



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

**Personalized Recommender Systems for Resource-based Learning**  
Hybrid Graph-based Recommender Systems for Folksonomies

Dem Fachbereich  
Elektrotechnik und Informationstechnik  
der Technischen Universität Darmstadt  
zur Erlangung des akademischen Grades eines  
Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte

**Dissertationsschrift**

von

**Dipl.-Inform. Mojisola Helen Erdt geb. Anjorin**  
Geboren am 16. September 1980 in Zaria, Nigeria

Erstreferent: Prof. Dr.-Ing. Ralf Steinmetz  
Korreferent: Prof. Dr.-Ing. habil. Ulrike Lucke

Tag der Einreichung: 12. September 2014  
Tag der Disputation: 19. November 2014

Darmstadt, 2014  
Hochschulkennziffer D17

---



# Personalized Recommender Systems for Resource-based Learning

## Hybrid Graph-based Recommender Systems for Folksonomies

Zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte Dissertation von Dipl.-Inform. Mojisola Helen Erdt geb. Anjorin, geboren am 16. September 1980 in Zaria, Nigeria

2014 – Darmstadt – D17



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Fachbereich Elektrotechnik  
und Informationstechnik

Fachgebiet Multimedia Kommunikation  
Prof. Dr.-Ing. Ralf Steinmetz

Personalized Recommender Systems for Resource-based Learning  
Hybrid Graph-based Recommender Systems for Folksonomies  
genehmigte Dissertation von Dipl.-Inform. Mojisola Helen Erdt geb. Anjorin, geboren am 16. September 1980 in Zaria, Nigeria

Tag der Einreichung: 12. September 2014

Tag der Disputation: 19. November 2014

Erstreferent: Prof. Dr.-Ing. Ralf Steinmetz

Korreferent: Prof. Dr.-Ing. habil. Ulrike Lucke

Technische Universität Darmstadt  
Fachbereich Elektrotechnik und Informationstechnik

Fachgebiet Multimedia Kommunikation (KOM)  
Prof. Dr.-Ing. Ralf Steinmetz

---

***Dedicated to all who hunger for knowledge.***

*Especially to the girls abducted in April 2014 in Nigeria  
who are still facing unimaginable terrors  
for simply wanting to go to school.*

*May the day come when all children  
everywhere in the world  
can learn together in peace.*



---

## Abstract

---

As the Web increasingly pervades our everyday lives, we are faced with an overload of information. We often learn on-the-job without a teacher and without didactically prepared learning resources. We not only learn on our own but also collaboratively on social platforms where we discuss issues, exchange information and share knowledge with others. We actively learn with resources we find on the Web such as videos, blogs, forums or wikis. This form of self-regulated learning is called *resource-based learning*. An ongoing challenge in technology enhanced learning (TEL) and in particular in resource-based learning, is supporting learners in finding learning resources relevant to their current needs and learning goals. In social tagging systems, users collaboratively attach keywords called *tags* to resources thereby forming a network-like structure called a *folksonomy*. Additional semantic information gained for example from activity hierarchies or semantic tags, form an *extended folksonomy* and provide valuable information about the context of the resources the learner has tagged, the related activities the resources could be relevant for, and the learning task the learner is currently working on. This additional semantic information could be exploited by recommender systems to generate personalized recommendations of learning resources.

Thus, the first research goal of this thesis is to develop and evaluate personalized recommender algorithms for a resource-based learning scenario. To this end, the resource-based learning application scenario is analysed, taking an existing learning platform as a concrete example, in order to determine which additional semantic information could be exploited for the recommendation of learning resources. Several new hybrid graph-based recommender approaches are implemented and evaluated. Additional semantic information gained from activities, activity hierarchies, semantic tag types, the semantic relatedness between tags and the context-specific information found in a folksonomy are thereby exploited. The proposed recommender algorithms are evaluated in offline experiments on different datasets representing diverse evaluation scenarios. The evaluation results show that incorporating additional semantic information is advantageous for providing relevant recommendations.

The second goal of this thesis is to investigate alternative evaluation approaches for recommender algorithms for resource-based learning. Offline experiments are fast to conduct and easy to repeat, however they face the so called *incompleteness problem* as datasets are limited to the historical interactions of the users. Thus newly recommended resources, in which the user had not shown an interest in the past, cannot be evaluated. The recommendation of novel and diverse learning resources is however a requirement for TEL and needs to be evaluated. User studies complement offline experiments as the users themselves judge the relevance or novelty of the recommendations. But user studies are expensive to conduct and it is often difficult to recruit a large number of participants. Therefore a gap exists between the fast, easy to repeat offline experiments and the more expensive user studies. Crowdsourcing is an alternative as it offers the advantages of offline experiments, whilst still retaining the advantages of a user-centric evaluation. In this thesis, a crowdsourcing evaluation approach for recommender algorithms for TEL is proposed and a repeated evaluation of one of the proposed recommender algorithms is conducted as a proof-of-concept. The results of both runs of the experiment show that crowdsourcing can be used as an alternative approach to evaluate graph-based recommender algorithms for TEL.





---

## Zusammenfassung

---

Das World Wide Web spielt in unserem Alltag eine immer bedeutsamere Rolle. Wir sind daher mit einer zunehmenden Informationslast konfrontiert. Oft lernen wir am Arbeitsplatz ohne Lehrer und ohne didaktisch aufbereitete Lernressourcen. Wir lernen nicht nur alleine, sondern auch gemeinsam mit anderen auf sozialen Plattformen, wo wir Themen diskutieren, Informationen austauschen und Wissen miteinander teilen. Wir lernen aktiv mit Ressourcen, wie Videos, Blogs, Foren oder Wikis, die wir im Web finden. Diese Form des selbstgesteuerten Lernens wird *ressourcenbasiertes Lernen* genannt. Eine Herausforderung im ressourcenbasierten Lernen sowie auch allgemein im technologiegestützten Lernen (TEL) ist die Unterstützung des Lernenden bei der Suche nach denjenigen Lernressourcen, die für seine aktuellen Bedürfnisse und Lernziele relevant sind. In Tagging-Systemen fügen Benutzer selbst Stichworte, sogenannte *Tags*, zu Ressourcen hinzu, wodurch eine netzwerkartige Struktur namens *Folksonomie* entsteht. Zusätzliche semantische Informationen — beispielsweise von Aktivitäts-Hierarchien oder semantischen Tags — bilden eine *erweiterte Folksonomie* und liefern wertvolle Informationen über den Kontext der Ressourcen der Lernenden, über die aktuelle Lernaufgabe der Lernenden sowie über die dazugehörigen Aktivitäten, die für die Ressourcen relevant sein können. Diese zusätzlichen semantischen Informationen könnten von Empfehlungssystemen genutzt werden, um personalisierte Empfehlungen von Lernressourcen zu erzeugen.

Somit ist das erste Forschungsziel dieser Arbeit, personalisierte Empfehlungsalgorithmen für ein ressourcenbasiertes Lernszenario zu entwickeln und zu evaluieren. Zu diesem Zweck wird im Rahmen dieser Arbeit das ressourcenbasierte Lernen unter Verwendung einer bestehenden Lernplattform analysiