in KB systems, the Advisor interacts with the user by asking details about the user preferences in order to refine the initial set of preferences. In this way, the Advisor tries to minimize the user efforts by asking the user details useful to discriminate among the available items.

id	price	opt-zoom	LCD-size	movies	mpix	waterproof	storage
Item1	170	4x	2.5	no	8.0	yes	4G
Item2	300	8x	3.0	yes	8.0	yes	8G
Item3	250	6x	2.0	yes	4.0	no	6G
Item4	150	3x	3.0	no	4.0	no	6G
Item5	270	6x	2.5	yes	8.0	yes	6G
Item6	220	6x	3.0	no	4.0	no	8G

Table 2.3: A camera catalog for a Knowledge Based recommender system

Case-based systems, such as the Entree system [45], use similarity metrics in order to find resources which match the user interest. In this scenario, the user can explicitly indicate the most relevant properties and the less important ones. The recommendation approach assigns a weight to resources which maximize the profit for the active user.

The recommendation of textual resources, on the other hand, is basically a supervised classification task since the system uses the user profile to evaluate if a resource is relevant to the active user. Techniques from IR are traditionally used to assess the relevance of an item according to the information stored in the profile of active user. For example, Pazzani and Billsus [40] proposed a recommender system where profiles of users and resources are described by a vector space representation: the relevance of an item is computed by the cosine similarity between the user profile and the resource profile.

When ontologies are used to model interests and items the generation of the recommendations can exploit other information stored in these knowledge sources. In [51] the user profile and the description of the items are defined as trees by using the information stored in the SCC taxonomy. Then the distance among an item and the interest of a user is computed by using an edit distance: lower is the number of operations to transform the document tree into the profile tree, higher is the relevance of the specific document.

By using Wordnet a user profile as well as resources can be represented as a vector of the set of senses. The authors of [] use this approach for modeling resources and interests and then they adopt a schema similar to the one used by [40] (described above). In fact, they also use the cosine similarity among the vectors which describe the user and the document in order to measure the interest of the user for the resource.

Observations on CB approaches

Since the representation of users and resource in CB systems is created by extracting features from resources, these systems offer two main advantages respect to the CF systems:

1. User Independence. The performance of the system does not depend on the

participation of users since the system does not have to compute similarity among users.

2. Items Representation. The cold start problem is also partially solved due to the fact that the system creates the representation of resources by extracting features from resources. The system does not have to wait for users' ratings.

However, the preprocessing phase, exploited by the Profiler in order to extract features from resources, adds some shortcomings. In particular, these methods can be applied only if the content has enough information to distinguish relevant items from the others. On the other hand, since the Profiler can use a specific set of features it cannot recognize other hidden features. The cold start problem is partially solved since an item can be suggested without waiting for other ratings but a CB recommender system still needs to process a sufficient number of ratings in order to have an accurate model of the user preferences.

The *plasticity problem* is another limitation of CB systems: the user profile may be overspecialized since the Profiler analyzed only resources in a specific domain of interests and, for this reason, the system cannot not satisfy other short-term interests.

The plasticity problem is also related to another problem referred as *homophily problem* [110]: since a CB system is trained by using examples provided by the user, the system suggests only resources very related to the information the user already knows. The homophily can generate, in a such a way, a informative trap since the active user is supported with a set of polarized information. On the other hand, it would be desirable to have a mechanism able to provide a larger information scope because it is not always true that the information the user needs is very similar to the contents the user visited in the past. This problem can be faced by augmenting the serendipity of the produced recommendation which is one of the characteristics of the characteristics of the recommendations generated by a CF recommender system.

2.2.4 Hybrid Recommender Systems

The discussions on CF and CB system highlights that both these classes of systems have advantages and shortcomings.

Both collaborative and content based approaches suffer from the cold-start problem which is the well-known problem of handling new items or new users. In a CF system, for example, the active user has to provide a sufficient set of opinions in order to be comparable to the other peers and each resource must be rated by a significant number of users. Similarly, also a CB recommender system needs to consider several user interactions in order to identify a relevant set of features able to describe the interests of the active user. However, by modeling the interests according to a well-defined set of features the system cannot recognize some latent features (i.e. characteristics not modeled by the system). Moreover, a CB recommender system can build a plastic user profile which cannot recognize changes in user's preferences and needs. In order to face this issue, some recommender systems build two user profiles, a long-term interest profile and short-term interest profile [40].Knowledge Based recommender systems and CF can manage the plasticity issue better than other systems due to the ability of recognizing some stereotyped scenarios. Collaborative recommender systems also can face this issue better than CB recommender system because they do not work on the features of the items.

Observing that different approaches have distinct strengths and weaknesses, researchers tried to combine results provided by different kind of recommender systems to enhance the user satisfaction. According to Burke [46], the results provided by distinct recommender systems can be merged by seven possible strategies (weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level strategies). This seven strategies can be implemented by three possible hybridization design approaches: monolithic, parallelized, and pipelined [87].

In the monolithic hybridization design, the logics of two or more Matcher modules are merged in just one algorithm. Following the Burke taxonomy a monolithic hybridization design can be used to realize a feature combination or a feature hybridization strategy. A feature combination strategy combines features from different recommendation techniques into one recommendation algorithm. For example, CF approach can be combined with a CB approach by computing similarities among users taking in account not only the way users rate resources but also the contents of the items they liked. In a feature augmentation hybrid system a recommending strategy is used to detect a set of features which will be used by another recommending technique. This strategy has been used in [120] to infer a set rules from the collaborative data in order to derive relevant features for a CB approach.

In a parallelized hybridization design there are two or more distinct Matcher modules running in parallel: the results of the Matcher modules are finally combined in a hybridization step. The weighted, mixed and switching strategies implement the parallelized design. In a weighted approaches each Matcher module associates a score to each resource. The final hybridization step computes the score of a resource by combining the results provided by each Matcher, for example, by a linear combination. The mixed approach basically suggests recommendation computed by all the Matcher module. An oracle is used in a switching hybrid system in order to decide what is the Matcher which is producing the best recommendations.

A pipelined hybridization design joins several Matcher modules in a pipelined architecture where each Matcher module refines the recommendations generated by a previous Matcher. A cascade hybrid, for instance, uses a starting Matcher to filter an initial set of recommendations which can be refined or reordered by other techniques. Finally, in meta-level strategy a recommendation technique is applied to produce a model which is the input used by another technique.

3

Collaborative Social Semantic Recommendations

Recommender systems face the information overload problem by using implicit and/or explicit feedback provided by the users. Social media, such as social networks, blogs, and social tagging systems, innovate the way the users interact with the Web and provide their feedback. This is extremely interesting from the perspective of researchers interested in Web personalization since social applications open new ways to model the user behavior, to infer the user interests and to filter information in a personalized way.

The main topics of this chapter are both modeling the user interests and filtering adaptively resources in social systems. More specifically, focusing on social tagging systems, this chapter proposes new collaborative filtering approaches able to take into account the active participation of the Web 2.0 users: the spontaneous participation of the users can be used to identify their interests and to adaptively filter the resources.

In order to reach this goal we follow the idea that tags can be used in order to identify connections among people, resources and interests. By linking people and resources to interests it is possible to empower collaborative. In fact, collaborative filtering recommender systems work on the assumption that users who showed a similar behavior (i.e. provided similar feedback) share the same interests. However, this is not properly true in systems, such as the social ones, where people are used to provide feedback about many different interests. In this scenario, each user may have more than one interest and this aspect can lower the precision of the recommendations since users who share a part of their feedback not necessarily share all their interests.

In order to clarify this concept the reader can think to the collaborative filtering system as a system which automatizes/simulates the word-of-mouth process. A human being who needs to have information about a specific topic can implement the word-of-mouth mechanism by explicitly asking suggestions to other people. However, the human usually does not select the neighbors randomly in the network of her friends, but it is reasonable that she will take in major consideration suggestions from experts in the specific field or at least interested in that specific topic. In fact, humans can recognize people who (with an higher probability) has the knowledge which can satisfy their information need. By using this ability a person can select a set of people (that are the neighbors selected by the collaborative filtering system) and she can ask them the suggestions about the specific information need. When the person explicitly specifies her information need (by using the natural language), the neighbors can properly respond to the specific request: the neighbors can provide their suggestions about the input topic by ignoring not pertinent information. For example, if a person is interested in having information about '*collaborative filtering recommender systems*' she will probably contact researchers and then the researchers will provide information about that specific topic.

By using this simple example we can learn that the human intelligence has a strong impact on the process used by the people for sharing information and, more specifically, we can deduce two main facts. The first one is that people adaptively select the neighbors according to the specific information need. The second one deals with the fact that the neighbors also filter their feedback according to the specific request.

Integrating the human intelligence of the users of social systems in order to better fit the word-of-mouth mechanism is the main aim of the approaches proposed in this chapter. The human intelligence is the main building block of social systems and it is implemented in different ways in different social systems such as blogs, social networks or social tagging systems. In this chapter we focus specifically on social tagging systems where the human intelligence produces the tags which describe, summarize and classify Web resources. Our aim here is to re-use/integrate the human intelligence (the tags) in a collaborative filtering systems in order to produce/simulate the intelligent behavior which characterize the humans who share recommendations. In fact, since tags are used by the users to classify resources, we use these labels for:

- identifying the user interests. By linking tags and resources to interests, we can classify the feedback provided by users into distinct topic of interests.

- Selecting adaptively the neighbors. By using the feedback provided for a topic of interest we identify a specific set of neighbors for the specific topic. In this way the intelligent behavior of humans who looks for people with specific knowledge is more properly simulated.

- Filtering the feedback of neighbors. The classification of the feedback of the neighbors can be exploited for filtering only the resources which concerns a specific information need of the active user. By throwing out the part of the feedback which is not related to the given interest, we better simulate the behavior of the real life neighbors (which produce the suggestions by taking into account the specific request of the active user).

However, the way the humans identify the neighbors, express their needs and generate the responses is something sophisticated whereas social tags do not produce a rigorous classification. For this reason, we need to reduce the gap between the ability of humans in detecting people, information, resources in a topic of interest and the low accuracy of the user generated classifications. In other words, we need to execute a further step aimed at identifying meaningful semantic relations among tags.

The mass of the socially annotated resources can be analyzed for identifying semantic relations which emerge from the work of the Web 2.0 users. Since these relations among tags are inferred from the social work of the users we call these relations '*social semantic relations*' and we discuss the possible approaches to infer such relations in Section 3.1.

The description of the methods for extracting the social semantic relations from social tagging systems will simplify the discussion (proposed in Section 3.2) about the state of the art in the field of recommending resources in folksonomies. The idea of adaptively generating neighborhood in CF systems is introduced in Section 3.3 and then, in Section 3.4 and Section 3.5, we provide a detailed description of the approaches presented in this work for computing adaptive neighborhood by using social semantic relations. The evaluation and the results of the proposed methods are described in Section 3.6 and, finally, future works conclude the Chapter in Section 3.7.

3.1 Mining Social Semantic Relations

Social tagging systems merge personal and social perspectives: the personal perspectives are embedded in personomies and the social ones come from the union of all the personomies. For this reason, personomies and folksonomies offer two distinct levels for mining social semantic relations. In this section we formalize the possible ways to analyze both a folksonomy (Section 3.1.1) and a personomy (Section 3.1.2) for inferring social semantic relations among tags, users and resources.

3.1.1 Data Mining in a Folksonomy

A folksonomy is defined by a ternary relation, which maps the tagging activities of all users: for each user, the ternary relation stores information about which tags have been applied on which resources. The ternary relation, which involves users, tags, and items, is the starting point to model knowledge, relationships and similarities in a folksonomy. However, mining similarities is not trivial because the ternary relation merges relations among objects of the same type as well among objects of different types. Two approaches have been proposed to handle this scenario:

1. projecting the 3-dimensional space into lower dimensional ones;

2. modeling the ternary relation by a 3-order tensor.

The projection of the ternary relation into binary relations (throwing away information about just one dimension) allows the system to extract the following three different matrices:

1. The *User-Resource* (UR) matrix. It describes the two-way relation between users and resources. Each row of this matrix is associated to a user which is described by a binary vector: if the user u tagged the resource r then the cell $UR(u, r)$ is set to 1 (0 otherwise).

2. The *Tag-Resource* (TR) matrix. It describes the two-way relation between tags and resources. Each row of the matrix, associated to a tag, is a vector, which counts how many times a tag has been applied on each resource.

3. The *User-Tag* (UT) matrix. It describes the two-way relation between users and tags. Each row of the matrix, associated to a user, is a vector, which counts how many times a user applied each tag.

These matrices describe relations among set of heterogeneous objects. Several notions of similarity between pairs of objects of the same type can be inferred by comparing two rows or two columns of the UR, TR, and UT matrices. The cosine and the Pearson similarities are commonly used to assess the similarity between two vectors. By means of this approach, given a pair of users, we can compute:

- UR_user_sim. Extracted from the UR matrix, this measure shows how much two users are similar according to the number of shared resources.

- UT_user_sim. Computed from the UT matrix, this measure specifies that two users are similar if they show a similar tagging behavior.

Given a pair of resources we can infer:

- $UR\_resource\_sim$. Computed from the UR matrix, it states that two resources are similar if they have been tagged by the same set of people;

- $TR\_resource\_sim$. Inferred from the TR matrix, it defines two resources as similar if they have been tagged in a similar way.

Finally, given a pair of tags we can infer:

- TR_tag_sim. It is calculated from the TR matrix and states that two tags co-occurring frequently on the same resources share a common meaning;

- UT_tag_sim. We report this similarity just for the sake of completeness, as it is not really significant. It is computed from the UT matrix and states that two tags, used by the same user, share a common meaning. However, users may have several distinct interests and for this reason they may use tags which are not in any relation.

Unfortunately, the UT, UR and TR matrices used to discover similarities are sparse since each user labels only a small subset of all available resources and use only few tags. This sparsity can reduce the effectiveness of the methods developed to find social semantic relations from these matrices: for instance, the $UR\_resource\_sim$ cannot be used to compare users who did not label the same resources.

Similarities inferred from the UT, UR, and TR matrices can be used to produce the *User-User* (UU) matrix, the *Resource-Resource* (RR) matrix and the *Tag-Tag* (TT) matrix in order to store respectively similarities between pairs of users, resources and tags. These matrices can be used to overcome the computational overhead needed to derive similarities in online scenarios and they represent also the starting point to develop graph-based mechanisms for extracting relevant information from a folksonomy. For instance, the TT matrix describes a graph where each node represents a tag and an edge connects two tags only if the similarity between them is greater than a certain threshold. For example, the PageRank algorithm [106] and the HITS algorithm [106] extract authoritative tags (i.e. tags semantically relevant) from this graph for a given set of input tags. Similarly, the RR and the UU graphs can be built (using respectively similarities between resources and users) and then explored to discover new resources and new users for a given seed of resources or users.

The similarities among pairs of objects of the same type can be used to group together tags, users, and resources with similar properties. This task can be exploited, for instance, in order to create clusters of tags with a similar meaning, people with shared interests, or resources related to same topics or contexts.

Obviously, data mining techniques based on the projection of the 3-dimensional space into lower dimensional spaces lose some information. A different approach to model the ternary relation is to model the 3-dimensional space by a 3-order tensor. The HOSVD [100] method, which generalizes the SVD [145] method to high dimensional spaces, has been experimented to discover latent semantic association among users, tags, and resources. However, there are not statistical evidence that this approach can extract a set of more meaningful social semantic relations.

3.1.2 Data Mining in a Personomy

A folksonomy collapses all users activities by combining all personomies which include different personal interests and tagging strategies. On the other hand, a personomy contains information about just one user and can be analyzed to extract knowledge about the semantic relations that the user built during her tagging activities. More specifically, a personomy can be represented by a *Personal-Tag-Resource* (PTR) matrix, which stores information about how the user applied tags on resources. Starting form PTR matrix the *Personal-Tag-Tag* (PTT) matrix can be built and analyzed to find patterns in the user tagging activities. This matrix describes a co-occurrence graph where each node represents a tag and a weighted edge connects two tags only if the user applied these two tags together. The weight associated to each edge is directly proportional to the number of times the two tags have been used together.

Graph clustering algorithms [134] can be used to detect patterns in the user tagging strategy grouping sets of tags usually applied together to describe items: distinct group of tags can therefore reveal that the user is interested in different and disjoined topics.

3.2 Recommending resources in social tagging systems

The resources in a social tagging system can be recommended by following two possible patterns:

1. The pro-active pattern. The recommender system analyzes the tags used by the user in order to recommend resources whenever it discovers some relevant ones without waiting for an explicit request of the user (for example by sending an email to the user).
2. The on-demand pattern. The user explicitly sends a request (by submitting a query composed by a set of tags) and the system adaptively ranks and filters the resources.

In both these two patterns the tags are used as a query and we call these approaches tag-aware recommendation.

Given a query, i.e. a set of tags, the simplest approaches give higher relevance to:

- resources associated to largest subset of tags in the query. This means that if a user (or a set of people) labeled a resource with all the tags in the query then the specific resource obtains the maximum score. In this case it is sufficient that only one user classifies a resource by using all the tags in the input query to give the maximum relevance to the specific resource. For this reason, the popularity of the resource is not the main feature of this mechanism to discriminate the resources. However, observing that people usually do not provide a large set of tags to classify a resource, it is quite obvious that larger is the set of users who label the resource, higher is the probability to have a more accurate set of tags which describe the resource. In this way the popularity of the resource still influences the computed relevance.
- resources more frequently associated to the tags in the query. Given a resource, larger is the set of users who labeled the resource with at least one of the tags in the query, higher is the relevance of the specific resource. In this case a weaker agreement among the active user and the other people is required. In fact

the relevance of a resource grows up according to the number of people which assigned at least one of the tags in the input query to the specific resource. This means that this second approach is also more strongly influenced by the popularity of the resource.

As we said, in both the approaches popular resources become also the most relevant. However, although the popularity is a good mean for assigning confidence to results, other parameters should also be considered, such as, for example, previous activities or habits of the user (for instance, how she usually applies tags or what resources she visited in the past).

In [136] [138], the authors suggest that tags are a useful mean for understanding the relationship between a user and one or more resources. Following this idea, recently, several researchers proposed some attempts for providing personalized recommendations. The following two subsections describe collaborative and content-based strategies to recommend resources using tags.

3.2.1 Tag-aware Collaborative Recommender Systems

Tag-aware collaborative recommender systems extend collaborative filtering techniques using tags to model user interests and to produce personalized recommendations. In this context, tags have been mainly used to fight the sparsity. In fact, tags have been used to extend the classical collaborative filtering approach by using the tagging histories (i.e. the set of tags utilized by the users) for calculating similarities among users. In this case, the recommender system assumes that people with similar interests share meaningful tags. This allows to fight the sparsity problem (described in the previous chapter), since the system increases the possibility of finding neighbors for people who labeled unpopular resources.

Both memory-based and model-based collaborative filtering approaches, aimed at comparing tagging histories to find similarities among users, have been proposed.

For instance, Social Ranking [155] is a memory-based recommender method that, given a user and a set of tags, computes a personalized ranking of resources. More specifically, Social Ranking extends the set of input tags including other similar tags by means of the TR_tag_sim: in this way, it discovers relevant tags for the user. Then, it calculates a score for a resource according both to the relevance of tags associated it and to the UT_user_sim calculated between the active user and the other users who tagged the specific resource.

Alternatively, a model-based approach has been proposed in [159], where the PTT matrix is considered to identify the distribution of user interests by clustering tags. Two distributions are then compared by means of the Kullback-Leibler divergence to assess the similarity among users.

Another model-based recommender system have been proposed in [158], where the authors present TagiCoFi: it uses tags for facing the sparsity problem inferring some relationships among users and resources also if the users did not explicitly tagged the resource.

3.2.2 Tag-aware Content-based Recommender Systems

Generally speaking, tag-aware content-based recommender systems use tags in order to go deeper into a semantics-based approach. More specifically, they exploit tags for modeling interests, classifying documents, and comparing document representation to user profiles.

Meaningful examples of this trend have been described in [139] and [55].

In [139], the authors describe a recommender system representing users are on the rows of the UT matrix and resources on the columns of the TR matrix. Tag clustering is used to group tags with similar meanings. Each cluster of tags can be seen as a bridge between users and resources; in fact, looking at the user profile, it is possible to understand what tag cluster is relevant for the user and, on the other hand, the description of resources is used to detect resources relevant for a specific cluster. The recommendation algorithm uses as input a tag, a user profile and tag clusters, and produces an ordered set of items. In order to generate a personalized order of items, it computes, for each tag cluster, a score that is associated to both the cluster and the resources labeled by tags in the cluster. More specifically, the score assigned to the cluster depends on the number of times the active user applied the tags in the cluster (this step allows to personalize results), while the score of a resource depends on the number of times users associated to it tags which are in the specific cluster. By using this information, the relevance of a resource for a specific cluster is computed as the product of the score assigned to the cluster by the score assigned to the resource. Finally, given a resource, its relevance is computed by summing the relevance of the resource over all tag clusters.

In [55], the authors present a different approach where both the textual description of items and tags are used to build the user profile. This approach uses the synsets of Wordnet, structures defined as sets of words with a similar meaning and used for defining a semantic indexing of documents. A disambiguation strategy associates a synset to each word in the document looking at words that precede and follow it. Similarly, tags are also disambiguated using the textual content of the resource. In this way, a document is defined as a bag-of-synsets in opposition to the classical bag-of-words [106]. Using this descriptive model, a Bayesian classifier considers the resources bookmarked by the user in order to learn about the synsets, which are relevant to her. Matching the synset representation of documents with the synsets in the user profile, the recommender system calculates a relevance value for each resource.

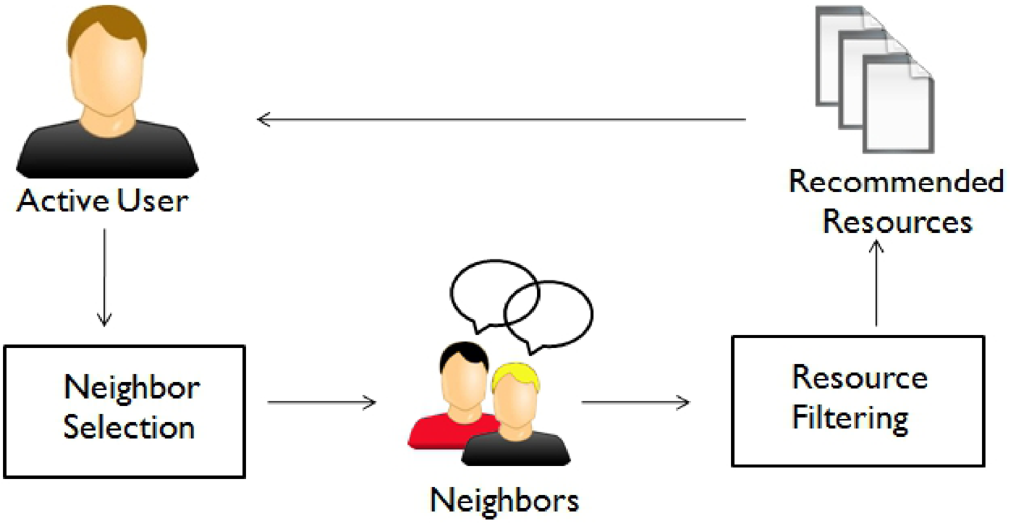


Figure 3.1: The user-based CF workflow

3.3 Adaptive neighbor selection and filtering

As we showed in the previous chapter, CF recommender systems implement the recommendation mechanism by automatizing the word-of-mouth process. The opinions of people are analyzed for identifying useful information which can be forwarded to the target user. This simulates the word-of-mouth process avoiding a direct communication among the people.

In a user-based CF recommender system [135], the simulation of this social process is exploited in two steps as illustrated in Figure 3.1. The first phase is usually referred as '*neighbor selection*' and it is the phase where the system identifies the set of people which share interests, knowledge, goals and tastes with the target user (referred also as the '*active user*'). In the '*resource filtering*' phase the system generates the list of recommendations by combining the feedback (i.e. information about what is relevant or unrelevant for a user) provided by the previously identified neighbors: the resources which are more relevant to the community of the neighbors are suggested to the active user. The rationale of this model is that people with a common information need provided similar feedback in the past and will have similar opinions/behaviors also in the future.

Obviously the precision of the filtering phase strongly depends on the accuracy of the neighbor selection. As reported in [135] the neighbor selection phase can produce better results when the items are homogeneous, i.e. when the items mainly differ in some subjective criteria. In fact, if there are not objective criteria for classifying the resources the recommender system can generate only one neighborhood without taking into account that the feedback of the users may be associated to different topics or contexts. Conversely, if there are significant subjective differences and

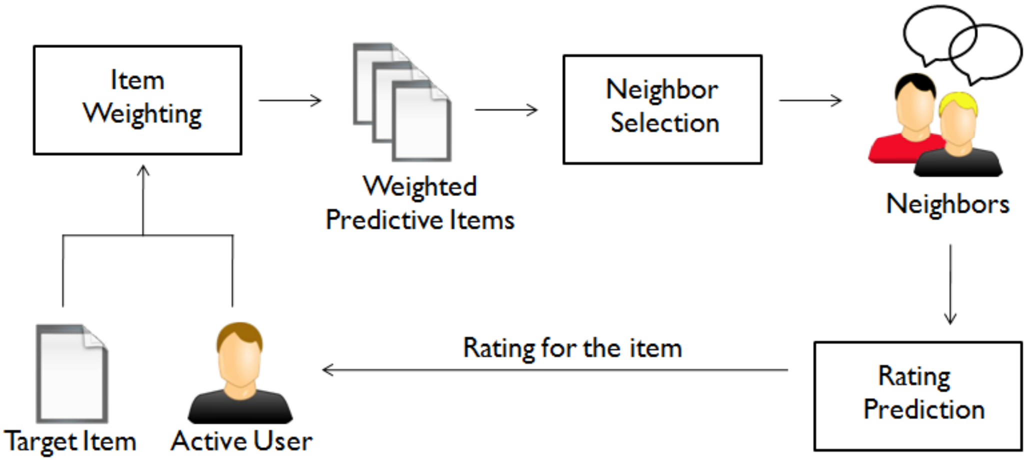


Figure 3.2: The BIPO framework

the entire feedback is used to identify just one neighborhood the accuracy of the neighbor selection is lowered.

In fact, if the user provides, for example, feedback about two distinct classes of objects then one neighborhood is probably not enough: a set of neighbors may provide better suggestions about items which certain characteristics, but a completely different neighborhood can better satisfy the quest for items with other properties. For facing this limitation we can analyze the characteristics of the feedback provided by the users in order to identify distinct neighborhoods.

This issue has been also recognized by Baltrunas and Ricci [33] who faced it within the *BIPO* (*Best Item Per Overlapping*) framework. In fact, the BIPO framework is aimed at predicting the rating of the active user for a target item by computing a locally adaptive neighborhood. As represented in Figure 3.2, given a target item, the BIPO framework takes into account only a part of the feedback provided by the users. This part of the feedback is constituted by the most predictive items, i.e. the items which share meaningful characteristics with the target resource. The rationale of the BIPO framework is that some resources are not relevant to predict the rating for a given target item. The ratings assigned to these noisy resources can even reduce the precision of the prediction. In order to clarify the logic of the framework, the reader can consider the movie domain and, more specifically, the reader can imagine the task of predicting the rating for an action movie. In this scenario, the ratings associated to other action movies or fantasy movies are more relevant than the ratings provided for documentaries.

Obviously, the task of finding the optimal subset of items (for each possible target item) would require a very expansive search in the space of the items. For overcoming this limitation it is possible to use an item weighting approach for finding the most predictive resources for a target item. Given a target item, a weighting approach assigns a score to the other available resources: the resources which are more correlated to the target item are considered as more predictive (they obtain an higher weight).

In this way a weighting schema can be used in order to:

- identify an appropriate set of predictive items for the specific target item. By assigning a weight to the items it is possible to filter the resources which are more strictly related to the target item. This basically means that the weighting approach is a function used to discard noisy items (i.e. the resources not related to the specific target item). In this way the prediction of the rating is not influenced by the ratings assigned to the resources which are not related to the target resource.

- compute the similarity among users by taking into account the characteristics of a specific item. Users who share (with the active user) similar opinions about resources more strongly related to the target item can probably provide more significant feedback for the target item.

The authors of the BIPO framework exploited five different approaches for identifying the most predictive items and they executed an experimentation on the MovieLens dataset [13] in order to evaluate the benefits of each approach. More specifically, the approaches proposed by Baltrunas and Ricci are:

- the Random approach. Just a baseline method used to evaluate the results provided by the other approaches. In this case, the weights for the items are randomly computed.

- the Variance approach. It gives to an item a weight equal to the variance of the ratings provided by all the users to that item. This method does not take into account the target item for assigning the weights.

- the IPCC approach. This approach uses the Pearson Correlation Coefficient to compute the weights. More specifically, in order to compute the weight for an item, it computes the Pearson correlation between the rating vectors which describe the specific item and the target item.

- the Mutual Information approach. It computes the information that an item gives about the target item. The rating vectors associated to the specific item and the target item are threaten as random variables and the the weight is computed as the entropy.

- the Genre weighting approach. This approach uses information about the genre of the target movie in order to assign a weight to the other items. Since the genre of the movie is defined by a set of tags, this approach assigns an higher weight to the resources which share a larger set of tags with the target item.

The experimentation executed on the MovieLens dataset shows that all the approaches outperforms the Random approach as well as traditional CF approach. These results confirm the theory that by adaptively selecting the neighbors there is an improvement of the precision of the CF system.

3.3.1 Adaptive user-based CF in social tagging systems

Users of social tagging systems are allowed to upload and classify resources without providing explicit feedback about what they like or dislike. However, it seems reasonable to assume that a user labels a resource if she is interested in it. For this reason, a user-based recommender system can model the users of social tagging systems by using a unary vector which stores information about the resources labeled by the active user. Given this representation of the user interests, a CF system can implement the recommendation process by using the traditional schema where:

- the neighbor selection is implemented by taking into account the number of resources shared among the users. Larger is the number of shared resources, higher is the probability that the two users share their interests. The Jaccard similarity or the Person coefficient can be used to compare the vectors which describe the users. By using one of these metrics, this phase selects the N users which are more 'similar' to the active one.

- The resource filtering phase produces the final set of suggestions by identifying the resources which are popular among the neighbors. The resources tagged by many neighbors are more relevant than others.

However, in social tagging systems, such as publication sharing systems (like CiteUlike or BibSonomy) or social bookmarking systems, we cannot assume that the items are very homogeneous. In fact, researchers and students label resources which may be in different research areas and, in the case of social bookmarking systems, the items are even more heterogeneous since the users can upload any possible resource on the Web.

For this reason, it makes sense to adopt an adaptive neighbor selection mechanism like the BIPO framework. Unfortunately, the BIPO framework cannot be used in this scenario due to two main aspects:

- Users of social tagging systems do not provide explicit feedback. For this reason, users of social tagging systems can be modeled by using unary ratings since we cannot deduce what the user dislikes. This prevents the usage of the Variance, IPCC and the Mutual Information approaches which use explicit ratings (i.e. Boolean ratings and scalar ratings) for identifying the most predictive items.

- The tags applied by the users do not respect a specific taxonomy. This reduces the accuracy of the results of the Genre weighting approach.

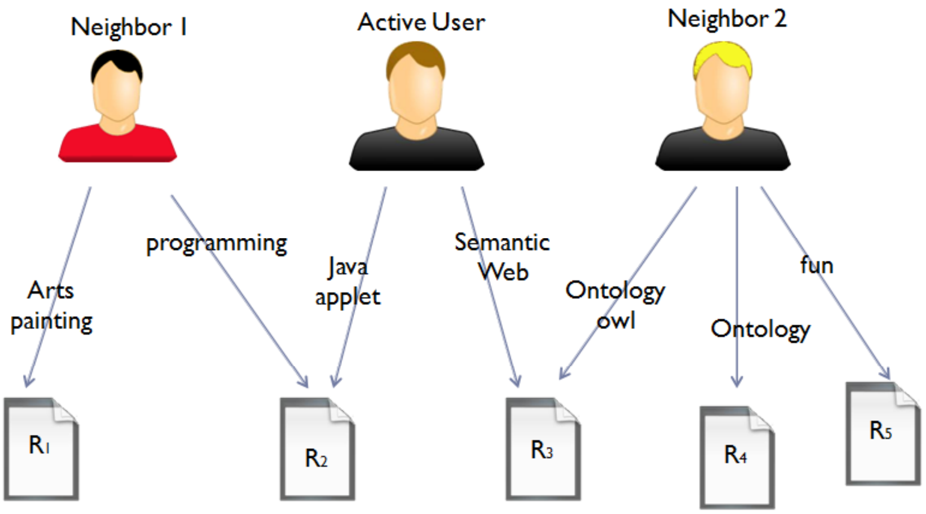


Figure 3.3: A simple example of the recommendation process in social tagging systems

However, an approach able to identify locally adaptive neighborhood can potentially improve the accuracy of the results in this scenario. In fact, in this scenario people provide implicit feedback about many distinct Topic of Interests (ToIs). This specific aspect can reduce the accuracy of the neighbor selection and the resource filtering steps exploited by a user-based CF system and, more specifically, two limitations can be identified.

The first criticality concerns the neighbor selection phase. In fact, by computing just one similarity value for each pair of users, the system ignores that the active user could be similar to a set of users for a certain specific topic, but she could share another ToI with a completely different neighborhood.

Following the 2-step workflow described above it is straightforward to recognize a further criticality in the resource filtering phase, since the system considers all the feedback provided by the neighborhood, without taking into account that such feedback could partly refer to different ToIs. This means that the system could suggest resources which are still interesting to the community of neighbors but which are completely not related to the ToIs of the active user.

In order to clarify these criticalities the reader can consider the simple example shown in Figure 3.3. In order to compute the recommendations for the active user we have to identify the neighbors by taking into account the number of shared resources. Using the previously described neighbor selection approach, the Neighbor 1 appears more similar to the active user than the Neighbor 2: the active user shares half of her bookmarks with the Neighbor 1 whereas the Neighbor 2 shares with the active user only one-third of them. Following the example, the mechanism considers the resource R_1 as the most relevant without taking into account that probably R_1 is not connected to the one of the ToIs of the active user.

In this case the neighbor selection strategy does not take in account that items may belong to different domains and each user may have variegated interests. In other words, the system does not properly simulate the behavior of human beings because the neighbor selection and the resource filtering are implemented without taking into account the interests of the active user.

An alternative approach could focus on the tags the users share in order to identify people with similar interests. However, several limitations lower the accuracy of this method. In fact, two users with the same interests can share the same interests but they could use a completely different sets of tags. This is showed in the example provided in Figure 3.3 where there is a very low agreement among the users also if the users are sharing a common piece of information. The low agreement among the users is caused by the lack of both guidelines able to reduce the number of noisy tags and mechanisms able to suggest tags with a clear semantic meaning. Here we are also interested in evidencing that due to these lacks the agreement on the used tags is a reliable parameter only if the number of shared tags is quite large. In fact, if users share few tags then these tags may be ambiguous or too general for identifying interests and similarities among the users.

These are the motivations of the methods that we are going to describe and where we assume that if two users

In order to face this issue we propose to link the resources to the ToIs of the users by exploiting the social semantic relations inferred from social tagging system. More specifically, by using social semantic relations we propose to group tags with a similar/shared meaning in order to identify the tags applied by the users to describe a specific ToI. By grouping the tags with a similar meaning we can also link the resources to the ToIs: the resources classified with tags in a specific cluster of tags are linked to the ToI described by the specific set of tags.

In this way, similarly to the BIPO framework, the feedback for a ToI can be used to identify a specific neighborhood (for the specific interest) and to discard the remaining feedback (the other resources) which is not linked to the ToI.

Differently from other approaches defined in literature we do not compute the similarities among the users by taking into account the tags they share but we mainly focus on the set of shared resources. There are two main motivations for this choice. The first motivation is due to the gray sheep issue (defined in the previous chapter): if the approach selects the neighbors by taking into account the number of shared tags, then it is more difficult to identify the neighbors for people who adopt unpopular classification schema, i.e. tags which have not been used by a significant set of users. The second motivation is the fact that tags provide a flat description of resources: tags often do not provide specific information about the resource and this means that they have not a strong impact of the classification of the resources. In fact, the reader can consider that two users can use the tag '*web*' for describing a very large set of applications, technologies and researches, which can also be very different among them.

Following these ideas, in this chapter we present two collaborative filtering algorithms aimed at:

1. finding locally adaptive neighborhood in social tagging systems;
2. filtering adaptively resources for a given ToI.

The main difference between the two proposed approaches is the mechanism used for inferring the social semantic relations. More specifically, in Section 3.4 we present an approach based on the idea of mining social semantic relations from the entire folksonomy. Then, in Section 3.5 we propose an alternative approach which does not need to process the entire corpus of annotated data in order to find relations among tags and ToIs. In fact, by mining the personomy of the each user, this second approach takes into account only the semantic relations built by each user.

3.4 Adaptive CF by Mining the Folksonomy: the ACFF approach

This section describes the ACFF (Adaptive CF from Folksonomy) approach which is the first of the two CF approaches presented in this chapter. This approach is based on the idea that semantic relations among tags can be inferred by looking at co-occurrences among tags over the entire folksonomy. A detailed description of the procedure used to identify the ToIs of the active user by using these relations is given in Section 3.4.1. Then, the mechanism used to adaptively identify the neighbors and to produce the recommendations is shown in Section 3.4.2.

3.4.1 ACFF approach: Identifying the ToIs

In order to extract the social semantic relations spontaneously generated by the community of users we process the annotated data by projecting the ternary relation involving users, resources and tags into a simpler binary relation. More specifically, we projected the ternary relation into the bi-dimensional *Tag-Resource* (TR) matrix (as described in Section 3.1.1) and then computed the similarities $sim(t_i, t_j)$ among each pair of tags t_i and t_j as the cosine similarity among the two corresponding rows of the TR matrix.

In order to have a set of meaningful relations we computed the similarities for the tags used by at least 10 users and we also used a similarity threshold in order to discard similarities too low. By computing the distances among the tags, we can filter for each tag a set of similar tags, i.e. the tags which have a shared meaning with an input tag. As an example, we show in Table 3.1 the list of the top 10 similar tags for the tag '*java*'.

By using these similarities we can identify the ToIs $\{ToI^1_{au}, \ldots, ToI^m_{au}\}$ of the active users au where the k-th ToI is defined by the pair

java	1.0
develop	0.358
programming	0.357
eclipse	0.251
development	0.244
framework	0.239
software	0.235
opensource	0.213
computing	0.208
informatic	0.202
jena	0.199

Table 3.1: The top ten weighted tags for the tag java computed by the ACFF approach

$$ToI_{au}^{k} = \left(T^{k}, R_{au}^{k}\right)$$

where

$$T^{k} = \left\{(t_1, w_{t_1}^{k}), \ldots, (t_n, w_{t_n}^{k})\right\}$$

is a set of weighted tags. More specifically, the set of the weighted tag T^k is composed by a collection of pairs $(t_i, w_{t_i}^{k})$ where:

- t_i is the i-th tag in the collection of the similar tags. The tag t_i in the collection T^k is semantically related with the concept expressed by the tags in T^k.

- $w_{t_i}^{k}$ is the weight associated to the tag t_i. This weight quantifies the semantic relatedness with the concept expressed by the set of tags. The weight/similarity w_{t_i} is computed as the distance among the tag t_i and the tag t_1 (which is the tag in T^k more frequently used by the user au), i.e. it is equal to $sim(t_1, t_i)$.

The set T^k is also used to infer R_{au}^{k} and, more specifically, we have that

$$R_{au}^{k} = \{r_1, \ldots, r_m\}$$

is the set of resources tagged by the user au with the tags in T^k.

In order to identify the ToIs of the active user we start from the most used tag for defining the set T^1: it is composed by the tags similar to the most used tag. By using T^1, we can filter the resources in the ToI_{au}^{1} and, more specifically, the set R_{au}^{1} is the set of the resources labeled by the active user with the tags in T^1. The procedure continues by executing the same step for the next most used tag which has not been included in T^1. This routine is repeated until each resource is included in at least one of the ToIs. By adopting this strategy, we assume that the most used tags are more significant to describe what is interesting for the user. Then in order to catch all the possible meanings of the most used tag we extract from the TR matrix the other tags which are more related to it. In this way, we can find also the other tags applied by the active user which have been used to describe the ToI.

The reader must take into account that the set T^k is not limited to the tags applied by the active user but, on the other hand, the set R^k_{au} is composed by the resources labeled by the active user. This strategy has been developed according to the idea that a tag can have an ambiguous meaning and the labeled resources are fundamental to catch the real meaning of the tag. In fact, the possible meanings of a tag (described by the set T^k) can be inferred by taking into account the social semantic relations, but to prune the unrelevant meanings we have to take into account the labeled resources. For this reason, the recommendation mechanism proposed in the following section takes into account the fact that the users share same resources for identifying people who share the same ToI.

3.4.2 ACFF approach: Recommending resources for a ToI

Given the ToI

$$ToI^k_{au} = \left(T^k, R^k_{au}\right)$$

the set of resources R^k_{au} is used to identify the neighbors. The neighbors for the topic ToI^k_{au} are the users in the community who share the largest set of resources in R^k_{au}. In order to filter the top N neighbors we assign a score to each potential neighbor, i.e. to each user who tagged at least one of the resources in R^k_{au}. The set of tags T^k is used to filter also the feedback provided by the neighbors for the specific ToI. More technically, for each potential neighbor j, we identify the set R^k_j which is the set of the resources labeled by the potential neighbor j with the tags in T^k. The Jaccard similarity coefficient is used to compute the similarity between the active user au and the potential neighbor j on the topic ToI^k_{au} as follows

$$user\_sim(au^k, j^k) = \frac{\left|R^k_{au} \cap R^k_j\right|}{\left|R^k_{au} \cup R^k_j\right|}$$

The score assigned to each potential neighbor is used in order to filter the top N neighbors. The resources labeled by these neighbors with the tags in T^k are the resources which can be suggested to the active user for ToI^k_{au}. In order to rank these resources we assign a score to each one of these resources. The score of a resource is computed by taking into account

- the similarity between each neighbor (who labeled the resource) and the active user;
- the weights of the tags associated by each neighbor to the resource.

In particular, given the resource r in R_j^k, the neighbor j contributes to the score of the resource r as follows

$$score(au^k, j^k, r) = user\_sim(au^k, j^k) * w(t_l)$$

where the tag t_l is the most weighed tag in T^k associated by the neighbor j to the resource r.

The final score of the resource r for ToI_{au}^k is then computed as the sum of the contributions of each one of the N neighbors as follows

$$score(r, ToI_{au}^k) = \sum_{i=1}^{N} score(au^k, i^k, r)$$

By assigning a score to each resource we can order the resources and we can finally filter the most relevant ones. Since we are interested in defining a ranking of the resources we do not normalize the resulting score.

3.5 Adaptive CF by Mining the Personomy: the ACFP approach

The ACFF approach is based on the assumption that there is a semantic relation among two tags if they co-occur on the same resources a sufficient number of times. However, by using the social semantic relation extracted from the folksonomy:

- we cannot properly support users who adopt a classification schema which is not shared by a sufficient set of users;

- the way each user combines the tags is not considered in the computation.

In fact, by inferring the semantic relations among the tags from the folksonomy we cannot take into account the personal tagging strategies adopted by each user. On the other hand, a personomy has information more strictly related to the specific user. This means that the analysis of a personomy can reveal relations among tags which disappear when we merge all the personomies.

In order to face this issue we propose a second approach, referred in this thesis as ACFP (Adaptive CF from Personomy) approach, where a community detection algorithm is used in order to extract the social semantic relation defined by the active user as shown in Section 3.5.1. On the other hand, the tags used by the neighbors for describing the specific ToI are inferred by filtering the tags assigned (by the neighbors) to the shared resources as shown in the Section 3.5.2.

3.5.1 ACFP: Identifying the ToIs

The set of ToIs $\{ToI_{au}^1, \ldots, ToI_{au}^t\}$ identified within the personomy of the active user, constitutes her interest profile. The ToI_{au}^k is defined as

$$ToI_{au}^k = \left(T_{au}^k, R_{au}^k\right)$$

where

$$T_{au}^k = \left\{(t_1, w_{t_1}^k), \ldots, (t_n, w_{t_n}^k)\right\}$$

is the set of weighted tags used by the active user au to annotate her resources in the topic k and

$$R_{au}^k = \left\{(r_1, w_{r_1}^k), \ldots, (r_m, w_{r_m}^k)\right\}$$

is the set of resources tagged by the user au with the tags in T_{au}^k.

More specifically, T_{au}^k is defined by a set of semantically related tags

$$tag(T_{au}^k) = \{t_1, \ldots, t_n\}$$

applied by the active user, where two tags are considered to be in a semantic relation if the active user has applied them together to classify one or more resources. The weight associated to each tag represents the relevance of the tag with respect to that ToI and it is used to compute the relevance of each resource $res(R_{au}^k) = \{r_1, \ldots, r_m\}$ tagged by the active user within that ToI.

In order to identify the semantic relations defined by the active user we analyze her personomy throwing out information about the tagged resources. In particular, given the personomy of the active user we build an undirected weighted graph P where: each node represents a tag; an edge connects two tags if they have been used together to label one or more resources; an edge connecting two tags is weighted by the number of times two tags have been used together.

Figure 3.4 shows the graph P for one of the user of the BibSonomy social tagging system [38], where we do not show the weights associated to edges to make the graph readable.

Given the graph representation of a personomy, we apply a community detection technique for grouping tags with a shared semantic. In particular, we follow the idea proposed in [99] where a node (representing a tag in our model) may be in more than one cluster identifying, in this way, overlapping clusters/coverage of tags. This community detection technique identifies groups of tags with a shared meaning by identifying subgraphs from the starting graph P, where each subgraph G maximizes the following fitness property

$$f_G = \frac{K_{in}}{(K_{in} + K_{out})^{\alpha}}$$

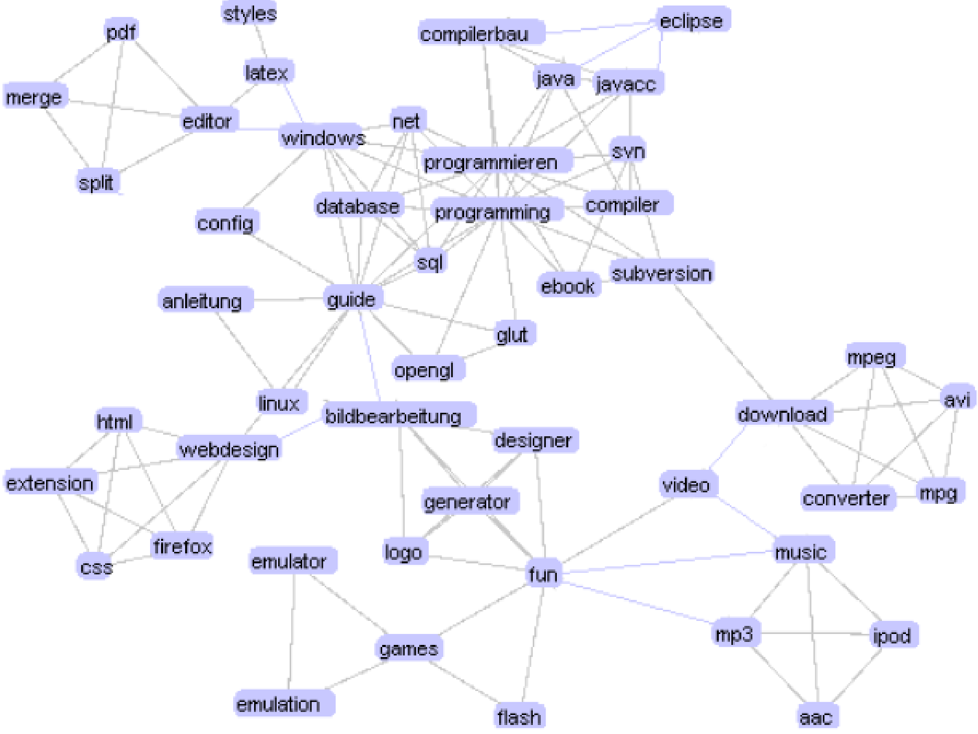


Figure 3.4: An example of the graph P for a user of the BibSonomy system

where K_{in} is the sum of the weights of the edges which connect two tags belonging to G, K_{out} is the sum of the weights which connect tags in G with the rest of the graph, and α is a parameter which controls the size of clusters. In other words, the fitness of a subgraph increases when we add to the subgraph a tag that the user has exploited frequently in co-occurrence with tags in the subgraph and rarely with the other. Given a node, the algorithm builds a coverage for the node in the graph by adding at each step, the node which maximizes the following fitness function

$$f_G^a = f_{G+a} - f_{G-a}$$

where $G + a$ $(G - a)$ is the subgraph obtained by adding (removing) the node a to the subgraph G. The process which adds tags to the coverage stops when there are not tags with a positive fitness value.

More specifically, given a tag/node t, the coverage for the node is initially defined by the subgraph G which contains only the node t. The subgraph G is then extended by a set of iterations where, at each iteration, the following steps are executed:

1. a loop is performed over all neighboring nodes of G not included in G;
2. the neighbor t_i with the largest fitness $f_G^{t_i}$ is identified;
3. if the fitness $f_G^{t_i}$ is greater than 0 the tag t_i is added to G and a new iteration is executed, otherwise the procedure ends.

By using a very similar procedure, the authors of [99] computed the clusters/communities of tags in a given graph by:

1. randomly picking a node in the graph (not yet assigned to one of the identified clusters/communities);
2. generating a coverage for the selected node.

On the other hand, our approach slightly modifies this strategy by taking into account the number of times the user applied each tag. More specifically, at the first iteration our approach defines a coverage for the most used tag. Then, the approach identifies the cluster for the most used tag which has not been yet included in a cluster. The clustering algorithm ends when each tag is at least in one cluster. At the end, each subgraph detected by the clustering phase contains the set of tags associated to a certain ToI for the active user.

However, given a subgraph G^k, some of the tags in G^k are less relevant than others since they possibly are used also for referring to other different ToIs. Therefore, we associate a weight w_t^k to each tag t in the subgraph G^k by computing the centrality of t in the specific subgraph by summing the weights of the edges which connect the tag t to the other tags in the subgraph. These weights are then normalized in $[0, 1]$ by dividing each weight by the maximum weight. In this way

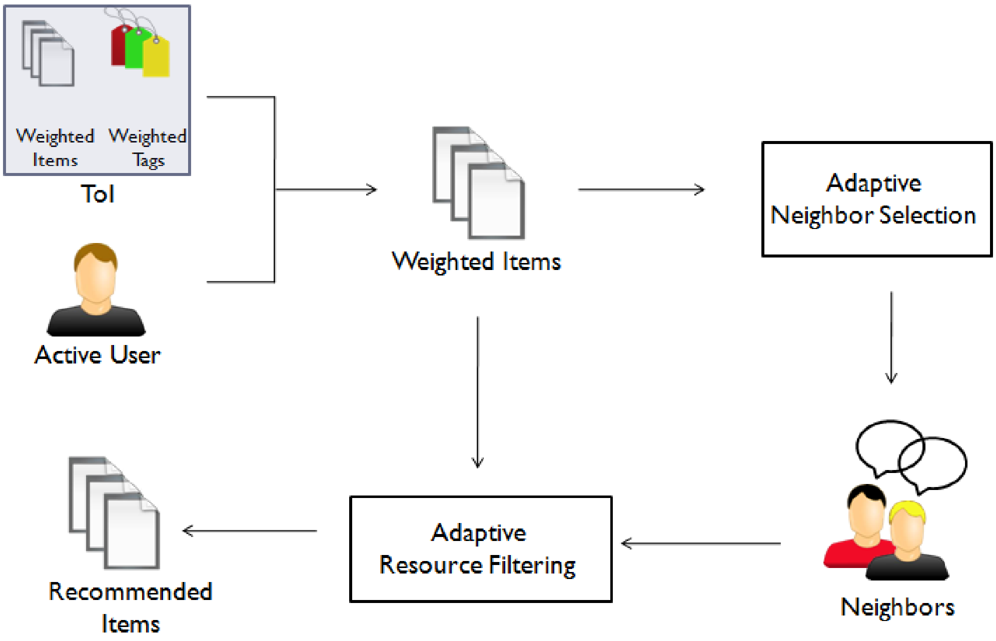


Figure 3.5: The recommendation process executed in the ACFP approach

we determine the set T^k_{au} and applying this strategy on the example showed above for the most used tag 'programming' with $\alpha = 1.0$ we obtain the weighted set of tags: (programming, 1.0), (programmieren, 0.77), (guide 0.54), (java, 0.45), (javacc, 0.40), (compilerbau,0.36), (windows,0.31), (database, 0.27), (net,0.27), (sql, 0.27), (glut,0.22), (opengl,0.22), (eclipse,0.22), (compiler, 0.18), (svn, 0.18), (subversion 0.18), (ebook, 0.18), (anleitung, 0.09), (linux,0.09).

The set T^k_{au} is used to infer the set R^k_{au} such that $res(R^k_{au})$ is composed by the resources that the active user labeled by tags in $tag(T^k_{au})$ and the weight w^k_r for the resource r is equal to the maximum weight of tags in $tag(T^k_{au})$ which the active user associated to r.

3.5.2 ACFP: Recommending resources for a ToI

This section focuses on the recommendation process graphically described in Figure 3.5. More specifically, given the $ToI^k_{au} = (T^k_{au}, R^k_{au})$ the approach uses the weighted set of items R^k_{au} in order to execute the *adaptive neighbor selection* phase (Section 3.5.2) and then, in the *adaptive resource filtering* phase, the personomies of the best neighbors are analyzed in order to filter and combine the feedback for the specific ToI (Section 3.5.2).

ACFP: Adaptive neighbor selection

Given the $ToI^k_{au} = (T^k_{au}, R^k_{au})$ of the active user, the set of weighted resources R^k_{au} is used to filter the set of neighbors for the ToI. In particular, the approach identifies people interested in the specific ToI by taking into account only the users who tagged the resources in $res(R^k_{au})$. We assume that people interested in ToI^k_{au} share with the active user relevant resources within the specific ToI. For this reason, let $R_{shared}(u, R^k_{au})$ be the set of resources that the user u share with the active user in $res(R^k_{au})$, we compute how much the specific interest of the active user is matched by the neighbor u by computing the following

$$InterestMatch(u, ToI^k_{au}) = \frac{\sum_{r_i \in R_{shared}(u, R^k_{au})} w^k_{r_i}}{\sum_{r_i \in res(R^k_{au})} w^k_{r_i}}$$

The logic behind this formula is that higher is the number and the relevance of the resources in R^k_{au} that the neighbor u tagged, higher is the interest of u in the specific ToI. By using the InterestMatch function, we can filter the set N^k_{au} of neighbors interested in ToI^k_{au}.

ACFP: Adaptive resource filtering

In order to produce recommendations we need to identify new resources (labeled by the neighbors in N^k_{au}) which are related to the specific ToI.

We follow the idea that, some tags in the personomy of the neighbor u are more trustworthy than others for finding resources relevant for ToI^k_{au}. In fact the neighbor may also have several ToIs and, for this reason, we are interested in discovering which tags utilized by u better account for the ToI of the active user.

In order to identify the most meaningful tags, we take into account the way each neighbor classified the items in $res(R^k_{au})$. More specifically, due to the fact that each neighbors u labeled only a subset of resources in R^k_{au} we take into account the set $R_{shared}(u, R^k_{au})$.

We consider more trustworthy the tags which have been used by the neighbor to label many relevant resources within ToI^k_{au} and, specifically, we measure the trustworthiness of a tag t_j in the collection of the neighbor u with respect to ToI^k_{au} as follow:

$$trustworthiness_u(t_j, ToI^k_{au}) = \frac{\sum_{r_i \in R_{shared}(u, R^k_{au})} w^k_{r_i} \cdot \phi(u, t_j, r_i)}{\sum_{r_i \in R_{shared}(u, R^k_{au})} w^k_{r_i}}$$

where $\phi(u, t_j, r_i) = 1$ if the user u applied the tag t_j on the resource r_i, 0 otherwise. By using this metric we compute, for each neighbor a new set of weighted tags. Following the example described in Section 3.5.1, in Table 3.5.2 we report the list of the most weighted tags for the three most similar neighbors.

$Neighor_1$		$Neighor_2$		$Neighor_3$	
develop	1.0	java	1.0	java	1.0
java	0.67	opensource	0.51	rdf	0.78
computing	0.57	development	0.30	tool	0.5
informatik	0.44	programming	0.21	toolkit	0.5
tools	0.30	semanticweb	0.20	rdbms	0.5
frameworks	0.21	rdf	0.20	library	0.36
eclipse	0.17	ruby	0.20	ruby	0.28
software	0.17	api	0.19	sysadmin	0.26
xml	0.12	framework	0.19	bash	0.26
ruby	0.09	library	0.15	language	0.26
programming	0.09	graph	0.14	antlr	0.26
plugins	0.09	owl	0.14	config	0.26
agile	0.08	visualization	0.10	schema	0.26
soap	0.08	tools	0.10	shell	0.26
knowledge	0.08	xfire	0.10	dictionary	0.26

Table 3.2: The most trustworthy tags of three neighbors for a given ToI

Following the example of Figure 3.4, in Table 3.5.2 we report the top 15 trustworthy tags for three neighbors. The reader can observe that the neighbors classified the shared resources by using different set of tags.

Following the principle that trustworthy tags are associated to relevant resources of the neighbor u, we assign an higher relevance to resource labeled by more trustworthy tags. Specifically, we compute $rel_u(r_j, ToI^k_{au})$, which is the relevance of the resource r_j in the personomy of the neighbor u with respect to ToI^k_{au}, as the highest trustworthiness associated to tags that the neighbor u assigned to r_j.

Finally, the relevance of a resource r_j for the active user with respect to ToI^k_{au} is computed summing the relevance of r_j over the collections of the neighbors N^k_{au} as follow:

$$rel(r_j, ToI^k_{au}) = \sum_{u \in N^k_{au}} InterestMatch(u, ToI^k_{au}) \cdot rel_u(r_j, ToI^k_{au})$$

This allows to produce the ranked list of resources which are recommended to the active user.

3.6 Evaluation

As reported in [87], recommender systems require that users interact with computer systems as well as with other people. For this reason, some methods, traditionally used in social behavioral research, can be used in order to analyze the way the users interact with a recommender systems and, more specifically, for responding to research questions such as:

- are users satisfied of the quality of the recommendations they receive?
- Why should people be interested in providing feedback such as ratings or tags?
- Is the quality of the recommendations a parameter which can increase the trust in the system?
- Why people are interested in receiving recommendations? Is it depending on the novelty or what else?

These research questions address different goals/dimensions of recommender systems which can be more formally defined. Formal definitions of such goals allow researchers to define metrics able to measure if a given strategy can effectively improve the user satisfaction and/or increase the incomings of e-commerce platforms which adopt such technologies.

One of the most meaningful aspects to take into account is related to the accuracy of the generated results, i.e. to find a response to the question '*is the recommender system returning relevant results?*'

However, as in the information retrieval field, the task of evaluating the precision of the results generated by a recommender system is not simple due to the many different dimensions of relevance [115].

A first step, toward the identification of the best way to measure the accuracy of the results is to define what is the task that the recommender system should support. A comprehensive discussion about the tasks of recommender systems (and the corresponding suitable metrics for evaluating each specific task) is available in [77] where the authors identify the following tasks:

- **Annotation in Context**. Given a certain context, the user is interested in finding information which can better support her current activity.
- **Find Good Items**. Many recommender systems return a ranked list of recommendations without showing the rating prediction or the scores associated to the resources. In this scenario the recommender system aims at finding a short list of good items, i.e., to find just the most interesting items for the user.
- **Find All Good Items**. The previously described task focuses on finding a reduced set of items. This is not surprising since the most general goal of a recommender system is to face the information overload problem: it does not make sense to produce information overload by forwarding too many information to the active user. However, there are some application domains where it is important not to lose any possible relevant piece of information. In such a

scenario, the recommender system is aimed at finding all items related to the specific information need.

- **Recommending Sequences**. The recommender system may be interested in recommending items in an order more suitable to the user preferences and knowledge. The reader can consider the case of a scientific paper recommender system, where it is important to provide information to students and researchers by taking into account their previous activities and knowledge.
- **Just Browsing**. Recommender systems are often considered as tools able to increase the incomings of e-commerce platforms by suggesting the right products to the consumers. However, this assumption is not completely true, since users can browse a Web site without necessarily buying an object. Anyway, the browsing experience can make the system attractive to people.
- **Find Credible Recommendations**. Trust is one of the most important aspects which can make the users willing both to use the system and to provide new feedback. If the system provides credible items then the users will trust the system and the recommendations. One of the approaches used to make the recommendations credible is, for instance, to explain why the system generated that set of recommendations.
- **Improve Profile**. The accuracy of the generated recommendations directly depends on the accuracy of the user profile. The way the system collects information about the user interests is, for this reason, one important task to be addressed.
- **Social Information Spreading**. Some users use the recommender system as a social tool which allows them, for example, to express their opinions. In this case, the recommender system is used as a forum where people can provide ratings in order to share their opinions with the others. By using a recommender system as a social tool the users can have a constructive or disruptive role. In fact, the some user can be motivated to help other people in finding new potentially interesting information, but, on the other hand, other people can introduce ratings in order to influence the opinion of the other users or attract their attention on something irrelevant.

In this work we are interested in supporting the users by suggesting them good items. However, the task of evaluating if a recommender system is returning good items is not simple since, from the user perspective, the accuracy of recommendations may depend on many different factors (such as novelty or diversity) which are hard to be evaluated in an objective way. Recently, some researchers focused on the idea of evaluating the results provided by a recommender system by measuring the perceived accuracy, i.e. to measure the degree to which users feel that recommendations match their interests and preferences [127]. However, the perceived accuracy (measurable by means of questionnaires aimed at collecting the explicit feedback from the users) can be influenced by many parameters such as the emotional situation of the users or the way the system presents the results. Due to these limitations, the evaluation of the perceived accuracy can produce results which are harder to be generalized.

On the other hand, in this work the results of the proposed approaches are evaluated by using a quantitative evaluation. A quantitative evaluation estimates the accuracy of the proposed recommendations by using historical datasets which include information about the ratings (in our scenario the labeled resources) provided by the users of existing systems. A quantitative analysis can be executed by using a part of the ratings of each user (in the historical dataset) to build the her user profile. These profiles can be then used as input of the evaluated recommender systems in order to compute the recommendations and, consequently, to evaluate the accuracy of systems: the similarity among the ratings predicted by the recommendation approach and the ratings provided by the active user (but not included in the user profile) is used to estimate the accuracy of the recommendation approach.

Since the ratings of users of social tagging systems can be modeled by a unary vector we do not have explicit ratings and, for this reason, our evaluation can only check if the proposed recommendations have been visited/tagged by the users. This evaluation works on the assumption that the user labels the resources she likes, but this assumption is not completely true. In fact, the main limitation of a such off-line evaluation depends on the high number of unrated resources: the users in the dataset usually evaluated just a small part of the available resources. This means that the system can produce meaningful recommendations which have not been rated/tagged by the active user: an off-line evaluation considers such recommendations as not relevant for the active user and this lowers the quality of the evaluation.

Anyway, since quantitative approaches are the most used techniques in the literature to discriminate among the accuracy of the recommendations provided by different systems [87] we chose to execute a quantitative analysis in order to compare our approaches to the Social Ranking mechanism. However, the reader must take into account that the Social Ranking algorithm that we used to evaluate the results has been implemented by ourself and it only follows the indications shown in [155]. For this reason our implementation of Social Ranking cannot take into account hidden parameters (such as any possible threshold on the number of tags or similarity among tags).

Given an input tag, Social Ranking computes the relevance of each resource by combining two similarities values. In fact, in Social Ranking, given a user u and a tag t_k, the relevance of a resource p is higher if

- the resource p has been labeled with tags similar to the tag t_k, i.e. with tags which more frequently co-occur with the tag t_k in the folksonomy.
- The resource p has been labeled by people similar to the user u, i.e. by users who share many tags with the user u;

In Social Ranking, the similarity among two tags is computed by following the same strategy we used in the ACFF approach. In fact, by computing the cosine similarity among the rows of TR matrix associated to the tag t_i and t_j, the similarity $sim(t_i, t_j)$ is estimated according to the number of times the two tags co-occur on the same resources. On the other hand, Social Ranking estimates the similarity $sim(u_i, u_j)$ between the users u_i and u_j by computing the cosine similarity between the rows of the UT matrix associated to the two specific users.

Technically, given a tag t_k and a user u, the Social Ranking algorithm assigns a score $R(p)$ as follows

$$R(p, t_k) = \sum_{u_i} \left(\sum_{t_x} sim(t_x, t_k) \right) \cdot (sim(u, u_i) + 1)$$

In the following subsections, we describe the characteristics of the datasets used in our evaluation 3.6.1, the procedure executed to compute the results 3.6.2, and finally, in Section 3.6.3, the results are shown and commented.

3.6.1 The Datasets

In order to evaluate the results produced by our approaches by means of a quantitative analysis several decision need to be taken. A first issue regards the possible application domains of social tagging systems. In fact, a specific strategy could provide more accurate results in a specific domain, but the same approach may not return satisfactory suggestions in another different domain. For instance, the users of publication sharing systems (which are mainly researchers) could have a tagging behavior very different from the users of social bookmaking Web sites. This aspect can influence the accuracy of the predicted recommendations. In order to decide if an approach outperforms another one we cannot use datasets from a specific domain, otherwise we cannot generalize the findings. Such problem is also worsened by a second issue which deals with the difficulties in collecting information from social tagging applications. In fact, due both to privacy issues (since some social tagging applications allow the user to have private posts which cannot be analyzed) and to limitations imposed by the social tagging Web sites (which often do not allow to automatically download the complete list of posts of each user in the system), it is not simple to collect information about the resources tagged by the users of social tagging systems.

In order to face the generalization problem we decided to use two distinct historical datasets (referred in this work as BibSonomy-item and BibSonomy-bookmark respectively) from two different domains: the domain of publication sharing systems and the domain of social bookmarking Web sites. These two dumps of the Bibsonomy system contain respectively the collection of tagged bibtex and the collection of tagged bookmarks of the users of the BibSonomy system from 1995 to 2011.

	BibSonomy-bibtex	BibSonomy-bookmark
Distinct users	4597	4392
Distinct tags	142330	99557
Distinct posts	526665	325627
Distinct items	497330	277140
Avg items per user	114,15	74,14
Avg tags per item	3,7	3,2

Table 3.3: Statistics about the used dataset

Publication sharing systems (represented by the Bibsonomy-bibtex dataset) and the social bookmarking Web sites (represented by the Bibsonomy-bookmark dataset) have two main features which allowed to face the generalization problem. The first feature is constituted by the users who interact with the two systems since: publication sharing systems are mainly used by researchers who usually upload resources about a set of research fields; social bookmarking Web sites are used by the largest set of Web users without no limitations to specific topics. This aspect can also influence the sparsity of the matrices used to infer similarities among tags, users, and items. The second relevant difference is given by the objects tagged by users. This is a meaningful aspect since users of publication sharing systems and social bookmarking Web sites can be differently influenced by some existing classifications. In fact, scientific publications are often classified by the same authors of the papers or by means of some other classification scheme (such as the ACM taxonomy) which can influence the users in generating their classification. On the other hand, in social bookmarking applications people are usually not influenced by other classifications: the author of a Web page can associate some metadata (by properly using the HTML language) to the document but this information is not shown to the users.

These two datasets are publicly available for research goals since the 2008 when they were published in the context of the ECML/PKDD Discovery Challenge. Today these datasets are continuously updated and they are still available to researchers who are interested in performing experiments in the area of social tagging systems [39].

In table 3.6.1 we report some statistics about the two datasets.

We did not preprocess the datasets in order to filter unpopular tags or resources since we were interested in testing the approaches in a realistic scenario. However, we computed the results by taking into account the recommendations generated for the users who tagged at least 40 items. This constraint appears reasonable since collaborative filtering approaches can produce effective recommendations when users rate a significant number of items [135].

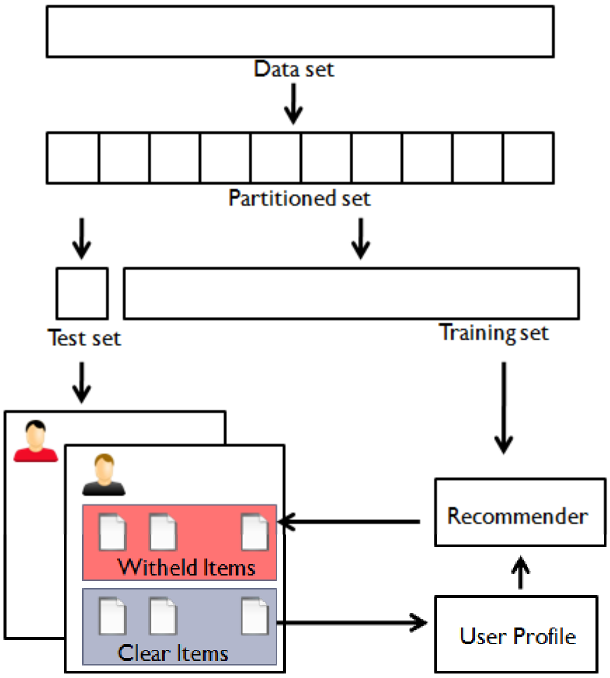


Figure 3.6: The N-cross validation process

3.6.2 Experimental Settings

In order produce reliable results not biased by a set of user profiles, the *N-fold cross-validation* technique was utilized. This technique is based on the idea of repeating an experiment by using as input N disjunct fractions of the available dataset. More specifically, by following the approach used in [122], in our work we used a 10-fold validation technique (graphically described in Figure 3.6) where, given a dataset, each one of the 10 folders is generated by taking into account the posts of 10% of the users in the dataset. In this way, each one of the starting datasets (both the BibSonomy-bookmark and the BibSonomy-bibtex datasets) has been partitionated into 10 folders randomly populated. By using these folders we repeated the experimentation where: one of the folder was used as test set; we merged the remaining folders in order to obtain the training set.

We used the training set to compute the similarities among tags (for the Social Ranking approach and the ACFF approach) and to extract the feedback of the neighbors in order to generate the recommendations for the users in the test set. More specifically, given a user in the test set, we randomly withheld 10 items from her posts. The remaining items (the Clear Items in Figure 3.6) were used to generate the user profile. In particular, the tag most frequently used to classify the Clear Items was used as input of the evaluated approaches. In fact, we had to select a specific tag for generating the recommendations since the Social Ranking approach generates a set of recommendations by using an input tag. For this reason, we also used the tag most frequently used to classify the Clear Items as centroid for generating the ToIs for our CF approaches.

The aim of the evaluation was to compare the ranking generated by the three mechanisms by taking into account the positions assigned to the withheld items. We compared the ranking by taking into account only the first 20 recommended items. A very similar approach was used in [122] to evaluate the results of collaborative filtering algorithms for the CiteUlike social tagging system.

The quality of the computed recommendations produced by the mechanisms was evaluated by adopting two measures, named *hit-rate*(HR) and *average reciprocal hit-rank* $(ARHR)$ respectively, which have been used also in [56] to compare collaborative filtering recommender systems. More specifically, the HR measure is defined as follow

$$\textit{hit-rate} = \frac{\textit{Number of hits}}{m}$$

where m is the total number of users considered in the evaluation and we count a hit when the system produces at least one correct recommendation (i.e. a recommendation for one of the withheld items). Given the lists of recommendations for the m users produced by a recommendation mechanism, the hit-rate is a value in $[0,1]$ which is higher when there is a larger number of users who received at least one recommendation for one of the withheld items.

The main limitation of the HR measure is given by the fact that hits are evaluated regardless of their position, i.e, a hit that occurs in the first position of the list of recommendations is treated equally to a hit that occurs in the last position. In other words, the capability of the recommender system to better rank resources is not recognized. In order to face this limitation we also used the *ARHR* measure which is defined as

$$ARHR = \frac{1}{m}\sum_{i=1}^{h}\frac{1}{p_i}$$

where h is the number of hits and p_i is the position of *i-th* hit. *ARHR* is still a value in $[0,1]$ but it represents a measure of how well the recommender mechanism is capable to rank a hit in high-score positions.

By using the *HR* metric and the *ARHR* metric we evaluated the quality of the results provided by the Social Ranking approach and the approach we presented in this work. More specifically, the reader can notice that the *HR* measure is an upper-bound for the *ARHR* metric and we consider the *ARHR* a more accurate measure. For this reason we consider the results of the *ARHR* for identifying the best results.

	HR				ARHR			
	10 N	20 N	30 N	40 N	10 N	20 N	30 N	40 N
BibSonomy - bookmarks	0.079	0.083	0.077	0.071	0.044	0.054	0.049	0.040
BibSonomy - bibtex	0.096	0.102	0.096	0.090	0.050	0.59	0.057	0.043

Table 3.4: HR and ARHR computed for the Social Ranking approach

3.6.3 The Results

Table 3.4 shows the HR and the ARHR values computed for the results returned by the Social Ranking approach, baseline for our evaluation. The table shows the values computed when the feedback of 10, 20, 30 and 40 neighbors were used. For the BibSonomy - bookmarks dataset, Social Ranking obtained the best results when it used the feedback of 22 neighbors ($HR = 0.085$ and $ARHR = 0.058$). On the other hand, for the BibSonomy - bibtex dataset, we obtained the best results when we used the feedback of 23 neighbors ($HR = 0.107$ and $ARHR = 0.062$).

Social Ranking can better face the sparsity problem. The sparsity problem is due to the fact that users usually rate only a small part of the total number of the available items and this makes harder the task of finding similarities on the basis of shared resources. For this reason Social Ranking computes the similarities among two users by taking into account the number of tags they share. The rationale of this model is that if two users shares many similar tags then it is more likely that they have a common information need. However, tags are often very generic or even ambiguous. In order to face this limitation we propose in our methods to discover similarities among the users by looking at the number of shared resources. According to our approaches, the fact that users share resources is more significant in order to find people with similar interests. However, our approaches can potentially increase the sparsity since a part of the feedback of the users is not used. In order clarify this concept, the reader can think, for instance, to the case where a user apply the tag 'Java' for describing a computer programming Web site and a different user use the same tag to describe a resource about an island in Indonesia. In this case the tag is noisy and, as a result, the similarity among the users computed by Social Ranking cannot be very precise. On the other hand, by taking into account that the two users do not share resources we can discard the false neighbors.

In Table 3.5 we show the data about the accuracy of the recommendations generated by using the ACFF approach. For the BibSonomy - bookmarks dataset we obtained the best results when we used the feedback of 18 neighbors ($HR = 0.093$and $ARHR = 0.062$). On the other hand, for the BibSonomy - bibtex dataset, we obtained the best HR and ARHR values when we used the feedback of 23 neighbors ($HR = 0.111$ and $ARHR = 0.065$ with).

Also if the ACFF outperforms Social Ranking, there is not a very significant improvement. In fact, the ACFF approach uses the social semantic relations extracted from the entire folksonomy and sometimes these relations cannot be used to identify

	HR				ARHR			
	10 N	20 N	30 N	40 N	10 N	20 N	30 N	40 N
BibSonomy - bookmarks	0.081	0.089	0.083	0.075	0.046	0.059	0.053	0.048
BibSonomy - bibtex	0.098	0.109	0.098	0.090	0.053	0.062	0.058	0.049

Table 3.5: HR and ARHR computed for the ACFF approach

		HR				ARHR			
	α	10 N	20 N	30 N	40 N	10 N	20 N	30 N	40 N
BibSonomy-bookmarks	1.0	0.119	0.121	0.116	0.100	0.087	0.093	0.084	0.075
	1.5	0.124	0.133	0.129	0.123	0.093	0.107	0.101	0.089
	2.0	0.117	0.121	0.120	0.113	0.091	0.098	0.087	0.073
BibSonomy-bibtex	1.0	0.138	0.151	0.148	0.125	0.103	0.118	0.127	0.114
	1.5	0.141	0.167	0.153	0.135	0.112	0.121	0.132	0.120
	2.0	0.133	0.144	0.133	0.119	0.101	0.105	0.109	0.099

Table 3.6: HR and ARHR computed for the ACFP approach

the ToIs in an accurate way. For instance the tag '*web*' or the tag '*tool*' co-occur with many other tags in the folksonomy which are used for describing also distinct interests.

On the other hand, as showed in Table 3.6, the ACFP approach offers a more significant improvement of the accuracy of the generated recommendations.

Table 3.6 shows that by inferring topic of interests from the personomies we obtain a more adequate description of the user interests since the number of hits increases as well as the position of the hits in the lists of recommendations. In the case of the BibSonomy-bookmarks dataset, the best results were generated when we used the feedback of 18 users and α was equal to *1.5* ($HR =$ 0.138 and $ARHR =$ 0.114). By setting α to 1.5 and by combining the feedback of the top 16 neighbors we obtained the best results for the BibSonomy-bibtex dataset ($HR =$ 0.185 and $ARHR =$ 0.138). This also depends on the fact that the ACFP approach can better manage the scenario where two users labeled the same resources by using two distinct sets of tags.

For discarding the hypothesis that the observed improvements of our approaches did not occurred by chance we executed some statistical tests. The statistical tests wanted to disprove the null hypothesis that the compared approaches have the same performances and observed differences depend just on noise. More specifically we calculated a p-value, that is the probability that observed differences occurred by chance. A parametric test, the two tailed paired t-test, and a non-parametric test, the Wilcoxon test [82], were executed. The results showed that the both the ACFF and the ACFP approaches outperforms the baseline approach with, respectively, p $\leq$ 0.05 and p $\leq$ 0.01. According to these results we can confirm the hypothesis that the observed improvements of our approaches did not occurred by chance.

3.7 Conclusions

In this chapter we proposed two novel methods aimed at improving the precision of recommendations computed by a user-based CF system in social tagging systems. In particular, we showed that, when the users provide feedback about distinct ToIs, the accuracy of a CF systems can be improved by approaches which identify specific neighborhood for each ToI.

The proposed approaches implement this idea in social tagging systems where the users classify their resources by using tags. The labels assigned to the resources are used in our work to classify also the feedback of users in order to link tags and resources to interests. By associating tags and resources to interests we identify people interested in specific ToIs. Then, given a ToI we use the feedback of people interested in the specific ToI for generating the recommendations. Moreover, by classifying also the feedback of the neighbors we can also better filter their feedback in order to recommend only the resources in the ToI of the active user.

The two proposed methods mainly differ in the approach used to identify the ToIs of the active user. The first approach uses the social semantic relation extracted from the entire folksonomy whereas the second one analyzes the personomy of each user in order to infer meaningful relations among tags. The experimentation showed that the idea of adaptively selecting the neighborhood for each ToI is reasonable and the proposed approaches outperforms other state of the art mechanisms. More specifically, the approach which utilizes the social semantic relations extracted from personomies provides more accurate results. This result shows us that personomies contain meaningful information about the user interests.

At the moment, we are interested in:

- merging the ACFF approach and the ACFP approach in an hybrid recommender system.
- extending the ACFP approach following some idea proposed in [99] in order to identify hierarchical organization of user interests;
- adding a more semantic layer by means of content/ontology based analysis.

This last step could potentially empower our approaches by extracting semantic information able to disambiguate and enrich the description of user interests, merging more strictly the social and the semantic perspectives. This is the main motivation of the following chapter where we are interested in defining approaches for supporting the user of social tagging system in classifying the resources by extracting semantically reach terms and multi-terms from documents.

4

Toward a more semantic representation: keyphrase extraction

Jeff Howe defined social tagging systems as one of the main examples of *crowdsourcing* systems[79]. Coined by Howe in the June 2006, the term crowdsourcing appeared the first time in the article '*The Rise of Crowdsourcing*' [16] for defining the act of sourcing tasks traditionally performed by specific individuals (with specific competences) to an undefined large community of people (the crowd). According to the Howe's theory the technological advances can significantly reduce the gap between professionals and amateurs: people can use cheap technologies to execute complex tasks. For this reason, some tasks which are traditionally executed by experts, such as the classification task, can be exploited by a large community of people thanks to new technologies. Money is not the only way to compensate the crowd for their work: prizes, services or the intellectual satisfaction can stimulate people to put their intelligence and talent into sophisticated tasks.

The large population of users of social tagging systems are the crowd used to classify a large amount of resources instead of knowledge engineer and domain experts. Social tagging systems do not provide a monetary compensation to the taggers, but people are compensated by:

- the services provided by the social tagging systems. The social collaboration is a good mean for retrieving meaningful information since each user can enjoy the classification produced by the other peers. Moreover, by tagging resources, people can easily find the resources they classified in the past.
- The intellectual satisfaction. Users can be interested in using social systems to propagate their ideas, to influence other people and to help people with similar information needs.

Obviously, by shifting the classification task from a set of experts to a larger and not trained set of people, the results of the classification cannot be rigorous. The lack of any control or guidelines generate noisy tags (i.e. tags without a clear semantic) which lowers the precision of the user generated classifications.

How can we reduce the gap between experts and Web users? The answer to this question is still in the Howe's idea. In fact, he emphasized the concept that the technologies can reduce the gap between experts and not experts. So, for reducing the gap between knowledge engineers and users of social tagging systems we need to support the users with technologies able to simplify the classification task.

In order to reach this aim, we can support people with tools able to suggest tags which can be used to classify the Web resources in a proper way. On the other hand, in this thesis we propose to suggest multi-terms, named keyphrases, to the users on behalf of tags. More specifically, in this chapter we propose two mechanisms for extracting semantically reach keyphrases from scientific papers and Web pages. The main motivation to suggest keyphrases is that multi-terms are potentially more meaningful: a single term may be ambiguous but by taking into account a sequence of terms we can obtain more significant labels.

The research area which focuses on the extraction of structured information from unstructured and semi-structured resources is named Information Extraction (IE). However, there are meaningful differences between the IE applications, the tag recommendation task and the keyphrase extraction task. In order to show the characteristics of tag recommendation systems, IE applications and keyphrase extraction approaches, we describe these three research areas respectively in the Sections 4.1, 4.2 and 4.3.

The approaches proposed to extract the keyphrases from scientific papers and Web pages are described respectively in Section 4.4 and Section 4.5. The evaluation approaches and the results are shown in Section 4.6 and an experiment aimed at showing the benefits of extracting keyphrases is given in Section 4.7. Finally, a discussion on future works concludes the chapter in Section 4.8.

4.1 Tag Recommendation

Tag recommendations can improve the usage of social tagging applications in several ways:

- Tag suggestions can increase the probability that people will assign many tags to resources. Users can just select one or more suggested tags instead of devising from scratch to meaningful tags.

- Tag suggestions can promote a common vocabulary among users. Proposing a well-defined set of tags, it become possible to reduce the problems connected to the absence of both guidelines and supervised methodologies for the tagging process.

The set of recommended tags can be selected by taking into account only the metadata associated to the items (such as tags applied by other users) or by integrating the analysis of previous user tagging activities. Following these criteria, tag recommender systems can be divided into two classes:

1. **Not personalized tag recommender systems**. These systems select for each document a set of meaningful tags, ignoring the specific user's tagging habits. In this way, different users will receive the same suggestions for the same resource.

2. **Personalized tag recommender systems**. These systems suggest the set of the most relevant tags for a resource according to the specific user and her personal way to classify resources.

4.1.1 Not Personalized Tag Recommendations

Not personalized tag recommender systems do not respect the traditional workflow of recommender systems because they do not build and maintain a user profile. Suggested tags can be extracted both from the content of specific resources and by taking into account the tags applied by the other people in the community.

When tags are extracted from the textual content of a resource, well known techniques from information retrieval, natural language processing, and machine learning for classifying documents are applied. These approaches split the content of a textual resource into short textual slots and then they evaluate if a term in the text can be suggested as a tag. A simple approach can, for example, select the tags by taking into account the number of times the tag appear in the document. Some meaningful example of these approaches will be better discussed in Section 4.3.

Other user generated annotations can be also used to suggest tags. The simplest approach can suggest, for instance, the most popular tags for a resource. However, due to sparsity of social tagging systems there are resources tagged by only few people and for this reason more sophisticated methods have been proposed [114] [144]. AutoTag [114], a tag recommender system, suggests tags for blog posts; this framework recommends tags following a three-step process: first, it selects resources similar to the starting document (according to the TFxIDF measure) by retrieving the tags associated to these resources; then, it associates a weight to each tag according to the number of times the tag has been applied to the set of similar resources; and, finally, it suggests the top ranked tags. TagAssist [144] outperforms AutoTag thanks to a pre-processing phase, where the Porter's stemmer [125] is used to compress the set of tags.

Other approaches consider that some users produce more meaningful and semantically rich classifications than others. FolkRank [88], for example, takes in account this feature by computing a ranking for users, resources, and tags through a PageRank-like algorithm [121]. FolkRank models a folksonomy by a tripartite graph where tags, resources, and users are represented by three sets of nodes; edges link users to their tags and their resources, moreover, edges connect each resource to tags which have been used to classify the specific resource. The algorithm is based on the idea that a node of this graph is important if it is connected to many important nodes. So, the random surfer model of PageRank is used to spread weights over the tripartite graph in order to assign a weight for users, resources and tags.

4.1.2 Personalized Tag Recommendations

Personalized collaborative approaches evaluate the relevance of a tag considering the specific user tagging preferences. Personalized collaborative strategies [68] [148] use people tagging strategies to detect the set of tags which can be suggested to the active user.

In [68], the authors adapt the classical K-nearest neighbor algorithm to the task of generating a list of recommended tags: given a resource, a set of K neighbors is defined evaluating both the *UR_user_sim* and *UT_user_sim* (described in the previous chapter) over users which tagged the same resource. Tags assigned by similar users will be more relevant than others.

The ternary relation among tags, users, and items is modeled as a 3-order tensor in [148]. Latent semantic analysis is performed on tensors to capture the latent association among users, resources, and tags. This approach builds a set of quadruplets (u, r, t, likeliness) where each quadruplet describes the probability that the user u will tag the resource r with the tag t.

Personalized content-based strategies analyze the relationship between the content of a resource and the tags applied by the active user in order to predict tags for new resources. Examples of this approach are provided in [36] and in [116]. The system proposed in [36] uses a Bayesian classifier for each tag employed by the user. Each classifier is trained using the textual content of documents tagged by the specific tag. In this way the text of a new document can be used for evaluating whether a tag can be suggested for that document.

STaR (Social Tag Recommender System) [116] is based on an approach similar to AutoTag. The main difference is that STaR provides personalized tag suggestions. This framework collects two sets of documents similar to a starting resource: the set containing resources tagged by the active user and the set containing documents tagged by other users. Tags applied by the active user are weighted according to the similarity of the tagged resources to the starting one. In a similar way, a weight is assigned to tags applied by the other users. Finally, the two sets of tags are merged and a ranking of the tags is computed as a linear combination of the two scores associated to each tag.

4.2 Information Extraction

When we reported the motivation of this thesis we listed many statistics in order to highlight two facts:

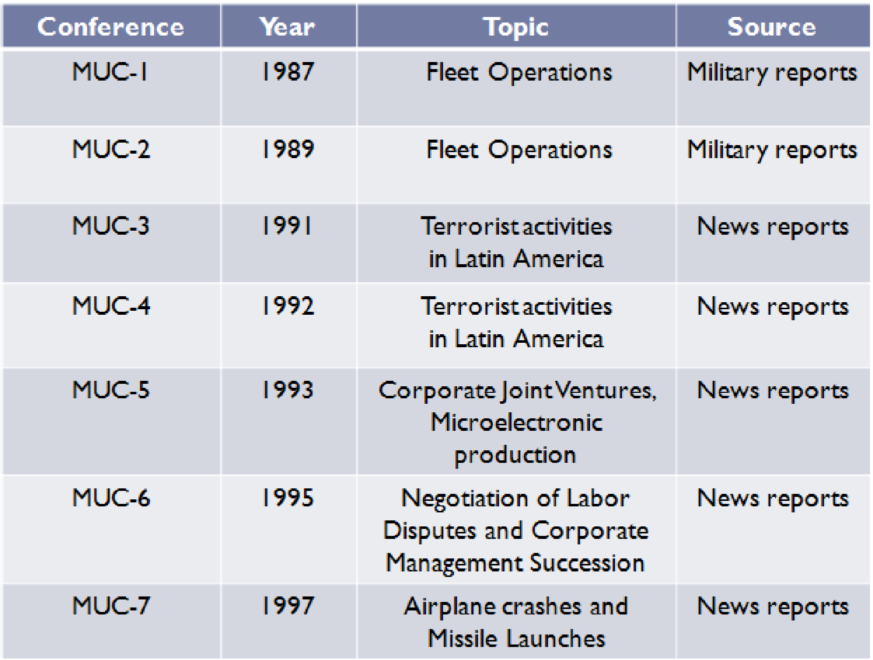
1. the World Wide Web can satisfy a very variegated set of information needs;

2. the knowledge that people need is disseminated over few resources of a very large collection of Web resources.

The huge number of available digital documents makes the Web as a bag of knowledge which can be used to provide specific knowledge about a given interest or to give just some general understandings. Following this idea we can say that the enormous size of available documents is a reasonable consequence of having a repository able to satisfy many different interests. In order to manage the resulting information overload we can use recommender systems which can filter a selected set of resources able to fill the knowledge lack of the users. However the real obstacle which recommender system have to face is not constituted by the mass of available resources since, today, we have technologies to store and manage massive sets of data. The real obstacle to an effective access to the knowledge available in Web resources is due to the fact that the knowledge is stored as unstructured free texts [109]. The task of modeling interests and the task of finding resources able to satisfy specific information need would be significantly simplified if we could move toward a fully structured representation of the knowledge available in the Web resources. In fact, as we showed in Chapter 2, the main drawback of content-based recommender systems depends on the steps which must be executed in order to have a structured representation of resources. When the recommendable resources are described by a set of knowledge engineers able to produce a precise classification of the available resources (as in the case of the catalogs of shops) we do not need preprocessing mechanism for classifying the items. On the other hand this approach does not scale up to the mass of Web document and, for this reason, we need approaches able to extract meaningful characteristics and information from the digital document on the Web. Since the knowledge stored in Web pages is mainly in the form of unstructured text, we need to overcome the limitations of natural language by translating the unstructured text into a structured and standard form. In the area of Natural Language Processing (NLP), researchers have developed several methods for generating more structured representations of the knowledge available in unstructured textual resources. These approaches are referred in literature as Information Extraction (IE) techniques [109] [58].

The first works in the area of IE were proposed in the Message Understanding Conferences (MUCs), a set a conferences held since the 1987 to the 1998 [73] organized under the supervision of the US government in the context of the DARPA project. Thanks to these conferences researchers had the opportunity to work on massive datasets focused on very specific domain (for instance, the MUC-3 and MUC-4 were based on the analysis of text about terrorism in Latin American countries whereas the MUC-7 was based on airplane crashes, and missile launches). In Table 4.1 we report the characteristics of the MUC conferences by specifying the domain topic and the text sources in the dataset used in the conference.

As the reader can see in Figure 4.1 each conference is focused on a specific topic. In fact, the IE systems are traditionally domain dependent and the task of adapting



Conference	Year	Topic	Source
MUC-1	1987	Fleet Operations	Military reports
MUC-2	1989	Fleet Operations	Military reports
MUC-3	1991	Terrorist activities in Latin America	News reports
MUC-4	1992	Terrorist activities in Latin America	News reports
MUC-5	1993	Corporate Joint Ventures, Microelectronic production	News reports
MUC-6	1995	Negotiation of Labor Disputes and Corporate Management Succession	News reports
MUC-7	1997	Airplane crashes and Missile Launches	News reports

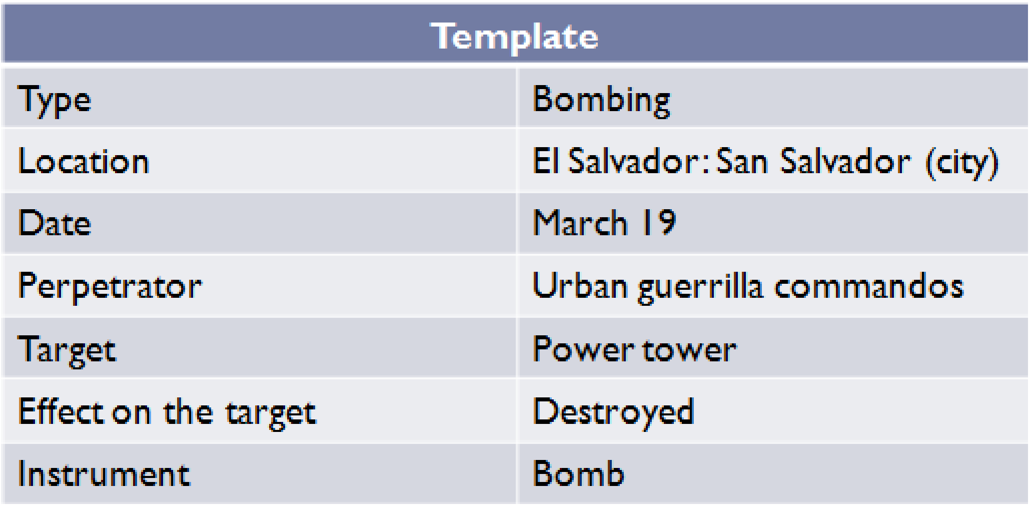
Figure 4.1: The list of the MUC conferences

existing IE mechanisms to new domains usually involve experts of the new target domain. Given a resource, IE systems extract a set of information by filling a well defined frame-like structure which is named *template*. A template is defined by a sequence of slots (one for a specific property or attribute) which must be filled with strings extracted from an input document. For example, the reader can consider the following text about the terrorist activities in Latin America:

19 March - a bomb went of this morning near a power tower in San Salvador leaving a large part of the city without energy, but no casualties have been reported. According to unofficial sources, the bomb - allegedly detonated by urban guerrilla commandos - blew up a power tower in the northwestern part of San Salvador at 0650 (1250 GMT).

A possible corresponding filled template is reported in Figure 4.2. The figure shows that slots of the template are defined according to the specific domain topic: in the example, the slots are used to model type, data, location and other information about terroristic acts. On the other hand, an IE system designed for a medical domain has to extract a completely distinct family of entities and relations such as diseases, virus, drugs, proteins, relations among diseases, connections between drugs and physical conditions of the patients.

The task of designing portable IE systems, i.e. designing IE systems which can be easily adapted to new domains, is an interesting open challenge which would simplify the task of modeling and accessing the information on the WWW [27] [28] [60] [67] [31] [59].



Template	
Type	Bombing
Location	El Salvador: San Salvador (city)
Date	March 19
Perpetrator	Urban guerrilla commandos
Target	Power tower
Effect on the target	Destroyed
Instrument	Bomb

Figure 4.2: An example of a template used in an IE application

The last MUC conference was in 1997 but in 1999 the Automatic Content Extraction (ACE) program started with the goal of generating new technologies in the area of information extraction. More specifically, the ACE program was aimed at:

- defining the research tasks to be addressed in the area of information extraction;
- providing evaluation metrics and tools for evaluating the results provided by the research groups;
- organizing research workshops for discussing about new ideas and proposal of researchers;
- building and publishing datasets for supporting both the training and the evaluation of new approaches.

4.2.1 Tasks in IE

By taking into account the challenges proposed in the MUC, five main tasks can be identified in the area of IE systems:

- **Named Entity Recognition (NER)**. The NER task, also referred as entity identification or entity extraction, is the task of identifying named entities in the text. Example of meaningful entities are for instance people, organizations and locations cited in the input text. This is a well studied task and it is also considered as one of most simple. The NER task is not influenced by the domain, i.e. NER approaches can be usually used to identify entities without regards of the specific topic. The same IE system can be used to identify entities in the medical field as well as in the financial one without losing precision. The state of the art NER systems have performances comparable to the human work: the best system proposed in the context of the MUC-7 had a precision of the 93.39% while human annotators scored 97.60%.

- **CO-reference Resolution (CO)**. The task of determining whether two expressions in natural language refer to the same entity in the text is named CO-reference Resolution (CO). This task is usually faced by humans who read a text. For example, given the text

  '*Felice and Angela went to the cinema. They enjoyed the movie.*'

  a human associates the pronoun '*They*' to the entities '*Felice*' and '*Angela*'. As the human, the CO module must look below or forward in the text for identifying these specific relations. The action of looking below is usually referred as *anaphoric reference* while the action of looking forward is named *cataphoric reference.*

  This task is domain dependent since information about the specific domain can be useful to solve co-references.

- **Template Element construction (TE)**. The Template Element task deals with merging information about an entity for creating only one description of the entity. The results of the TE task depend on the quality of the results of the NE phase and co-reference resolution phase.

  The TE task was the object of the MUC-6 and MUC-7. The best MUC-7 system scored around 80% for TE while humans achieved 93%. Since this task depends on the results of the CO task we have that this process is domain dependent too.

- **Template Relation construction (TR)**. The template relation construction task is aimed at finding relations between entities identified in the TE phase. This IE module is projected to detect a set of possible relations between the template elements. For example, this module can be developed for recognizing an employee relationship between a person and a company, a family relationship between two persons, or a subsidiary relationship between two companies. Extraction of relations among entities is a central feature of almost any information extraction task, although the possibilities in real-world extraction tasks are endless [29]. In MUC-7 the best TR scores were around 75%. TR is also a weakly domain dependent task.

4.2.2 General architecture of IE systems

Different IE systems may have different goals: some IE tools are designed for supporting directly the target user, other systems are used for generating a systematic

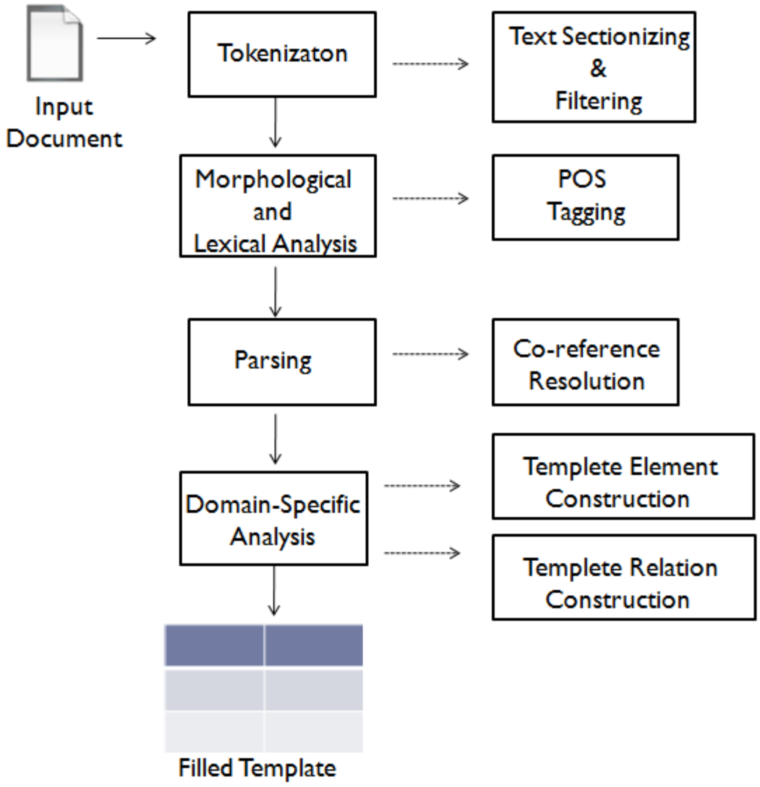

Input Document
Tokenizaton
Text Sectionizing & Filtering
Morphological and Lexical Analysis
POS Tagging
Parsing
Co-reference Resolution
Domain-Specific Analysis
Templete Element Construction
Templete Relation Construction
Filled Template

Figure 4.3: The general architecture of IE systems

representation of a collection of documents. However, it is possible to identify a general architecture for a IE system by taking into account the core activities exploited in a IE system.

As described in Figure 4.3, IE applications execute four pipelined main phases: *tokenization*, *morphological and lexical processing*, *parsing*, and *domain-specific analysis*. Here, we propose a brief discussion about these phases without providing a detailed description about technologies and approaches for implementing each module. We encourage interested readers referring to other sources in the literature to find more detailed descriptions about these modules.

The input of every IE system is one or more unstructured texts and the output is a filled template. Given a domain topic, one or more knowledge-engineers define the structure of the template by taking into account the most relevant aspects of the specific domain. According to the characteristics of the input collection and the structure of the template the following four pipelined steps are designed:

- **Tokenization**. By treating the text as sequence of characters this module identifies the elementary parts of natural language: *words*, *sentences*, *punctuation marks* and *separators*. Usually, punctuation reliably indicates sentence boundaries. However, in processing some languages like Chinese or Japanese, it is not evident from the orthography where the word boundaries are. Therefore, the characteristics of the specific language must be evaluated to realize this module. The output of this module is a list of strings named tokens.

- **Morphological and Lexical analysis**. The *Parts-Of-Speech (POS) tagging* is executed by this module by associating to the tokens some labels such as noun, adjective, verb. The POS tagging is a very meaningful activity since certain facts and entities are reported in the text by following some well-known parts-of-speech patterns.

- **Parsing**. The syntactic analysis is executed for identifying meaningful syntactic structures of the analyzed document. IE systems are only interested in specific types of information in a text (e.g., extracting only continuous noun forms) and ignore portions of text which are not relevant for their task. Therefore, parsing the portions and finding relevant grammatical relationships is necessary to filter out irrelevant information. The CO-reference resolution is implemented in this phase for overcoming this limitation.

- **Domain-specific Analysis**. The domain-specific analysis is the core of most IE systems. The syntactic and semantic information added by the previous modules is used in this phase for filling the templates. This module fills the templates, which are in general constructed as attribute-value pairs. For extracting facts, events, and relations, the system needs domain specific extraction patterns. A knowledge base contains the domain-specific expressions (e.g., instrument, perpetrator which are shown in 4.2) which are used to fill specific slots of the template.

This general architecture can be extended to other ad hoc modules according to the requirements and specific goals of the application.

4.2.3 Applications

IE are mainly used as a building block for developing intelligent Text Mining (TM) applications. In fact, also if TM is often used as synonym of IE, these two research areas have distinct goals. In fact, as we showed in the previous sections, IE approaches work on unstructured information sources (usually in a specific domain) in order to identify relevant entities and relation among the entities. On the other hand, TM mechanisms work on structured data set for inferring new knowledge. In order to apply data mining techniques, IE approaches are usually used on a collection of input texts for translating the unstructured information sources into a structured representation [90].

Since the largest part of the information available on the Web is today in the form of unstructured text, IE tools and TM approaches are often used together in a pipelined architecture as shown in Figure 4.4: IE approaches are exploited for

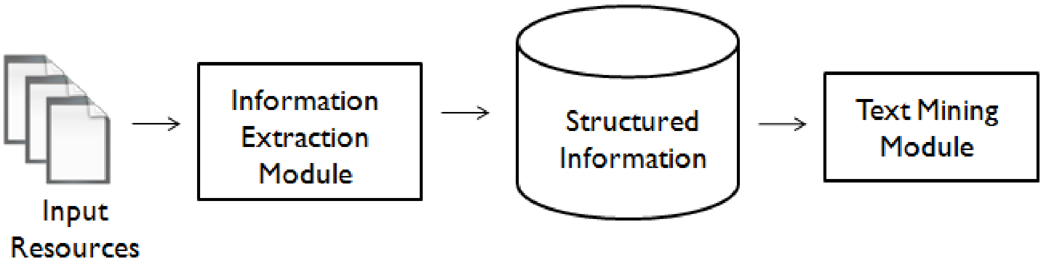


Figure 4.4: Using IE and TM approaches

obtaining the structured representation which can be used by other intelligent and sophisticated TM approaches. Recently, IE systems have been used to generate Linked Data, i.e. connected knowledge bases for storing data linked by humans in documents published on the Web. Linked Data is one of the main research directions aimed at following the Semantic Web vision. In fact, by storing entities and relations in a standard way it is possible to apply the idea of a machine readable representation of the knowledge available on the Web. In order to create such standard description of entities and relations reported in legacy Web documents we need technologies able to identify such information. IE systems can be used to face this task and there are several projects based on this idea. One of the main project is DBpedia which is focused on the extraction of structured contents from the articles published on Wikipedia. In this way, the integration of the social knowledge of the users of Wikipedia and the semantic technologies able to identify entities and relations can by integrated to have a machine readable representation of the knowledge of the Web 2.0 users. Those knowledge bases became the building blocks of other intelligent TM mechanisms such as

- Documents Clustering. The accuracy of systems for clustering textual resources strongly depends on the approach used to model the knowledge in the resources. By using metrics like the TFxIDF, the documents can be described by taking into account the frequency of the words which appear in the text. However, by integrating IE systems the documents can be also described according to some well defined dimensions. By following the example showed in Figure 4.2, the clustering approach can group the resources into explicit semantic categories and in this case the resources can be grouped according to the location of the event, according to the effect of the type of the event, and so on.

- Intelligent Information Retrieval. Entities and relations among entities can be explored in order to answer to more complex requests. For this reason, specific query languages have been proposed to navigate knowledge Bases such as Linked Data. SPARQL is one of these languages which is able to retrieve and manipulate data stored in Resource Description Framework format.

4.2.4 Building the knowledge of IE systems

As we said in Section 4.2.2, IE applications are usually domain dependent tools. In fact, in order to extract entities and relations these systems use a knowledge base which code information about the specific scenario. The knowledge base may be composed by explicit semantic information about the given domain of interest into account, but it can take into account also some syntactic aspects of the structure of the documents (since the entities could be reported in a certain parts of the resource).

According to the approach used to build the knowledge base, the large family of IE systems can be classified into two set of applications. More specifically, the IE systems can be divided into

1. **The knowledge engineering techniques**, such as PLUM [54], FASTUS [78], GE NLTOOLSET [96], and PROTEUS [133], where one or more *'knowledge engineers'* work in order to build the knowledge base of the IE system. In fact, these approaches are characterized by the development of rules defined by the *'knowledge engineers'* which are people familiar with both the requirements of the application domain and the functions of the targeted IE system.

   The engineer is not necessarily a computer scientist, but she must be able to recognize the main aspects of the specific domain. On the other hand a computer scientist can join the knowledge engineer in identifying the meaningful requirements that should be caught by the IE system. For this reason, a good practice is to provide the knowledge engineers with a significant corpus of documents in order to stimulate the process of inferring the needed rules. By using this paradigm it is possible to catch some intuitions which should otherwise be lost since the knowledge engineer is usually not an expert in defining/identifying requirements. The task of finding a good set of rules is the real core of the knowledge engineering mechanism since only the rules identified by the experts are considered by the developed system. In order to check if the identified set of rules provide accurate results another good practice is to exploit an iterative process where at each iteration the two following two steps are executed:

   (a) run the system by using the rules identified by knowledge engineers;

   (b) check if the system if the system is providing accurate results. If this does not happen, modify the rules and execute the step 1 another time.

   Obviously, this approach is based on the usage of the feedback of the knowledge engineers which must be able to suggest proper modifications of the rules.

2. **The learning techniques**. On the other hand approaches like AutoSlog [131], PALKA [93], EXDISCO [153], Snowball [27] do not use a knowledge base

explicitly generated by knowledge engineers. Since these approaches learn automatically the knowledge needed for extracting information from the digital documents they are also referred as learning approaches. In order to create the required knowledge base these systems can use statistical methods or other machine learning algorithms. However, such approaches are not properly independent from the work of a set of experts.

In fact, machine learning mechanisms need an extensive annotated corpus (a set of documents, that are pre-tagged or labeled with the required information) for identifying the relevant features which allow to accurately extracting the information. The annotated texts are the input of machine learning algorithms used to identify the characteristics useful for extracting the required information. By using the knowledge provided by the annotators, the system builds the knowledge base which can be used for gathering information from new texts of the same domain.

Since these systems are based on the usage of machine learning approaches they are also referred as *supervised learning* and meaningful example are the AutoSlog [131], the PALKA [93], the CRYSTAL [143], the LIEP [81], the WHISK [142], the RAPIER [47] and the SRV [65] systems. On the other hand *unsupervised learning* approaches aims at learning rules from a small set of annotated resources by using some bootstrapping methods [91]. These systems allow to avoid the training phase, but they usually cannot capture some latent features. Relevant example of unsupervised mechanisms are the AutoSlogTS [132], the Snowball [27], the QDIE [146] and the EXDISCO [153] systems.

4.3 Keyphrase Recommendation

The concept of keyword has been extensively used in the area of Information Retrieval. In fact keywords have been used to filter and classify documents since, as reported in [62], a keyword is a term which succinctly and accurately describes a subject or an aspect of the subject discussed in a document. A Keyword is a metadata defined by a single word term (such as '*Intelligence*' or '*Network*') which can be associated to the documents in order to simplify the task of filtering and accessing the knowledge available in a collection of digital resources. On the other hand, people are used to interact with information systems by submitting queries composed by sequences of keywords and, for this reason, they are familiar with this concept. The users of search engines, for example, do not probably know a formal definition of the concept of keyword, but they perceive it as an index which simplifies the task of finding Web resources.

The tags associated to the resources can be also defined as keywords since they still metadata which describe digital resources. However tags are socially created by humans which are not necessarily experts of the specific domain. On the other hand, in the field of Information Retrieval keywords are usually defined by a set of knowledge engineers, but also automatic systems are used in order to associate keywords to documents avoiding the manual work of the experts.

Differently from keywords, keyphrases which are metadata composed by more than one word (for instance, '*Artificial Intelligence*' or '*Social Network*' are keyphrases). According to some statistics reported in [32] and [141] the average length of a search query in the Altavista search engine querylog is equal to 2.4 terms and we showed in the previous chapter that users of social tagging systems use in average more than *3* tags for labeling a resource. This evidences that humans are used to express more than one term for defining a concept, for retrieving a document or for classifying information.

Keyphrases have been used for classifying documents according to several distinct settings. For example, the task of classifying documents by using keyphrases from a controlled vocabulary is referred as *keyphrase assignment* or *controlled indexing*. On the other hand, the *free* or *uncontrolled indexing* is the task of manually extracting the keyphrases without using a controlled vocabulary. Finally, the task of extracting the most indicative phrases from a document without referring to any vocabulary is named *keyphrase extraction*.

In this chapter, we will formulate the task of recommending tags as the task of suggesting keyphrases by extracting a meaningful set of phrases from the input documents. More specifically, given a document we want to identify a set of phrases which identify relevant entities or significant concepts reported in the document. Our approach is mainly aimed at overcoming the limitations tags as keywords which cannot capture some meanings which are defined by multi-terms. This drawback is actually recognized by the most recent release of social tagging systems such as Delicious which allow the users to create tags as multi-terms.

As showed in the previous sections, IE systems are used to extract information for populating a frame-like structure defined by a set of experts. On the contrary, a keyphrase extraction system points to extract meaningful phrases without taking into account a specific template. This means that keyphrase extraction mechanisms are more loosely coupled with a specific domain and it can be used to implement a domain independent mechanism for suggesting tags. In this work we propose to use a keyphrase extraction mechanism for suggesting annotations to the users of social tagging systems, but keyphrase extraction systems have also been used for:

- **Summarizing contents**. By identifying relevant entities and concepts reported in the textual resource, the keyphrase mechanism can summarize the contents of the resource. In this way, a reader can quickly determine if the specific document is in her interests. This can improve the user satisfaction because the users can skip the irrelevant resources by reading just the summary. Newspapers, for instance, provide a short summary of the contents of an article and similarly, scientific papers summarize the main concepts in the abstract and by using keyword (or keyphrases) at the very beginning of the document.

- **Indexing and classifying documents**. Digital libraries can be explored by using a proper index and keyphrases can be used to generate an index over a collection of resources. keyphrases can be used to support the users in browsing a collection of resources or for implementing a mechanism for searching contents [74]. An index composed by keyphrases can be also used for exploiting more sophisticated classification mechanisms by using, for example, clustering algorithms [92, 75, 44].

Given these two main functions a keyphrase extraction system can be used as a base to develop other intelligent tasks such as

- **Subject metadata enrichment**. The main goal of a search engine is to assist users in finding what they need with a minimum effort. However, search engines often return a high number of hits, and users have to manually explore and filter this amount of information. In order to simplify this manual activity, some search engines display some metadata associated to the returned results (for example document title, authors and short textual descriptions named snippet). The purpose of this metadata is to assist the users in attaining an idea of document content without actually opening the respective hit. However, this metadata is not actually rich enough for users to predict the content of the document. Since, the title always may not properly reflect the document's content. Furthermore, the snippet containing the query terms may not correctly represent the content of the document, since it is chosen to be displayed just because it contains the query terms and it can not tell whether or not the document addresses the query in a substantial way. Brook et al. [43] proposed a mechanism of enriching the metadata of the return hits by providing a set of keyphrases automatically extracted from the document for each return hit. Authors claim that, with the metadata that is augmented by keyphrases, users can predict the content of the document more easily, quickly, and accurately, and they may save a lot of time spending on downloading and examining the irrelevant documents.

- **Thesaurus creation**. A thesaurus is a set of terms that are used in a specific domain of knowledge and it is originally intended for indexing and retrieving documents, thesauri have increasingly been seen as knowledge bases and used beyond the domain of digital library. Branka et al. [95] described a method for the creation of a thesaurus in the roofing domain using keyphrases. Branka et al. utilized *Extractor* [24], a software module that extracts keyphrases from documents, in order to collect the candidate thesaurus terms from Internet sources. Authors claim that, utilizing Extractor or any keyphrase extraction software is highly advantageous in processing huge text corpora available on the Internet while eliminating irrelevant terms. The methodology used is found to be exceedingly useful, although it is not sufficient by itself for constructing a thesaurus for other domains as considerable human intervention is required.

In this work we propose to improve the quality of the user generated annotations by means of a keyphrase extraction mechanism. By integrating an automatic mechanism for suggesting keyphrases instead of tags we can:

- reduce the efforts of the users interested in classifying Web resources. Given a document, users can select a set of the proposed keyphrases instead of producing her own classification.
- obtain semantically reacher annotations. As showed in the previous chapter the task of identifying relations among tags is hard since there are not explicit semantic relations among the uni-grams provided by the users. This means that the same uni-Sgram can assume several meanings. For example, the term '*Java*' has due different meanings in the two bi-grams '*Java island*' and '*Java programming*'. The ACFP method proposed in the previous chapter try to face this limitation, but a keyphrase recommender system would simplify the task of identifying the social semantic relations in social tagging systems.

In the next paragraph we survey and classify the main approaches used to extract keyphrases from textual resources.

4.3.1 Keyphrase Extraction approaches

Keyphrase extraction techniques can be divided into two main categories: supervised approaches and unsupervised approaches.

1. *Supervised approaches* are based on the usage of machine learning mechanisms task. In this approach, a model is constructed by using a set of training documents, that have already keyphrases assigned (by humans) to them. The trained model is utilized to select keyphrases from previously unseen documents. Peter Turney was the first one to formulate the keyphrase extraction problem as a supervised learning problem. Turney proposed *GenEx* [123, 124], a hybrid genetic algorithm for keyphrase extraction, consisting of two software module named respectively *Genitor* and *Extractor*. Extractor is used to extract a set of candidate phrases. According to Turney, all the phrases having maximum length three (consecutive sequence of words) in the document which are not stop words are candidate phrases. In order to compute the frequency of the phrases, the candidate phrases are stemmed by truncation of five characters. This a common step executed by the The frequency of each candidate phrase is multiplied by it's position in the document and the result is assigned as a score to it. Scores of candidate phrases with length more than one are

boosted. Once scores are calculated, Extractor utilizes 12 numeric parameters such as relative length of a most frequent phrase, number of words in a phrase etc., to present top-ranked phrases as output. In order to determine the best parameter settings, Tureny uses Genitor, a genetic algorithm trained on documents with keyphrases-assigned.

KEA [86], another notable keyphrase extraction algorithm works on similar principles of Turney but uses a different learning technique. In the candidate identification stage, KEA first determines text sequences by setting sentence boundaries such as punctuation marks, numbers. The resulted sequences are split into tokens and extracts phrases which do not start and end with a stop word. The maximum and minimum length of a phrase can be defined by user. Each candidate phrase is stemmed and most frequent full form is used in presenting the output. In ranking stage, KEA uses two features: TFxIDF measure (a phrases frequency in a document compared to its inverse frequency in the document collection) and the position of the phrase first occurrence. A Naive Bayes classifier analyzes training data with manually assigned keyphrases and creates two sets of weights: for candidates matching manually assigned keyphrases and for all other candidates. Using the weights, the overall probability of each candidate phase is calculated. At the end, candidate phrases are ranked according to their probabilities and top ranked keyphrses (maximum number of keyphrases to be extracted per document can be set by user) are included into the final keyphrase set.

Hulth [83] introduces linguistic knowledge (i.e., *pos tags*) in determining candidate sets. For candidate identification, she combines original n-gram extraction with certain *pos-patterns* [84, 85]: 56 potential pos-patterns which extract valid noun phrases are used by Hulth in selecting candidate phrases. Hulth separates TFxIDF into two separate features: term frequency, inverse document frequency. She adopted KEA's first occurrence feature, and added a new feature that records the pos-pattern of the candidate. For final selection of keyphrases, she experimented several classifiers, including Naive Bayes, bagged decision trees and other ensembles of classifiers and shown that a combination of several prediction models yields the best results. The experimentation carried out by Hulth has also shown that, using a pos tag as a feature in candidate selection, a significant improvement of the keyphrase extraction results can be achieved.

Nguyen and Kan [117] augmented KEA with several new features such as pos patterns suggested by Hulth [84], candidate phrase suffix sequence, and a binary feature that records whether the candidate phrase is an acronym. Authors presented their algorithm exclusively for scientific publications. They utilized a classifier to capture the positions of phrases in a document with respect to logical sections such as abstract, related works and references found in scientific discourse.

Another system that relies on linguistic features is LAKE (Learning Algorithm for Keyphrase Extraction) [30]: it exploits linguistic knowledge for candidate identification and it applies a Naive Bayes classifier in the final keyphrase selection.

Since keyphrase extraction task is clearly defined and datasets are publicly available, this area is well studied in machine learning community. Many other systems have been reported [53, 157, 154, 152], but are not summarized in this thesis due to space reasons.

Turney showed that machine learning mechanisms can be effectively used in order to extract keyphrases from Web pages . In fact, in [123] he reported that the 80% of the keyphrases extracted by his mechanism were human acceptable, and more than 45% of extracted keyphrases. The main drawback of this approach is the fact that the machine learning approaches require a training corpus constituted by a set of keyphrased documents [119]. This task must be executed by collecting the feedback of a group of experts and, unfortunately, it is a time-consuming task. Since these approaches need a training set for the specific domain of interest we cannot reuse keyphrased documents provided for a different domain and, for this reason, there are just few real-word applications based on these approaches.

2. *Unsupervised approach.* Differently from supervised mechanisms, unsupervised approaches do not need a training phase. In this case, set of candidate phrases (e.g., *all words in a document*) are extracted from the input document, and a ranking strategy (e.g., *first few high frequency words*) is used to recognize the most relevant phrases.

   B&C [34] is one of the first unsupervised keyphrase extraction system. It utilizes pos information for identification of grammatically correct candidate phrases. B&C employs a dictionary in order to determine pos information of each word and extracts all nouns and, optionally their adjective or nominal modifiers. All resulted noun phrases are treated as candidate phrases. In filtering stage, B&C calculates frequency of the head noun of each candidate phrase and keep only phrases with N most frequent head nouns (N is user specified threshold). Each resulted phrase is scored by using simple heuristics based on the length of the phrases, frequency and frequency of the head noun. Finally, top K (K is user specified threshold) highest scoring phrases are selected as keyphrases. B&C has reported evaluation results involving human judges and shown the unsupervised approach they follow performs as good as the Turney's keyphrase extraction system, GenEx [123].

   In [42], Bracewell et al. extract noun phrases from a document, but cluster the terms which share the same noun term. The clusters are ranked based on term and noun phrase frequencies. Finally, top-n ranked clusters are selected as keyphrases for the document. In [102], Liu et al. propose another unsupervised method, that extracts keyphrases by using clustering techniques which assure that the document is semantically covered by these terms. Another unsupervised method that utilizes document cluster information to extract keyphrases from a single document is presented in [150].

Employing graph-based ranking methods for keyphrase extraction is another widely used unsupervised approach, exploited in [101, 80, 112].

Unlike many other systems that filter candidate phrases using weighting formulas, [112] proposes unsupervised method based on a graph-based ranking model, by representing a document as a term graph. First, all nouns and adjectives are extracted from a document and added as vertices in a document term graph. Relations are identified in between terms that co-occur each other in a pre-defined window size. These relations are used to draw edges between vertices in the graph. Vertices are weighted iteratively using TextRank, a graph based ranking model similar to the PageRank algorithm [137]. Once a score is assigned to each vertex in the graph, vertices are sorted in reversed order of their score and the top T vertices (T is set to a third of the number of vertices in the graph) are retained for post-processing stage to determine single and multi-word keyphrases. Authors are reported outperformed results with their approach comparing with Hulth's [83] supervised keyphrase extraction system. Some other systems that follow graph-based ranking model for keyphrase extraction can be traced in [101, 80].

Unsupervised methods provide accurate candidate generation techniques than supervised ones. While the majority of machine learning approaches simply extract word n-grams, heuristic methods compensate the lack of training data by complex analysis using linguistic and morphological processing [34, 42]. However, unsupervised methods do not take into consideration the particular document set characteristics (such as scientific publications or blog entries), where supervised methods do. Thus it is questionable, whether a single ranking function that is well applied for a particular document collection will perform as well on any other documents collection. However, keyphrase extraction technique is one among many other techniques we employ for tagging, and due to the simplicity and flexibility provided by unsupervised approaches, in this chapter we considered unsupervised approach for keyphrase extraction and implemented two domain-independent keyphrase extraction approaches.

4.4 Extracting Keyphrases from Scientific Paper

In this section we describe the DIKpEP (Domain Independent Keyphrase extraction for Paper) system proposed to extract the keyphrases from scientific papers.

The workflow of this unsupervised keyphrase extraction approach is shown in

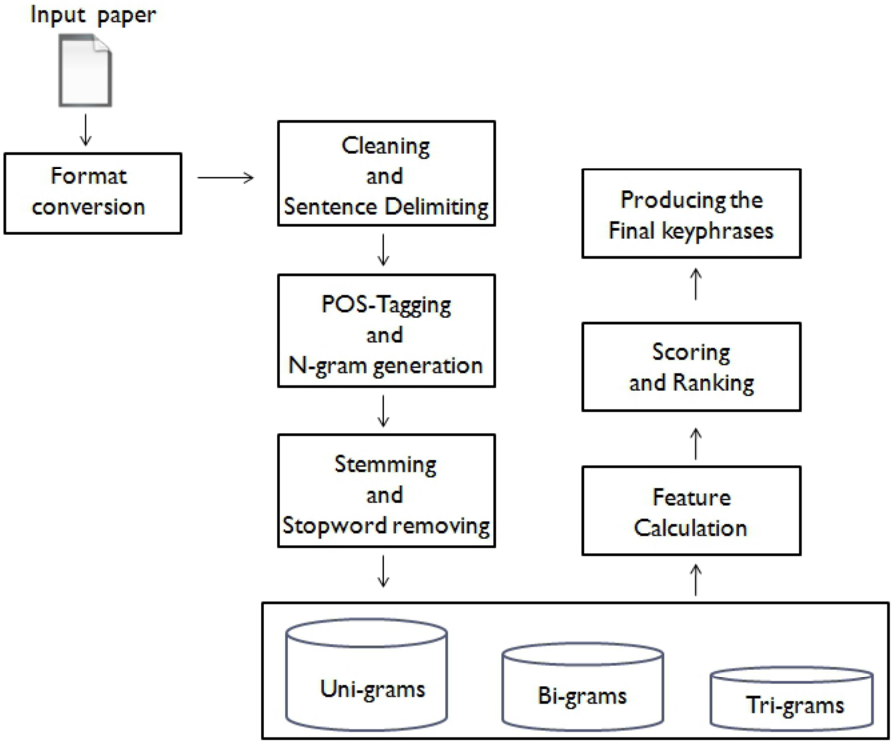

Input paper
Format conversion
Cleaning and Sentence Delimiting
POS-Tagging and N-gram generation
Stemming and Stopword removing
Uni-grams
Bi-grams
Tri-grams
Feature Calculation
Scoring and Ranking
Producing the Final keyphrases

Figure 4.5: The workflow used for extracting keyphrases from scientific papers

Figure 4.5 and, more specifically, it is organized over the three following pipelined steps:

1. **Candidate phrase extraction**. A syntactic analysis is used for identifying the candidate keyphrases, i.e a subset of n-grams extracted from the input document. This step filters out some useless n-grams according to the idea that there are some POS-patterns which do not have a significant meaning;

2. **Feature calculation**. Some characteristics of the candidate keyphrases are evaluated;

3. **Scoring and filtering**. The candidate keyphrases are scored according to the evaluated features. The score assigned to the features is used for ranking the candidate keyphrases and, finally, the top scored keyphrases are extracted.

Candidate Phrase Extraction

The candidate phrase extraction step deals with filtering the set of candidate keyphrases. For reaching this goal, the following tasks are executed:

- **Format conversion**. We assume that the input document can be in any format (e.g., *pdf*), and as our approach only deals with textual input, our system first exploits document converters to extract the text from the given input document.

- **Cleaning and sentence delimiting**. The plain text form is then processed to delimit sentences, following the assumption that no keyphrase parts are located simultaneously in two sentences. Separating sentences by inserting a sentence boundary is the main aim of this step. We have used an adequate delimiter for sentence boundary. The following heuristics are applied in setting the sentence boundaries.

  - Special symbols such as ‘.’, ‘@’, ‘_’, ‘&’, ‘/’, ‘-’, ‘” are replaced with the sentence delimiter wherever they appear in the input document, but with the following exceptions:

    * The symbols ‘.’, ‘@’, ‘_’, ‘&’, ‘/’, ‘-’ are allowed if they are surrounded by letters or digits (e.g., *e-commerce*, *hiperlan/2*).
    * The symbol ‘” is allowed if it is preceded by a letter or digit (e.g., *pearson's correlation*).

  - Other punctuation marks (e.g., ‘?’, ‘!’) are simply replaced by sentence delimiter,
  - Apostrophes are removed and the entire input text is converted into lowercase.

  The result is a set of sentences each containing a sequence of tokens, bounded by the sentence delimiter.

- **POS tagging and n-gram extraction**. We assign a pos tag (noun, adjective, verb etc.) to each token in the cleaned text, by using Stanford log-linear part-of-speech tagger[1]. The Stanford pos tagger uses 36 types[2] of pos tags. The assigned pos tags are later utilized in the filtration of candidate phrases and calculation of pos value feature. The next step in our procedure is to extract n-grams. We have observed that in the dataset utilized for the experimentation, phrases that are constituted by more than 3 words are rarely assigned as keyphrases, so, in our process, we set the value of ‘n’ to the maximum value 3. We extract all possible subsequences of phrases up to 3 words (uni-grams, bi-grams, and tri-grams).

- **Stemming and Stopword removing**. From the extracted n-grams, we remove all phrases[3] that start and/or end with a stopword and phrases containing the sentence delimiter. Partial stemming (i.e., unifying the plural forms and singular forms which mean essentially the same thing) is performed using

[1] http://nlp.stanford.edu/software/tagger.shtml.

[2] Pos tagging follows the Penn Treebank tagging scheme.

[3] In our use of this term, a phrase is n-gram with n = 1, 2, 3

the first step of Porter stemmer algorithm [125]. To reduce the size of the candidate phrase set, we have filtered out some candidate phrases by using their pos tagging information. Uni-grams that are not labeled as noun, adjective, and verb are filtered out. For bi-grams and tri-grams, only pos-patterns defined by Justeson and Katz [89] and other patterns that include adjective and verb forms are considered.

- **Separating n-gram lists**. Generally, in a document, uni-grams are more frequent than bi-grams, and bi-grams are more frequent than tri-grams and so on. In the calculation of phrase frequency (explained in the next step) feature, this shows a bias towards n-grams which are having small value of 'n'. In order to solve this problem, we have separated n-grams of different lengths (n = 1, n = 2, n = 3) and arranged them in three different lists. These lists are treated separately in calculation of feature sets and in final keyphrase selection. As a result of step 1, we obtain a separate list of uni-gram, bi-gram, and tri-gram candidate phrases (with corresponding pos tags) per document after the proper stemming and stopword removal phases.

Feature Calculation

The feature calculation step characterizes each candidate phrase by statistical and linguistic properties. Five features for each candidate phrase are computed:

1. ***Phrase frequency***: this feature is the classical term frequency (tf) metric, utilized in many state of the art keyphrase extraction systems [123][83][85], in our use of this feature, instead of calculating it with respect to the whole length of the document, we compute it with respect to each n-gram list. With a separate list for each n-gram in hand, the phrase frequency for phrase P in a list L is:

$$frequency(P, L) = \frac{freq(P, L)}{size(L)}$$

   where:

   - $freq(P, L)$ is the number of times P occurs in L;
   - $size(L)$ is the total number of phrases included in L.

2. ***Pos value***: as described in [83][34], most author-assigned keyphrases for a document turn out to be noun phrases. For this reason, in our approach, we stress the presence of a noun in a candidate phrase while computing a pos value for the phrase. A pos value is assigned to each phrase by calculating the number of nouns (singular or plural) normalizing it by the total number of terms in the phrase. For instance, in a tri-gram phrase, if all tokens are noun forms, then the pos value of the phrase is 1, if two tokens are noun forms, then the pos value is 0.66, and if one noun is present, the value is 0.33. All

remaining phrases which do not include at least one noun form are assigned the pos value 0.25. The same strategy is followed for bi-gram and uni-gram phrases.

3. ***Phrase depth***: this feature reflects the belief that important phrases often appear in the initial part of the document especially in news articles and scientific publications (e.g., *abstract, introduction*). We compute the position in the document where the phrase first appears. The phrase depth value for phrase P in a document D is:

$$depth(P, D) = 1 - \left[\frac{first\_index(P)}{size(D)}\right]$$

where:

- $first\_index(P)$ is the number of words preceding the phrase's first appearance;
- $size(D)$ is the total number of words in D.

The result is a value in $[0, 1]$. Highest values represent the presence of a phrase at the very beginning of the document.

4. ***Phrase last occurrence***: we give also importance to phrases that appear at the end of the document, since keyphrases may also appear in the last parts of a document, as in the case of scientific articles (i.e., in the conclusion and discussion parts). The last occurrence value of a phrase is calculated as the number of words preceding the last occurrence of the phrase normalized with the total number of words in the document. The last occurrence value for phrase P in a document D is:

$$last\_occurrence(P, D) = \frac{last\_index(P)}{size(D)}$$

where:

- $last\_index(P)$ is the number of words preceding the phrase's last appearance;
- $size(D)$ is the total number of words in D.

5. ***Phrase lifespan***: the span value of a phrase depends on the portion of the text that is covered by the phrase. The covered portion of the text is the distance between the first occurrence position and last occurrence position of the phrase in the document. The lifespan value is a computed as the difference between the *phrase last occurrence* and the *phrase first occurrence* values as follows:

$$lifespan(P, D) = \frac{[last\_index(P) - first\_index(P)]}{size(D)}$$

where:

- $last\_index(P)$ is the number of words preceding the phrase's last appearance;
- $first\_index(P)$ is the number of words preceding the phrase's first appearance;
- $size(D)$ is the total number of words in D.

The result is a value between 0 and 1. Highest values mean that the phrase is introduced at the beginning of the document and carried until the end of the document. Phrases that appear only once through out the document have the lifespan value 0.

In the end, we get a feature vector for each candidate phrase in the three n-gram lists.

Scoring and Ranking

In this step a score is assigned to each candidate phrase which is later exploited for the selection of the most appropriate phrases as representatives of the document. We call the resulting score value as *keyphraseness* of the candidate phrase. More technically, the keyphraseness of a phrase P with non empty feature set $\{\mathrm{f}_1,\mathrm{f}_2,...,\mathrm{f}_5\}$, with non-negative weights $\{\mathrm{w}_1,\mathrm{w}_2,..,\mathrm{w}_5\}$ is:

$$keyphraseness(P) = \frac{\sum_{i=1}^{5} w_i f_i}{\sum_{i=1}^{5} w_i}$$

In the first stage of our research, we assigned equal weights to all features, yielding to the computation of the average. Therefore:

$$keyphraseness(P) = \frac{1}{n}\sum_{i=1}^{n} f_i,$$

where:

- n is the total number of features (i.e., 5 in our case);
- f_1 is the phrase frequency;
- f_2 is the phrase pos value;
- f_3 is the phrase depth;
- f_4 is the phrase last occurrence;
- f_5 is the phrase lifespan.

A feature could have more impact than others on keyphraseness and influences how candidate phrases will be selected, as the features have not the same nature (e.g., frequency vs. pos value, depth vs. lifespan). To compensate this phenomena different weights are assigned to each feature. For weight calculation, we are proposing a novel approach that computes associate weights to features by examining the already existing ground truth author-assigned keyphrases. For this, we utilized a publicly available keyphrase extraction dataset[4][117] which contains 215 full length scientific documents from different computer science subjects. Each document in the dataset contains a first set of keyphrases assigned by the paper's authors and a second set of keyphrases assigned by volunteers, familiar with computer science papers. We considered author assigned and volunteer assigned keyphrases as ground truth keyphrases for the documents. For these keyphrases, the five feature values are computed as showed before. Since DIKpEP is capable of extracting only phrases that are explicitly stated in the documents, we extracted from each document the keyphrases such that: the keyphrase was in the corpus of the paper; the keyphrase was assigned by the authors or by the volunteers. Moreover, for each extracted keyphrase we also computed the values of the 5 features used by the DIKpEP mechanism for scoring the keyphrase. Following this procedure we obtained 1000 keyphrases with corresponding feature values are computed where highest values (i.e., near to 1) represent the goodness of the feature. We extracted and evaluated only the features of the manually assigned keyphrases since we assume that these keyphrases are semantically significant. On the other hand, we cannot assume that the volunteers labeled with all the meaningful keyphrases in the document and, for this reason, we considered only the manually assigned keyphrases. The 1000 keyphrases with five feature values are represented in matrix form as follows:

$$F = \begin{bmatrix} f_{(1,1)} & f_{(1,2)} & f_{(1,3)} & f_{(1,4)} & f_{(1,5)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{(j,i)} & f_{(j,i)} & f_{(j,i)} & f_{(j,i)} & f_{(j,i)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{(1000,1)} & f_{(1000,2)} & f_{(1000,3)} & f_{(1000,4)} & f_{(1000,5)} \end{bmatrix}$$

where $f_{(j,i)}$ represents the value of the ith feature calculated for the jth extracted keyphrase, $i = \{1, \ldots, 5\}$, $j = \{1, \ldots, 1000\}$. Each row of the matrix is associated to a specific keyphrase extracted from a specific document. For each feature f_i the mean μ_{f_i} and the variance $\sigma^2_{f_i}$ of the vector $\left[f_{(1,i)}, \ldots, f_{(j,i)}, \ldots, f_{(1000,i)}\right]$ are calculated as follows:

$$\mu_{f_i} = \frac{\sum_{j=1}^{1000} f_{(j,i)}}{1000}$$

$$\sigma^2_{f_i} = \frac{\sum_{j=1}^{1000} f^2_{(j,i)}}{1000} - \mu^2_{f_i}$$

[4] http://wing.comp.nus.edu.sg/downloads/keyphraseCorpus/

The mean reflects the central tendency of the feature and the variance reflects the variability of the feature values with respect to the mean. Obviously, the feature having high mean and low variance contributes maximum to the final keyphrase result. The weight for f_i is computed by simply dividing the mean with the variance.

In the equation form, the weight of f_i is:

$$weight(f_i) = \frac{\mu_{f_i}}{\sigma^2_{f_i}}$$

The weights are normalized and assigned to the features. Our weight calculation approach produced the highest weight to the phrase depth feature, then second highest is given to pos value, and so on for last occurrence, lifespan, frequency features, respectively. The final weights are shown in Table 4.1.

Feature Name	Weight
phrase frequency	0.10
pos value	0.30
phrase depth	0.32
phrase last occurrence	0.16
phrase lifespan	0.12

Table 4.1: Final weights assigned to the features of the DIKpEP approach

Producing Final Keyphrases. The scoring process produces three separate lists L_1, L_2, and L_3 containing respectively all the uni-grams, bi-grams and tri-grams with their keyphraseness values. We then select some keyphrases, which are considered to be the most important from each list. In [97], Kumar&Kannan proposed a strategy for selecting final keyphrases from n-grams, after an extensive statistical analysis regarding the length of the author assigned keyphrases for scientific documents. In this paper, the documents we are utilizing for the experiment are scientific in nature, in order to produce the *'k'* final keyphrases, we have followed the same strategy that is proposed and utilized in [97]. In every list, the candidate phrases are ranked in descending order based on the keyphraseness values. Top 20% (i.e., 20% of *'k'*) keyphrases are selected from "L_3", Top 40% (i.e., 40% of *'k'*) are selected from "L_2", and remaining 40% of rest of *'k'* keyphrases are selected from "L_1'. In this way top k keyphrases for the given document are extracted and merged in the final list of extracted keyphrases. This list is finally ordered by taking into account the keyphraseness value.

4.5 Extracting Keyphrases from Web Pages

We extended the DIKpEP in order to have a new domain independent named DIKpEW (Domain Independent Keyphrase extraction for Web pages) which can extract meaningful keyphrases from the Web pages. The domain independent approach proposed to extract keyphrases from scientific papers works under two main assumptions:

1. **Scientific papers are usually written in English**. This simplifies the analysis of the textual content since we can utilize a POS-Tagger for the English language without taking into account the characteristics of other languages. Moreover, the structure of the English language allows us to identify a specific set of POS-patterns in order to filter meaningful multi-terms from the text.
2. **Scientific papers organize their contributions according to a well-defined schema**. The abstract, the introduction and the conclusion are the sections where the authors usually summarize the goals, the issues and the findings of the work. For this reason, we assign a score to each keyphrase by evaluating the position of the keyphrase in the text: it is plausible that keyphrases in the first part and in the last section of the paper better describe the resource.

These two assumptions are not true when we have to extract the keyphrases from Web pages. In fact, many Web pages submitted on social bookmarking Web sites are not in English and, moreover, the Web pages do not follow the structure adopted by scientific papers.

In order to extract keyphrases from Web pages we extended the workflow described in the previous section by:

1. modifying the module used to format the textual contents of the Web page;
2. modifying the set of features used to rank the candidate keyphrases.

The resulting workflow is shown in Figure 4.6 where the red pieces of the chart represents the new and modified modules. In the following subsections we are going to describe the proposed approach by showing the functionalities executed by each module.

Format conversion

The format conversion module has three main goals:

1. **Filtering the unrelevant parts of the document**. Unfortunately, the main contents of the Web pages are mixed with other textual parts such as headers, footers or comments which are completely unrelevant.

   In order to discard the these useless and noisy parts from the Web page we used an open source Web service named Boilerpipe[5]. The Boilerpipe service,

[5] http://code.google.com/p/boilerpipe/

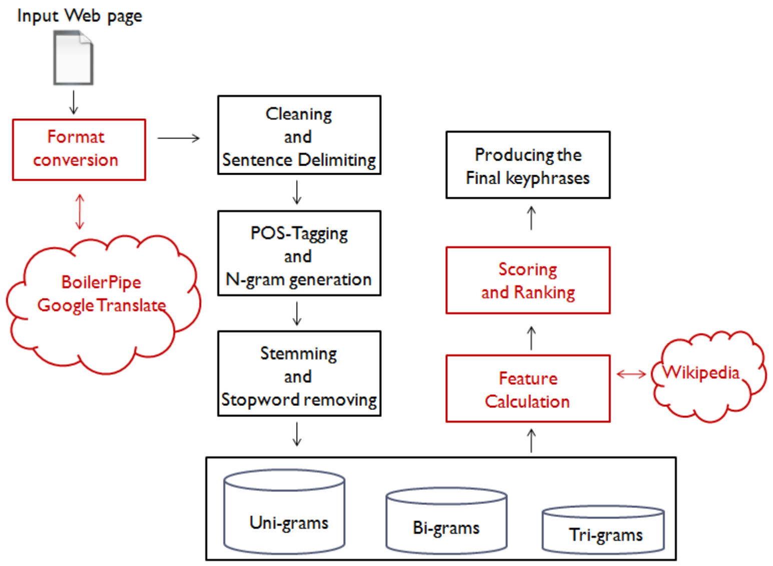


Figure 4.6: The workflow used for extracting keyphrases from Web pages

developed by some researchers from the L3S Research Center of Hannover, can remove the '*surplus*' text from a Web page. Given a Web page, Boilerpipe returns the main article in the Web page by discarding other information (banner, footers, advertisement, etc.).

2. **Extracting metadata defined by the authors of the Web page**. HTML pages are often enriched by their authors with some labels and summaries stored in the HTML tags (such as the keywords, the description and the title tags).

3. **Translating the text into the English language**. We cannot assume that Web pages are always written in English. In order to use the POS-Tagger as well as the POS-Patterns adopted for filtering the candidate keyphrases we translate the text extracted by the Boilerpipe service in English. More specifically we used the Google Translate Api in order to recognize the input language and to translate (if it is necessary) the text in English.

The output of the Format conversion module is a text in English composed by the title of the Web page, followed by the metadata extracted from the HTML tags, and concluded by the text extracted by the Boilerpipe service.

Feature Calculation

In order to identify the new set of features, we have to take into account the characteristics of Web pages. Differently from the case of scientific papers, Web pages do not follow a strict organization of the contents since there is not an abstract or a conclusion paragraph. However, Web pages usually contain some relevant information at the very beginning of the page. This is particularly true in the case of newspapers where a short summary of the news usually follows the title. This is still true in the majority of the Web sites where it is common to provide a general description of the contents in the first lines of the document. By taking into account these characteristics we keep three of the features used in DIKpEP for extracting the keyphrases from scientific papers, that are:

- ***Phase frequency***. As in the scenario of scientific publications the number of times that a keyphrase appears in the document is evaluated for identifying meaningful terms and multi-terms.

- ***Pos value***. Following the same rationale of the DIKpEP approach, we still take into account the Pos value feature.

- ***Phrase depth***. The phrase depth is evaluated in order to assign a higher score to keyphrases which appear in the first part of the Web page. As we said, authors of Web pages often provide meaningful concepts in the first lines of the document.

These features are computed exactly as showed for the DIKpEP mechanism. The reader can notice that we do not evaluate two of the features of the DIKpEP approach: the last occurrence and the life span. This is due to the fact that Web pages usually do not have a conclusion or a final section for reporting the main concepts or findings.

On the other hand, we introduced the following four boolean feature:

- ***Title***. It checks if a given keyphrase is in the title of the Web page. We followed the hypothesis that the title summarizes meaningful concepts which are more deeply discussed in the main article. For each keyphrase we computed a boolean feature which is equal to 1 if the keyphrase is in the title of the Web page, 0 otherwise.

- ***Description***. The authors of Web pages often add a short description of the main contents of the Web page by using the 'description' HTML tag. According to the idea that the summary provided by the author may contain meaningful information we compute this boolean feature for each keyphrase: the feature is equal to 1 if the keyphrase is in the description, 0 otherwise.

- ***Keyword***. Also if authors of Web pages are not forced to classify their published resources they usually add some keywords in order to be indexed by search engines. Since these terms are labels generated by the same authors of the Web pages we consider these keyphrases as meaningful multi-terms. The keyword feature is then computed as a boolean value which is equal to 1 if the keyphrase is one of the keyword defined by the author of the Web page, 0 otherwise.

- ***Wikipedia***. As the POS value feature is used to filter more significant terms and multi-terms, the Wikipedia feature is also used to assign an higher score to the keyphrases which have (more probably) a well-defined meaning. More specifically, we assume that it is more likely that the keyphrase is associated to a certain well-known meaning if there is a Wikipedia page for a given keyphrase. The Wikipedia feature is then equal to 1 if Wikipedia has a page for describing the keyphrase, 0 otherwise.

Scoring and Ranking

In order to compute the Keyphraseness of the candidate keyphrases we computed a weighted combination of the evaluated features. More specifically, a very preliminary experimentation was executed for defining a proper set of weights for the features. We could not exploit an off-line analysis for learning these weights in a more automatic way due to the lack of a dataset of Web pages annotated with keyphrases. By using this preliminary experimentation we assigned to the features the weights reported in Table 4.2:

Feature Name	Weight
phrase frequency	0.5
pos value	0.5
phrase depth	0.6
title	0.9
description	0.6
keyword	0.6
wikipedia	0.9

Table 4.2: The weights assigned to the features of the DIKpEW approach

These weights are used to compute and order the lists of uni-grams, bi-grams and tri-grams similarly to the DIKpEP approach.

4.6 Evaluation

This section describes the executed evaluation and reports the accuracy for the two keyphrase extraction systems presented in this chapter. The two approaches have been evaluated by following two different strategies due to the different characteristics of the available datasets.

In fact in order to evaluate the accuracy of the keyphrases extracted from the scientific papers we used a set of articles annotated by the corresponding authors. In this case, we could compute the precision of the results by counting the number of filtered keyphrases which have been both filtered by our mechanism and used by the authors to classify the paper.

On the other hand, Web pages are not labeled by the corresponding authors. For this reason, we exploited a live evaluation by involving real users who evaluated the accuracy of the results.

4.6.1 Evaluating the DIKpEP approach

In order to compute the accuracy of the DIKpEP mechanism we tested it on a publicly available keyphrase extraction dataset [117]. We evaluated the precision of the results by computing the number of matches between the keyphrases attached to the document and the keyphrases extracted automatically. The same partial stemming strategy that is exploited in candidate phrase selection is used also in matching keyphrases. For instance, given the following keyphrase sets S_1 {*component library, facet-based component retrieval, ranking algorithm, component rank, retrieval system*} and S_2 {*component library system, web search engine, component library, component ranks, retrieval systems, software components*} suggested by our system, the number of exact matches is 3 corresponding to {*component library, component rank, retrieval system*}. More specifically, we have carried out two experiments in order to evaluate the accuracy of the DIKpEP system.

Evaluating DIKpEP: Experiment 1

For the first experiment, we have considered keyphrase extraction works presented by Nguyen and Kan in [117] and KEA [86] as baseline systems. From the available 215 scientific paper, Nguyen and Kan have taken 120 documents to compare these with KEA. The maximum number of keyphrases for each document (i.e., *'k'*) is set to ten in Nguyen and Kan. We have taken their results [117] as reference, and in the first experiment we have worked on 120 documents randomly selected from the 215 documents. Since the authors of [117] did not provided the list of the paper used to train their system we had to repeat the experimentation several times by selecting different sets of papers as training set by using a 10-cross validation technique. In both the experiments, we removed the bibliography section from each document in the dataset in order to better utilize the *phrase last occurrence* feature. Table 4.3 shows the average number of correct keyphrases of the three algorithms when 10 keyphrases are extracted from each document: the first row shows the average number of correct keyphrases i.e., 3.03 suggested by KEA, the second row shows the average number of correct keyphrases i.e., 3.25 suggested by Nguyen and Kan, the third rows shows the average number of correct keyphrases i.e., 4.75 suggested by our system DIKpEP when equal weights are assigned to the features, whereas the last row shows the average number of correct keyphrases i.e., 5.04 suggested by DIKpEP after assigning associate weights to the features. In either of the cases, our system significantly outperforms the other two.

In order to prove the strength of our results, some significance tests have been executed to verify whether the observed differences among KEA, DIKpEP before

System	Average # of exact matches
KEA	3.03
Nguyen&Kan	3.25
DIKpEP (before assigning weights)	4.75
DIKpEP (after assigning weights)	5.04

Table 4.3: Comparing DIKpEP to the state of the art approaches

assigning weights and DIKpEP after assigning weights are truly meaningful or occurred by chance. The statistical tests want to disprove the null hypothesis that the compared approaches have the same performances and observed differences depend just on noise. To obtain this result statistical tests have to calculate a p-value, that is the probability that observed differences occurred by chance. In particular, both a parametric test, the two tailed paired t-test, and a non-parametric test, the Wilcoxon test [82], have been executed proving that DIKpEP before assigning weights outperforms KEA in a statistically significant way (p-value ≤ 0.01) when five, seven or ten tags are returned. Moreover, DIKpE after assigning weights always outperforms DIKpEP, in a statistically significant way, DIKpEP before assigning weights and KEA with a p-value ≤ 0.01.

Evaluating DIKpEP: Experiment 2

For the second experiment, we have extracted keyphrases for all 215 documents and compared our approach exclusively with the results provided by KEA. KEA is publicly available and considered as an emerging benchmark for evaluation in literature. The source code for KEA is available for download[6] and we have utilized a total of 70 documents (with keyphrases assigned by authors) extracted from the 215 documents dataset to train the KEA algorithm. For each document, we extracted 7, 15 and 20 top keyphrases using our approach and KEA.

Extracted	Average number of correct keyphrases		
Keyphrases	KEA	DIKpE(before assigning weights)	DIKpE(after assigning weights)
7	2.05	3.52	3.86
15	2.95	4.93	5.29
20	3.08	5.02	5.92

Table 4.4: Performance of DIKpEP when the features are weighted

The results are shown in Table 4.4 and graphically represented in Figure 4.7. In the graph, the lowest line describes the performances of KEA which, on average, correctly recommends respectively 2.05, 2.95, 3.08 keyphrases when it returns 7, 15 and 20 keyphrases. The precision of our approach both before assigning weights

[6] http://www.nzdl.org/Kea/download.html.

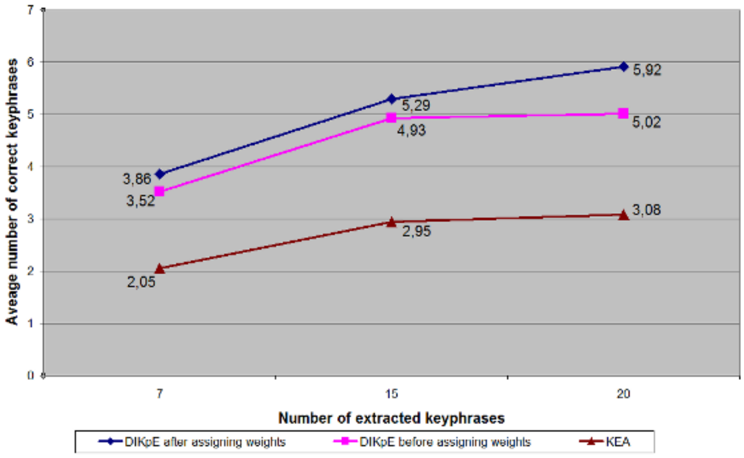


Figure 4.7: Performance of DIKpEP compared to KEA

(the central line) and DIKpEP after assigning weights (the upper line) outperform significantly the results of KEA, where DIKpEP after assigning weights gives the best results. It is clear that even though our system does not undertake any training activity, it greatly outperforms KEA performance.

A sample output of the DIKpEP system for three sample documents is shown Table 4.5. For each document top seven keyphrases extracted by DIKpEP are presented: keyphrases that are assigned by the document authors are shown in normal font, Italics indicates keyphrases that are assigned by volunteers, and boldface in DIKpEP's row (third row) shows keyphrases that have been automatically extracted and matched with author or volunteer assigned keyphrases. Even if some keyphrases of DIKpEP do not match with any of the keyphrases, they are still correctly related to the main theme of the document.

4.6.2 Evaluating the DIKpEW approach

Differently from the case of scientific papers, Web pages are usually not classified with keyphrases by their authors. So, in order to evaluate the results returned by the proposed approach, we exploited a live evaluation involving real users who judged the accuracy of the extracted keyphrases.

Due to the lack of keyphrases associated to Web pages, we could not use KEA for comparing our results to one of the state of the art mechanisms. In fact, the KEA mechanism needs to be trained by using a corpus of annotated documents. Moreover, at the best of our knowledge there is not another freely available API for extracting keyphrases from Web pages. In order to overcome this limitation we decided to use as baseline approach a system where keyphrases are scored and ranked according to the frequencies of the keyphrases. This choice seems reasonable since, as the DIKpEW system, the baseline approach takes into account only the information available in a specific document (without considering the characteristics

Document	#26. Accelerating 3D Convolution using Graphics Hardware.	#57. Contour-based Partial Object Recognition using Symmetry in Image Databases.	#136. Measuring e-Government Impact: Existing practices and shortcomings.
keyphrases assigned by the document authors	**convolution** hardware acceleration volume visualization	**object** image **contour** recognition **symmetry**	**e-government** law interoperability architectures **measurement** evaluation
keyphrases assigned by volunteers	***3D convolution*** ***filtering*** ***visualization*** ***volume rendering***	*occlusion* ***object recognition*** ***symmetry*** *contour* *estimation*	*benchmark* ***measurement*** ***e-government*** ***public administration*** ***business process***
keyphrases assigned by DIKpE system	high pass filters **volume rendering** filter kernels **3d convolution** **convolution** **visualization** **filtering**	partial object recognition **object recognition** objects in images occlusion of objects **objects** **symmetry** **contours**	measuring e-government impact **business process** e-governmental services **public administration** **e-government** **measurement** business

Table 4.5: Top seven keyphrases extracted by DIKpEP system from three sample documents

of the documents in a specific collection). This baseline mechanism is still domain independent and the results are not biased by the characteristics of a specific corpus.

More specifically, the baseline mechanism assigns a score to the set of candidate keyphrases according to the frequency of the keyphrases: the most frequent keyphrases obtain an higher score. By using the score assigned to the keyphrases, the baseline mechanism can extract: the two top scored uni-grams, the five top scored bi-grams and the three top scored tri-grams. The final set of keyphrases is composed by these 10 filtered keyphrases: the filtered keyphrases are merged and ordered by taking into account the score associated to them.

The results returned by both our mechanism and the baseline approach were evaluated by using a Web application where a set of volunteers judged the accuracy of the results. Since our approach is mainly aimed to support the users of social tagging systems we created a Web based application which simulates the interaction of a user with a social tagging system. By using this application, the authenticated users submitted an URL and then the evaluation framework returned to the users a list of keyphrases for the specific document. The list of returned keyphrases was composed by the results produced by both the DIKpEW approach and the baseline mechanism. In fact, when the users submitted an URL the baseline approach and our system were used to extract the two ranked list. Then, the two set of keyphrases were presented to the evaluators in a random order as showed in Figure 4.8.

By merging and presenting the keyphrases without a specific order we avoided to influence the human evaluators who could not recognize the keyphrases returned by one of the two compared approaches.

The evaluators had to vote each returned keyphrase by using the following 5-Likert scale:

- **Excellent**. The keyphrase is very meaningful. It reports relevant facts, people, topics or other elements which characterize the Web page.

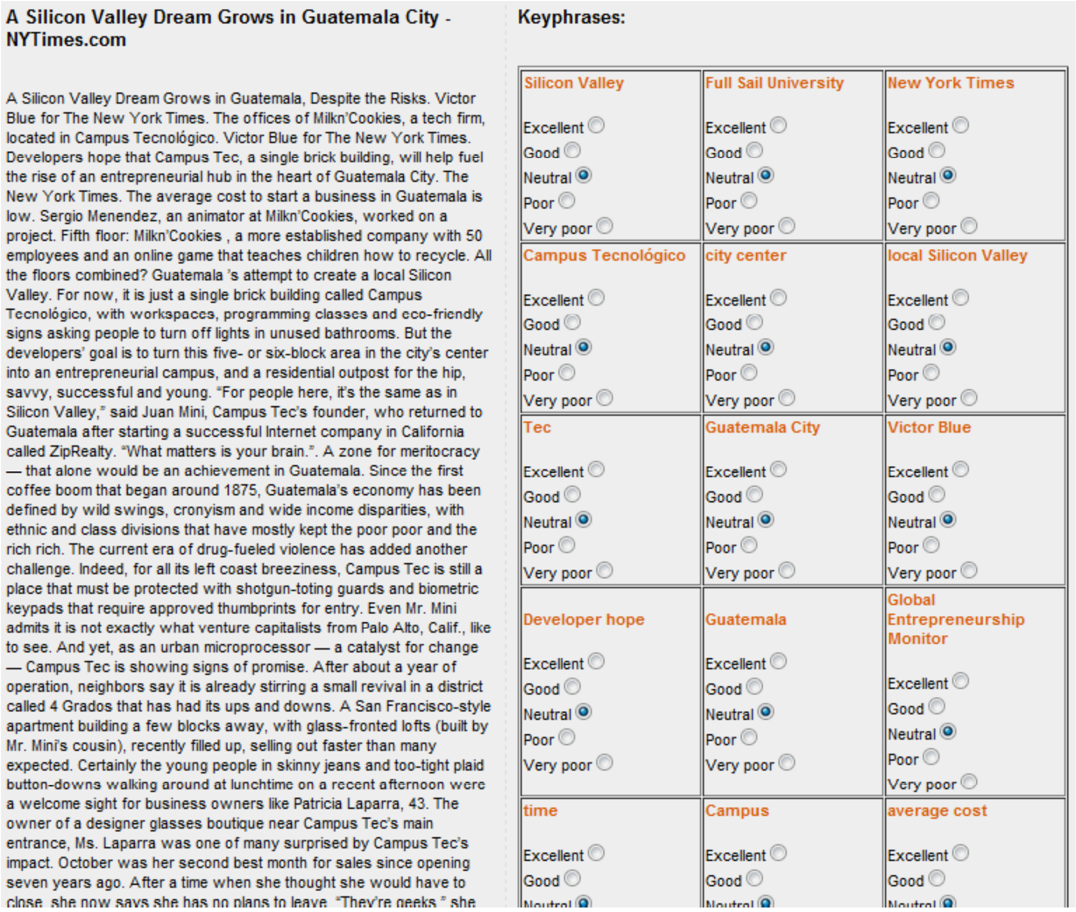
- **Good**. The keyphrase is still significant for classifying the document but it is not the best. The keyphrase reports facts, people, topics or other elements which characterize the Web page which are more weakly connected to the main content of the page.

- **Neutral**. You are not sure about the significance of the keyphrase for the document.

- **Poor**. The keyphrase does not properly describe the contents.

- **Very Poor**. The keyphrase does not make sense.

The evaluation involved 26 volunteers (20 men and 6 women): the volunteers were students and workers where the oldest participant was 63 years old, the youngest was 22 years old and the average age was 37 years. By using our Web application,



A Silicon Valley Dream Grows in Guatemala City - NYTimes.com

A Silicon Valley Dream Grows in Guatemala, Despite the Risks. Victor Blue for The New York Times. The offices of Milkn'Cookies, a tech firm, located in Campus Tecnológico. Victor Blue for The New York Times. Developers hope that Campus Tec, a single brick building, will help fuel the rise of an entrepreneurial hub in the heart of Guatemala City. The New York Times. The average cost to start a business in Guatemala is low. Sergio Menendez, an animator at Milkn'Cookies, worked on a project. Fifth floor: Milkn'Cookies , a more established company with 50 employees and an online game that teaches children how to recycle. All the floors combined? Guatemala 's attempt to create a local Silicon Valley. For now, it is just a single brick building called Campus Tecnológico, with workspaces, programming classes and eco-friendly signs asking people to turn off lights in unused bathrooms. But the developers' goal is to turn this five- or six-block area in the city's center into an entrepreneurial campus, and a residential outpost for the hip, savvy, successful and young. "For people here, it's the same as in Silicon Valley," said Juan Mini, Campus Tec's founder, who returned to Guatemala after starting a successful Internet company in California called ZipRealty. "What matters is your brain.". A zone for meritocracy — that alone would be an achievement in Guatemala. Since the first coffee boom that began around 1875, Guatemala's economy has been defined by wild swings, cronyism and wide income disparities, with ethnic and class divisions that have mostly kept the poor poor and the rich rich. The current era of drug-fueled violence has added another challenge. Indeed, for all its left coast breeziness, Campus Tec is still a place that must be protected with shotgun-toting guards and biometric keypads that require approved thumbprints for entry. Even Mr. Mini admits it is not exactly what venture capitalists from Palo Alto, Calif., like to see. And yet, as an urban microprocessor — a catalyst for change — Campus Tec is showing signs of promise. After about a year of operation, neighbors say it is already stirring a small revival in a district called 4 Grados that has had its ups and downs. A San Francisco-style apartment building a few blocks away, with glass-fronted lofts (built by Mr. Mini's cousin), recently filled up, selling out faster than many expected. Certainly the young people in skinny jeans and too-tight plaid button-downs walking around at lunchtime on a recent afternoon were a welcome sight for business owners like Patricia Laparra, 43. The owner of a designer glasses boutique near Campus Tec's main entrance, Ms. Laparra was one of many surprised by Campus Tec's impact. October was her second best month for sales since opening seven years ago. After a time when she thought she would have to close, she now says she has no plans to leave. "They're geeks," she

Keyphrases:

Silicon Valley Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	Full Sail University Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	New York Times Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○
Campus Tecnológico Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	city center Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	local Silicon Valley Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○
Tec Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	Guatemala City Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	Victor Blue Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○
Developer hope Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	Guatemala Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○	Global Entrepreneurship Monitor Excellent ○ Good ○ Neutral ● Poor ○ Very poor ○
time Excellent ○ Good ○ Neutral ●	Campus Excellent ○ Good ○ Neutral ●	average cost Excellent ○ Good ○ Neutral ●

Figure 4.8: The interface of the application used by the volunteers for evaluating the accuracy of the keyphrases extracted by DIKpEW approach

Extracted Keyphrases	i	rel_i
Social Tagging	1	$Excellent$
usefull	2	$VeryPoor$
Web 2.0 system	3	$Good$

Table 4.6: An example of the manually evaluated ranked results

people involved in the evaluation had two weeks for providing their feedback. As result, the volunteers evaluated the keyphrases generated for 209 Web pages written in Italian and in English. Since each user was allowed to freely select the input (the Web pages) for the evaluation mechanism we could not compute statistical measures for assessing the reliability of agreement among the participants. We choose to not focus the evaluation on a specific set of Web pages in order to make easier the evaluation task. In fact, people could chose to evaluate the keyphrase extracted from the Web pages the visited (which are also probably interesting for them) reducing in this way the efforts of the volunteers.

On the other hand, used the Normalized Discounted Cumulative Gain (NDCG) metric to evaluate the results of our evaluation. The NDCG metric is commonly used in the area of Information Retrieval in order to measure the accuracy of results of ranking mechanisms. This measure is specifically used in scenarios where the ranked results are associated to different relevance levels. In fact, it takes into account the position of the results and the usefulness (or gain) of the result to compute the accuracy of the evaluated mechanism: the final gain is computed from the top to the bottom of the result list with the gain of each result discounted at lower ranks. This measure is based on the assumption that more relevant documents must appear earlier in the result list. More formally, given a ranked list of resources where the resource (in our case the keyphrase) in position i is associated to a relevance score rel_i (in our case the position is defined by our algorithm and the relevance by the evaluators) we can compute the gain for this list as follows

$$DCG = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{log_2 i}$$

where n is the number of results in the ranked list and in our specific case n is equal to 10. In our evaluation the relevance scores associated to the possible evaluations were: Excellent = 4; Good = 3; Neutral = 2; Poor = 1; Very poor = 0. To simplify the description of the DCG metric the reader can consider the following simple example where we compute the DCG for the the manually evaluated ranked list of keyphrases reported in Table 4.6

the DCG for this evaluation is computed is equal to

$$4 + 0 * log_2 2 + 3 * log_2 3 = 5.887$$

By computing the DCG over each evaluation provided by the users, we obtained

	NDCG@5	**NDCG@10**
Base_Ita	0.484	0.437
DIKpEW_Ita	0.558	0.614
Base_Eng	0.485	0.576
DIKpEW_Eng	0.523	0.686

Table 4.7: Performance of DIKpEW compared to the baseline mechanism

an assessment of the accuracy for each evaluated Web page. In order to compute the NDCG for an approach we normalized the DCG values in $[0, 1]$ and then we computed the mean of these normalized values.

We computed 8 distinct NDCG values for evaluating and comparing the accuracy of:

- the top 5 and top 10 keyphrases extracted by DIKpEW from Web pages written in Italian (DIKpEW_Ita);
- the top 5 and top 10 keyphrases extracted by the baseline system from Web pages written in Italian (Base_Ita);
- the top 5 and top 10 keyphrases extracted by DIKpEW from Web pages written in English(DIKpEW_Eng);
- the top 5 and top 10 keyphrases extracted by the baseline system from Web pages written in English(Base_Eng);

Table 4.7 summarizes the computed results.

According to the results showed in the table the DIKpEW outperforms the baseline mechanism in the case of both Web pages written in Italian and Web pages written in Italian. Moreover, the accuracy of the results computed for the Web pages in Italian are comparable to the accuracy for the Web pages in English. This means that the noise introduced during the translation phase does not significantly lowers the accuracy of the results. This is due to two facts:

1. the DIKpEW approach uses statistical information about the keyphrase (such as the position of the keyphrase) for computing the keyphraseness without executing more complex natural language techniques.
2. the wikipedia feature allows us to throw out the bi-grams and tri-grams which have not a clear meaning.

4.7 Tags vs Keyphrases

In order to evaluate if the proposed approaches can effectively be used for modeling the user interests and for filtering the resources in a more accurate way, we integrated the DIKpEP in a content-based recommender system. The recommender system uses DIKpEP approach for extracting the keyphrases from the scientific papers in order to have: a description of the user interest, i.e. the user profile and the descriptions of the available papers, i.e the resource profile. The user profile is generated by combining the profiles of the resources which are interesting for the user.

Since the aim of this section is to understand if keyphrases can provide a better description of both resources and user interests we will compare two versions of the recommender systems:

1. the version which extracts only uni-grams for generating the description of resources and user interests

2. the version which uses also bi-grams and tri-grams for modeling both resources and interests.

We are going to describe the process of computing the recommendations by using uni-grams, bi-grams and tri-grams in Section 4.7.1. This description allows the reader to understand also the characteristics of the baseline mechanism used to evaluate the results presented and discussed in Section 4.7.2.

4.7.1 Using Keyphrases to Compute recommendations

Given a user, the papers that she tagged are considered relevant papers for building the User Profile. The profile is constituted by three (ordered) lists of weighted and stemmed keyphrases: the list of uni-grams (the uni-gram profile), the list of bi-grams (the bi-gram profile), and the list of tri-grams (the tri-gram profile).

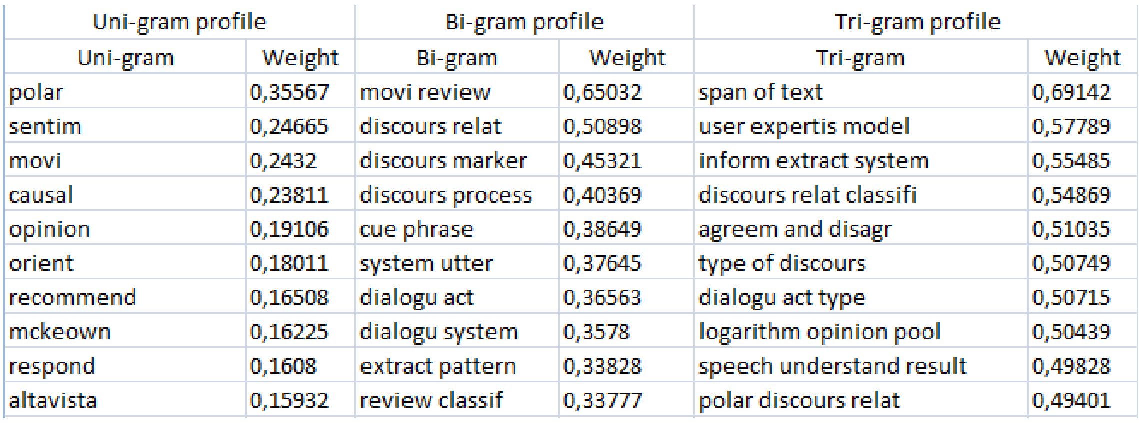
More specifically, given the list of relevant papers $\{p_1, \ldots, p_n\}$ for the active user, we exploit the DIKpEP approach for extracting uni-grams, bi-grams, and tri-grams from each relevant paper. This step produces three lists of weighted keyphrases:

1. the list of the weighted uni-grams $\{UniGrams(p_1), \ldots, UniGrams(p_n)\}$;

2. the list of the weighted bi-grams $\{BiGrams(p_1), \ldots, BiGrams(p_n)\}$;

3. the list of the weighted tri-grams $\{TriGrams(p_1), \ldots, TriGrams(p_n)\}$.

All uni-grams lists $\{UniGrams(p_1), \ldots, UniGrams(p_n)\}$ are then merged to build the uni-gram profile and, similarly, the lists of bi-grams and tri-grams are merged to build the bi-gram and the tri-gram profiles. More specifically, given the lists of uni-grams extracted from the relevant papers, each distinct uni-gram is stemmed and then inserted in the final list of relevant keyphrases. The weight assigned to each uni-gram in the uni-gram profile is computed by summing the weights associated to it by the keyphrase extraction technique. The weight of each keyphrase is then multiplied by the idf value associated to the specific keyphrase. The same technique is applied to produce the bi-gram profile and the tri-gram profile. Figure 4.9 shows the most relevant uni-grams, bi-grams and tri-grams extracted from the user profile of one of the users (using a set of 10 relevant documents and removing the uni-grams with a keyphraseness lower the 0.9, the bi-grams with a keyphraseness lower than 0.8 and the tri-grams with a keyphraseness lower than 0.8) in the dataset described in the next section.



Uni-gram profile		Bi-gram profile		Tri-gram profile	
Uni-gram	Weight	Bi-gram	Weight	Tri-gram	Weight
polar	0,35567	movi review	0,65032	span of text	0,69142
sentim	0,24665	discours relat	0,50898	user expertis model	0,57789
movi	0,2432	discours marker	0,45321	inform extract system	0,55485
causal	0,23811	discours process	0,40369	discours relat classifi	0,54869
opinion	0,19106	cue phrase	0,38649	agreem and disagr	0,51035
orient	0,18011	system utter	0,37645	type of discours	0,50749
recommend	0,16508	dialogu act	0,36563	dialogu act type	0,50715
mckeown	0,16225	dialogu system	0,3578	logarithm opinion pool	0,50439
respond	0,1608	extract pattern	0,33828	speech understand result	0,49828
altavista	0,15932	review classif	0,33777	polar discours relat	0,49401

Figure 4.9: An example of a user profile: the most relevant n-grams

In order to compute the relevance of a new paper p_k for a given user profile, the approach follows a similar path: it extracts the three lists of keyphrases from the paper and, then, these keyphrases are stemmed.

The final step is performed by the Matching Module (see Figure 1), which takes in input three lists, $UniGrams(p_k)$, $BiGrams(p_k)$ and $TriGrams(p_k)$, and the user profile. The matching process is based on the cosine similarity producing three similarity values, one for each category of n-grams. Then, an appropriate combination (linear in the first experiments) of these three similarity values is used to compute a unique score to be assigned to the considered paper p_k. Finally, they highest score papers are recommended to the active user.

4.7.2 Evaluating the content-based approach

The proposed approach extracts keyphrases from scientific paper in order to have a richer description of user interests which, in turn, is used to improve the quality of a content-based recommender system. The main assumption is that keyphrases have meaningful contextual information (not accounted in the classical bag-of-word model) which can be used to improve a cognitive filtering mechanism. In order to validate our claim we performed some experimental evaluation by using a publicly available dataset which contains 597 full papers extracted from the ACL Anthology Reference Corpus (ACL ARC)[7]: this dataset has been built from a significant subset

[7] http://acl-arc.comp.nus.edu.sg/

Table 4.8: The precision of the proposed approach and using only uni-grams

	Uni-grams Based	The Proposed Approach
p@1	0.83	0.93
p@3	0.61	0.77
p@5	0.65	0.80
p@7	0.64	0.73

of the ACL Anthology, a scientific digital library of papers on natural language processing and computational linguistics, composed by 10921 papers published since February 2007. The dataset includes specific data about 28 researchers (15 junior researchers and 13 senior researchers), interested in natural language processing. In particular, each researcher reported his relevant papers.

In our evaluation we used this feedback to both build the user profiles of researchers and evaluate the precision of the computed recommendations. In particular, given a researcher we divided the set of his preferences into two set of papers. We used one of these sets (composed by 20 papers) as training set in order to build the user profiles while the second set has been used as test set for comparing the results provided by the recommendation engine. According to this setting we computed the recommendations for the researchers in the dataset. Moreover, in order to evaluate the improvement with respect to a baseline bag-of-word approach, we also built the user profiles and document representations using only uni-grams. By using this setting (which does not include bi-grams and tri-grams) we computed a new set of recommendations.

Table 4.8 compares the precision of the recommendations obtained by using only unigrams to the precision obtained by using the approach described in 4.7.1 where the precision is computed as the ratio among the number of correct recommendations and the number of produced recommendations.

The table shows that by tanking into account also bi-grams and tri-grams we obtain an higher precision. This confirms that keyphrases can provide useful information for describing both the user interests and the resources.

4.8 Conclusion

Social tagging system are one of the main example of crowdsourcing systems: they outsource the classification task by using the collective intelligence of the Web 2.0 users for assigning metadata to resources. However, by shifting the classification task from a set of experts to a larger and not trained set of people, the results of the classification cannot be rigorous due to the lack of any control or guidelines. In order to improve the results of the user generated tasks we can better support the users by extracting meaningful terms from the textual resource.

In this chapter we focused specifically on the task of extracting meaningful n-grams (i.e multi-terms which can be suggested or used to classify the resources) from scientific paper and Web pages. The evaluations show that the proposed approaches can effectively be used to support the users of social tagging systems in classifying the resources they upload.

Future work will focus on the integration of the knowledge available in systems such as DBPedia [5] or other RDF sources. In fact, the proposed approaches can return keyphrases which are in the scientific paper and Web pages. However, we believe that by integrating other structured resources we can obtain meaningful classifications of resources.

Conclusions

The social semantic relations emerging from the tagging activities executed by the user of social tagging systems offer new opportunities to both model users and to adaptively filter the resources. This issue has been investigated in this thesis where we have proposed some methods for using and enriching the social semantic relations in order to improve the access to the knowledge stored in social tagging systems. In fact, the social semantic relations can be integrated in a recommender system in order to better model the user interests and to improve the accuracy of the resource filtering.

More specifically in Chapter 2 we proposed two CF approaches which integrate the social semantic relations for modeling the ToIs of the users of social tagging systems. These two approaches differ mainly in the mechanism used to infer the social semantic relations. The first of the two approaches described, the ACFF mechanism, follows the idea that the social semantic relations involving tags and resources can be inferred by taking into account the agreement of the users about the meaning of tags. From a technical point of view, this means that the co-occurrences among tags on the same resource is a good parameter to infer the similarity among the tags. The identified social semantic relations are used in our mechanism in order to classify the feedback of the users: tags and resources of the users are associate to the ToIs. This classification is then used for identifying a specific neighborhood for a given ToI. As a result, given a ToI of a user, the mechanism identifies a specific set of people which share the ToI with the user and the feedback of these people is combined to produce the recommendations. This mechanism extend the classical idea of CF recommender system which use the opinions of people with similar interests for filtering the more useful resources. In our work we focus on the idea that if users share only a subset of their interests we have to consider only the shared ToIs for generating the recommendations. This idea is then exploited also in the second CF approach proposed in this work, the ACFP mechanism. The ACFP mechanism is based on the idea that each user can build social semantic relations which are not shared by a large community of users. In fact, this approach analyzes the personomy of the user in order to identify meaningful relations among tags and resources.

On the other hand, Chapter 3 focuses on the task of enriching the semantic value of the relations built by the users. In order to reach this aim we proposes two mechanisms for supporting the user during the tagging process. The proposed approaches want to extract meaningful keyphrases respectively from scientific papers (the DIKpEP approach) and Web pages (the DIKpEW approach) in order to avoid typing errors and the generation of tags with a poor semantic value. Our mechanisms are domain independent and the keyphrases are identified by evaluating a set of features which takes into account the structure of both scientific paper and Web pages.

We showed that the discussed approaches are plausible since they outperforms other state of the art mechanism. We also integrated the proposed mechanisms in the larger PIRATES framework.

Future works will focus on the integration on the mechanisms proposed in Chapter 2 and Chapter 3. Moreover, in Chapter 3 we started to integrate the information stored in Wikipedia for extracting keyphrases from Web pages and at the moment the Wikipedia is used for identifying coherent keyphrases. Future works will analyze the contents of this knowledge source for identifying keyphrases which are not reported in the corpus of the specific resource but are meaningful to classify the document.

A

Appendix A - Published Papers

- P. Casoto, A. Dattolo, **F. Ferrara**, N. Pudota, P. Omero, and C. Tasso. Generating and sharing personal information spaces. In *Proceedings of the Workshop on Adaptation for the Social Web, 5th ACM International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 29 July - 1 August 2008, pages 14-23.

- A. Dattolo, **F. Ferrara** and Carlo Tasso. Supporting Personalized User Concept Spaces and Recommendations for a Publication Sharing System. Geert-Jan Houben, Gord I. McCalla, Fabio Pianesi, Massimo Zancanaro (Eds.): *User Modeling, Adaptation, and Personalization, 17th International Conference, UMAP 2009, formerly UM and AH*, Trento, Italy, June 22-26, 2009. Lecture Notes in Computer Science (5535) Springer 2009, pp. 325-330

- A. Dattolo, **F. Ferrara**, C. Tasso. Modeling a publication sharing system 2.0. In *Proceedings of the 2nd International Conference on Human System Interaction - HSI09*, Catania, Italy, May 21-23, 2009, IEEE, pp. 495-501.

- A. Dattolo, **F. Ferrara** and Carlo Tasso. Neighbor Selection and Recommendations in Social Bookmarking Tools. In *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications - ISDA 2009*, November 30 - December 2, 2009, Pisa, Italy, pp. 267-272

- A. Dattolo, **F. Ferrara** and C. Tasso. The role of tags for recommendation: a survey. In *Proceedings of the 3rd International Conference on Human System Interaction - HSI'2010*, Rzeszow, Poland, May 13-15, 2010, pp. 548-555. IEEE press.

- A. Dattolo, **F. Ferrara** and C. Tasso. On social semantic relations for recommending tags and resources in folksonomies. Mroczek, et alt. (Eds.): *Computer Systems Interaction: Background and Applications 2, Part1- Advances in Intelligent and Soft Computing (Volume 98)*, pp. 311-326. Springer-Verlag Berlin Heidelberg, 2011.

- **F. Ferrara**, I. Torre, L. Sarti, C. Tasso, A. Dattolo, S. Bocconi and J. Earp. Resources and users in the tagging process: approaches and case studies. *Journal of e-Learning and Knowledge Society, Vol 6, No 2*, pp. 37 - 51 (May 2010).

- N. Pudota, A. Dattolo, A. Baruzzo, **F. Ferrara** and C. Tasso. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems, Special Issue: New Trends for Ontology-Based Knowledge Discovery*, 25(12):1158-1186 (December 2010).

- **F. Ferrara**, N. Pudota and C. Tasso. A Keyphrase-Based Paper Recommender System. M. Agosti, et alt. (Eds.): *IRCDL, Communications in Computer and Information Science (CCIS) 249*, pp. 14-25, 2011. Springer-Verlag Berlin Heidelberg 2011.

- **F. Ferrara** and C. Tasso. A Personalized Intelligent Recommender and Annotator TEStbed for text-based content retrieval and classification: the PIRATES Project. M. Agosti, et alt. (Eds.): *IRCDL, Communications in Computer and Information Science (CCIS) 249*, pp. 136-139, 2011. Springer-Verlag Berlin Heidelberg 2011.

- G. Leitner, **F. Ferrara**, A. Felfernig and C. Tasso. Decision Support in the Smart Home. In *Proceedings of the Workshop on Human Decision Making in Recommender Systems, at RecSys 2011* October 23-27, 2011, Chicago, IL, USA

- **F. Ferrara** and C. Tasso. Extracting and Exploiting Topics of Interests from Social Tagging Systems. In *Proceedings of the International Conference on Adaptive and Intelligent Systems - ICAIS'11* September 06-08, 2011, Klagenfurt, Austria, pp. 285-296

- **F. Ferrara** and C. Tasso. Improving Collaborative Filtering in Social Tagging Systems. In *Proceedings of the Workshop on Knowledge Extraction and Exploitation from semi-Structured Online Sources, at the Conference of the Spanish Association for Artificial Intelligence - CAEPIA 2011*, 8 November 2011, pp. 463-472

Bibliography

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

[2] Citeseerx. Website. http://citeseer.ist.psu.edu/.

[3] Citeulike: Portal. Website. http://www.citeulike.org/.

[4] Connotea: free online reference management for clinicians and scientists. Website. http://www.connotea.org/.

[5] Dbpedia. Website. http://dbpedia.org/About.

[6] Delicious.com. Website. http://www.delicious.com/.

[7] Esp game - from wikipedia, the free encyclopedia. Website. http://en.wikipedia.org/wiki/ESP_game.

[8] Experian hitwise - facebook reaches top ranking in us. Website. http://weblogs.hitwise.com/heather-dougherty/2010/03/facebook-reaches-top-ranking-i.html.

[9] Facebook. Website. http://www.facebook.com/.

[10] Folksonomy coinage and definition. Website. http://vanderwal.net/folksonomy.html.

[11] gwap.com - home. Website. http://www.gwap.com/gwap/.

[12] Mark zuckerberg - person of the year 2010 - time. Website. http://www.time.com/time/specials/packages/article/0,28804,2036683_2037183_2037185,00.html.

[13] Movielens data sets — grouplens research. Website. http://www.grouplens.org/node/73.

[14] Netflix prize. Website. http://www.netflixprize.com/.

[15] New youtube statistics: 48 hours of video uploaded per minute; 3 billion views per day - search engine watch. Website. http://searchenginewatch.com/article/2073962/New-YouTube-Statistics-48-Hours-of-Video-Uploaded-Per-Minute-3-Billion-Views-Per-Day.

[16] The rise of crowdsourcing. Website. http://www.wired.com/wired/archive/14.06/crowds.html.

[17] The six billionth pic on flickr - flickr boasts 6 billion photo uploads - softpedia. Website. http://news.softpedia.com/newsImage/Flickr-Boasts-6-Billion-Photo-Uploads-2.jpg/.

[18] Social commerce statistics - bazaarvoic. Website. http://www.bazaarvoice.com/resources/stats.

[19] Welcome to flickr - photo sharing. Website. http://www.flickr.com/.

[20] Welcome to groupme! - the social semantic web. Website. http://groupme.org/GroupMe/.

[21] Wikipedia - from wikipedia, the free encyclopedia. Website. http://en.wikipedia.org/wiki/Wikipedia.

[22] Wikipedia, the free encyclopedia. Website. http://en.wikipedia.org/.

[23] Wordnet - about wordnet. Website. http://wordnet.princeton.edu/.

[24] The world of relevant information in the palm of your hand. Website. http://www.extractor.com/.

[25] Youtube - broadcast yourself. Website. http://www.youtube.com/.

[26] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transaction on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[27] E. Agichtein and L. Gravano. Snowball: extracting relations from large plain-text collections. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94, New York, NY, USA, 2000. ACM.

[28] M. Alessandro, M. Paul, and Sanda M. H. Open domain information extraction via automatic semantic labeling. In Ingrid Russell and Susan M. Haller, editors, *FLAIRS Conference*, pages 397–401. AAAI Press, 2003.

[29] D. E. Appelt and D. J. Israel. Introduction to information extraction technology. A tutorial prepared for IJCAI-99, Stockholm, Schweden, 1999.

[30] E. D. Avanzo, B. Magnini, and A. Vallin. Keyphrase extraction for summarization purposes: the LAKE System at DUC2004. In *Proceedings of Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL 2004)*, Boston, USA, 2004.

[31] L. P. Aye and T. Nilar. Domain adaptive information extraction using link grammar and wordnet.

[32] R. Baeza-Yates, A. Gionis, F. P. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri. Design trade-offs for search engine caching. *ACM Transactions on the Web*, 2(4):1–28, 2008.

[33] L. Baltrunas and F. Ricci. Locally adaptive neighborhood selection for collaborative filtering recommendations. In *Proc. of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 22–31, Hannover, Germany, July 2008.

[34] K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pages 40–52, London, UK, 2000. Springer-Verlag.

[35] P. Basile, M. de Gemmis, A. L. Gentile, P. Lops, and G. Semeraro. Uniba: Jigsaw algorithm for word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 398–401, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[36] P. Basile, D. Gendarmi, F. Lanubile, and G. Semeraro. Recommending smart tags in a social bookmarking system. In *Proc. of the Workshop on Bridging the Gap between Semantic Web and Web 2.0 the 4th European Semantic Web Conference*, pages 22–29, Innsbruck, Austria, June 2007.

[37] N. Belkin and W. Croft. Information filtering and information retrieval: two sides of the same coin. *Communications of the ACM*, 35(12):29–38, 1992.

[38] Dominik Benz, Andreas Hotho, Robert Jaschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, and Gerd Stumme. The social bookmark and publication management system bibsonomy. *The VLDB Journal*, 19(6):849–875, 2010.

[39] BibSonomy. Bibsonomy dataset, dumps for research purposes. `http://www.kde.cs.uni-kassel.de/bibsonomy/dumps`.

[40] D. Billsus and M. Pazzani. User modeling for adaptive news access. *UMUAI*, 2(10):147–180, 2000.

[41] Y Blanco-Fernandez, J Pazos-arias, A Gil-Solla, M Ramos-Cabrer, and M Lopez-Nores. Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Transactions on Consumer Electronics*, 54(2):727–735, 2008.

[42] D. B. Bracewell, F. Ren, and S. Kuroiwa. Multilingual single document keyword extraction for information retrieval. In *Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pages 517–522, Wuhan, 2005.

[43] W. Y. F. Brook and L. Quanzhi. Document keyphrases as subject metadata: incorporating document key concepts in search results. *Information Retrieval*, 11(3):229–249, 2008.

[44] K. Bruce and B. Chad. Learning user information interests through the extraction of semantically significant phrases. In *AAAI 1996 Spring Symposium on Machine Learning in Information Access*, 1996.

[45] R. Burke. Interactive critiquing for catalog navigation in e-commerce. *Artificial Intelligence Review*, 18:245–267, December 2002.

[46] R. Burke. The adaptive web. chapter Hybrid web recommender systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.

[47] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pages 328–334, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[48] I. Cantador, A. Bellogín, I. Fernández-Tobías, and S. López-Hernández. Semantic contextualisation of social tag-based profiles and item recommendations e-commerce and web technologies. volume 85 of *Lecture Notes in Business Information Processing*, chapter 9, pages 101–113. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[49] Ivn Cantador, Alejandro Bellogn, and Pablo Castells. News@hand: A semantic web approach to recommending news. volume 5149 of *Lecture Notes in Computer Science*, pages 279–283. Springer, 2008.

[50] P. Casoto, A. Dattolo, and C. Tasso. Sentiment classification for the italian language: A case study on movie reviews. *Journal of Internet Technology*, 9(4):365–373, 2008.

[51] Kannan Chandrasekaran, Susan Gauch, Praveen Lakkaraju, and Hiep Phuc Luong. Concept-based document recommendations for citeseer authors. In *Proceedings of the 5th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH '08, pages 83–92, Berlin, Heidelberg, 2008. Springer-Verlag.

[52] M. Clements. Personalization of social media. In *Proc. of the BCS IRSG Symposium: Future Directions in Information Access*, pages 93–98, Glasgow, Scotland, August 2007.

[53] A. Csomai and R. Mihalcea. Linguistically Motivated Features for Enhanced Back-of-the-Book Indexing. In *Proceedings of ACL-08: HLT*, pages 932–940, Columbus, Ohio, 2008. Association for Computational Linguistics.

[54] M. A. Damaris, B. Sean, F. Heidi, G. Herbert, I. Robert, and M. W. Ralph. BBN: description of the PLUM system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 169–176, Virginia, USA, 1992. Morgan Kaufmann.

[55] M. De Gemmis, P. Lops, G. Semeraro, and P. Basile. Integrating tags in a semantic content-based recommender. In *Proc. of the 2nd ACM Int. Conf. on Recommender Systems*, pages 163–170, Lausanne, Switzerland, October 2008.

[56] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *Transaction on Information Systems*, 1(22):143–177, 2004.

[57] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.

[58] R. Ellen. Information extraction as a stepping stone toward story understanding. *Understanding language understanding: computational models of reading*, pages 435–460, 1999.

[59] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[60] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. In *Proceedings of the 19th national conference on Artifical intelligence*, pages 391–398. AAAI Press / The MIT Press, 2004.

[61] D. Fallows. The internet and daily life. Website. `http://www.pewtrusts.org/uploadedFiles/wwwpewtrustsorg/Reports/SocietyandtheInternet/pewinternet081204.pdf`.

[62] J. P. Feather and R. P. Sturges. *International encyclopedia of information and library science.* Routledge, London, UK, 1997.

[63] F. Ferrara, N. Pudota, and C. Tasso. A keyphrase-based paper recommender system. In *Proc. of the 7th Italian Research Conference on Digital Libraries*, Pisa, Italy, January 2011.

[64] Felice Ferrara and Carlo Tasso. A personalized intelligent recommender and annotator testbed for text-based content retrieval and classification: The pirates project. In Maristella Agosti, Floriana Esposito, Carlo Meghini, and Nicola Orio, editors, *Digital Libraries and Archives*, volume 249 of *Communications in Computer and Information Science*, pages 136–139. Springer Berlin Heidelberg, 2011.

[65] D. Freitag. Toward general-purpose learning for information extraction. In *Proceedings of the 17th international conference on Computational linguistics*, pages 404–408, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[66] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1301–1306. AAAI Press, 2006.

[67] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, pages 71–80, New York, NY, USA, 2007. ACM.

[68] J. Gemmell, T. Schimoler, M. Ramezani, and B. Mobasher. Adapting k-nearest neighbor for tag recommendation in folksonomies. In *Proceedings of the 7th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems in in conjunction with the 21st International Joint Conference on Artificial Intelligence*, July 2009.

[69] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005.

[70] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[71] S. Golder and A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[72] G. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14:403–420, 1970.

[73] R. Grishman and B. Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[74] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1-2):81–104, 1999.

[75] J. Han, T. Kim, and J. Choi. Web Document Clustering by Using Automatic Keyphrase Extraction. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, pages 56–59, Washington, DC, USA, 2007. IEEE Computer Society.

[76] U. Hanani, B. Shapira, and Shoval P. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11:203–259, 2001.

[77] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transaction on Information Systems*, 22:5–53, 2004.

[78] J. R. Hobbs, D. Appelt, M. Tyson, J. Bear, and D. Israel. SRI International: description of the FASTUS system used for MUC-4. In *Proceedings of the 4th conference on Message understanding*, pages 268–275, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[79] J. Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business.* Crown Publishing Group, New York, NY, USA, 1 edition, 2008.

[80] C. Huang, Y. Tian, Z. Zhou, C. X. Ling, and T. Huang. Keyphrase Extraction Using Semantic Networks Structure Analysis. In *Proceedings of the Sixth International Conference on Data Mining*, pages 275–284, Washington, DC, USA, 2006. IEEE Computer Society.

[81] S. B. Huffman. Learning information extraction patterns from examples. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260, London, UK, 1996. Springer-Verlag.

[82] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 329–338, New York, NY, USA, 1993. ACM.

[83] A. Hulth. Improved given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[84] A. Hulth. *Automatic Keyword Extraction: Combining Machine Learning and Natural Language Processing.* VDM Verlag, Saarbrücken, Germany, 2008.

[85] A. Hulth and B. B. Megyesi. A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 537–544, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[86] H. W. Ian, P. Gordon, Eibe F., G. Carl, and G. N. Craig. KEA: Practical Automatic Keyphrase Extraction. In *Proceedings of Digital Libraries 99 (DL'99)*, pages 254–255, New York, NY, USA, 1999. ACM Press.

[87] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems An Introduction.* Cambridge University Press, 2010.

[88] R. Jäschke, , A. Hotho, L. Schmidt-Thieme, and G. Stumme. Folkrank: A ranking algorithm for folksonomies. In *Proceedings of FGIR*, 2006.

[89] J. Justeson and S. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, pages 9–27, 1995.

[90] H. Karanikas, C. Tjortjis, and B. Theodoulidis. An approach to Text mining using Information Extraction. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), Workshop on Knowledge Management Theory and Applications(KMTA)*, Lyon, France, 2000.

[91] K. Katharina and M. Silvia. Information Extraction. A Survey. Technical Report, Asgaard-TR, 2005.

[92] M. H. Khaled, N. M. Diego, and S. K. Mohamed. Corephrase: Keyphrase extraction for document clustering. In Petra Perner and Atsushi Imiya, editors, *MLDM*, volume 3587 of *Lecture Notes in Computer Science*, pages 265–274. Springer, 2005.

[93] J. T. Kim and D. I. Moldovan. Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):713–724, 1995.

[94] Yehuda Koren and Robert M. Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer US, 2011.

[95] B. Kosovac, D. J. Vanier, and T. M. Froese. Use Of Keyphrase Extraction Software For Creation Of An AEC/FM Thesaurus. *Electronic Journal of Information Technology in Construction*, 5:25–36, 2000.

[96] G. Krupka, P. Jacobs, L. Rau, L. Childs, and I. Sider. GE NLToolset: description of the system as used for MUC-4. In *Proceedings of the 4th conference on Message understanding*, pages 177–185, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[97] N. Kumar and S. Kannan. Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *Proceeding of the eighth ACM symposium on Document engineering*, pages 199–208, New York, NY, USA, 2008. ACM.

[98] F. Lancaster. *Information retrieval systems: characteristics, testing, and evaluation.* Wiley, New York, 1979.

[99] A. Lancichinetti, S. Fortunato, and J. Kertsz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(033015), 2009.

[100] L. Lathauwer, B. Moor, and J. Vandewalle. A multilinear singular value decomposition. *Journal of Matrix Analysis and Applications*, 4(21):1253–1278, 2000.

[101] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[102] Z. Liu, L. Peng, Y. Zheng, and M. Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266, Singapore, 2009. Association for Computational Linguistics.

[103] Pasquale Lops, Marco Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.

[104] T. Malone, K. Grant, F. Turbak, S. Brobst, and M. Cohen. Intelligent information sharing systems. *Communications of ACM*, 30(5):390–402, 1987.

[105] D Maltz and E. Ehrlich. Pointing the way: Active collaborative filtering. In *Proc. of the ACM SIGCHI'95 conference on Human factors in computing systems*, pages 202–209, Denver, Colorado, USA, May 1995.

[106] C. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[107] C. Marlow, M. Naaman, D. Boyd, and M. Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *Proc. of the 17th ACM Conference on Hypertext and Hypermedia*, pages 31–40, Odense, Denmark, August 2007.

[108] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. Website. `http://www.adammathes.com/academic/computermediatedcommunication/folksonomies.html`.

[109] A. McCallum. Information extraction: distilling structured data from unstructured text. *ACM Queue*, 3(9):48–57, 2005.

[110] Miller McPherson, Lynn S. Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[111] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. The adaptive web. chapter Personalized Search on the World Wide Web, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.

[112] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In *Proceedings of Empirical Methods in Natural Language Processing*, page 404411, Barcelona, Spain, 2004. Association for Computational Linguistics.

[113] M. Minio and C. Tasso. User modeling for information filtering on internet services: Exploiting an extended version of the umt shell. In *Workshop on User Modeling for Information Filtering on the WWW. In connection with the 5th UM International Conference*, pages 2–5, Kailua-Kona, Hawaii, USA, January 1996.

[114] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of 15th International Conference on World Wide Web*, pages 953–954. ACM, 2006.

[115] S. Mizzaro. How many relevances in information retrieval. *Interacting With Computers*, 10(3):305–322, 1998.

[116] C. Musto, F. Narducci, M. De Gemmis, P. Lops, and G. Semeraro. Star : a social tag recommender system. In *Proceedings of the ECML/PKDD 2009 Discovery Challenge Workshop*, 2009.

[117] T. Nguyen and M. Y. Kan. Keyphrase extraction in scientific publications. In Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Sølvberg, and Edie M. Rasmussen, editors, *ICADL*, volume 4822 of *Lecture Notes in Computer Science*, pages 317–326. Springer, 2007.

[118] The official Google blog. We knew web was big. Website. `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`.

[119] M. Olen. *Human-competitive automatic topic indexing*. PhD thesis, University of Waikato, Hamilton, New Zealand, July 2009.

[120] Derry OSullivan, Barry Smyth, and David C. Wilson. Preserving recommender accuracy and diversity in sparse datasets. *International Journal on Artificial Intelligence Tools*, 13(1):219–235, 2004.

[121] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

[122] Denis Parra and Peter Brusilovsky. Collaborative filtering for social tagging systems: an experiment with citeulike. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 237–240, New York, NY, USA, 2009. ACM.

[123] T. Peter. Learning to extract keyphrases from text. Technical Report ERB-1057, National Research Council, Institute for Information Technology, 1999.

[124] T. Peter. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336, 2000.

[125] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[126] P. Pu and L. Chen. Trust building with explanation interfaces. In *Proc. of the 11th the Int. Conf. on Intelligent User Interface*, pages 93–100, Sydney, Australia, January 2006.

[127] P. Pu, L. Chen, and H. Rong. Evaluating recommender systems from the users' perspective: Survey of the state of the art (to appear). *User Modeling and User-Adapted Interaction Journal*, 2012.

[128] N. Pudota, A. Dattolo, A. Baruzzo, F. Ferrara, and C. Tasso. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems, Special Issue on New Trends for Ontology-Based Knowledge Discovery*, 25(12):1158–1186, 2010.

[129] E. Qualman. Socialnomics: How social media transforms the way we live and do business. 2009.

[130] J. Quinlan. Learning efficient classification procedures and their application to chess end games. In *Machine Learning. An Artificial Intelligence Approach*, pages 463–482, 1983.

[131] E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816. AAAI Press/The MIT Press, 1993.

[132] E. Riloff. Automatically generating extraction patterns from untagged text. *Proceedings of the Thirteenth Annual Conference on Artificial Intelligence*, pages 1044–1049, 1996.

[133] Y. Roman and G. Ralph. NYU: Description of the Proteus/PET system as used for MUC-7 ST. In *Proceedings of the 7th Message Understanding Conference: MUC-7*, Fairfax, Virginia, 1998.

[134] S. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[135] J. Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, chapter 9, pages 291–324. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2007.

[136] S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In *Proc. of the 18th International Conference on World Wide Web*, pages 671–680, Madrid, Spain, April 2009.

[137] B. Sergey and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[138] Van Setten, R. M., Brussee, H. Van Vliet, L. Gazendam, Y. Van Houten, and M. Veenstra. On the importance of who tagged what. In *Proc. of the Workshop on the Social Navigation and Community based Adaptation Technologies at the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 552–561, Dublin, Ireland, June 2006.

[139] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in collaborative tagging systems using hierarchical clustering. In *Proc. of the 2nd ACM Int. Conf. on Recommender Systems*, pages 259–266, Lausanne, Switzerland, October 2008.

[140] C. Shirky. Ontology is overrated: Categories, links, and tags. Website. `http://www.shirky.com/writings/ontology_overrated.html`.

[141] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[142] S. Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1-3):233–272, 1999.

[143] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–1319, San Francisco, CA, USA, 1995. Morgan Kaufmann.

[144] S. Sood, S Owsley, K. Hammond, and Birnbaum. L. Tagassist: Automatic tag suggestion for blog posts. In *Proceedings of the International Conference on Weblogs and Social Media*, 2007.

[145] G. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 4(35):551–566, 1992.

[146] K. Sudo. *Unsupervised discovery of extraction patterns for information extraction*. PhD thesis, New York, NY, USA, 2004. Adviser-Sekine, Satoshi and Adviser-Grishman, Ralph.

[147] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *Proc. of the ACM SIGIR 2001 Workshop on Recommender Systems*, New Orleans, Lousiana, 2001.

[148] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 43–50. ACM, 2008.

[149] C. Tasso and F. A. Asnicar. ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In *Adaptive Systems and User Modeling on the WWW, 6th UM International Conference*, 1997.

[150] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence*, pages 855–860. AAAI Press, 2008.

[151] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Proc of the Collaborative Web Tagging Workshop at the 15th World Wide Web Conference*, Edinburgh, Scotland, May 2006.

[152] H. K. Yaakov, G. Zuriel, and M. Asaf. Automatic extraction and learning of keyphrases from scientific articles. In Alexander F. Gelbukh, editor, *CICLing*, volume 3406 of *Lecture Notes in Computer Science*, pages 657–669. Springer, 2005.

[153] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational linguistics*, pages 940–946, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[154] F. B. W. Yi, L. Quanzhi, S. B. Razvan, and C. Xin. Finding nuggets in documents: A machine learning approach. *Journal of the American Society for Information Science and Technology*, 57(6):740–752, 2006.

[155] V. Zanardi and L. Capra. Social ranking: Finding relevant content in web 2.0. In *Proc. of the 2nd ACM Int. Conf. on Recommender Systems*, pages 51–58, Lausanne, Switzerland, October 2008.

[156] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15:574–595, 2010.

[157] C. Zhang. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.

[158] Y. Zhen, W. Li, and D. Yeung. Tagicofi: tag informed collaborative filtering. In *Proc. of the 3th ACM International Conference on Recommender systems*, pages 69–76, New York, NY, USA, October 2009.

[159] T. Zhou, H. Ma, M. Lyu, and I. King. Userrec: A user recommendation framework in social tagging systems. In *Proc. of the 24th AAAI Conference on Artificial Intelligence*, pages 1486–1491, Atlanta, Geogia, USA, July 2010.

[160] A. Zollers. Emerging motivations for tagging: Expression, performance, and activism. In *Workshop on Tagging and Metadata for Social Information Organization in connection with the WWW2007 Conference*, May 2007.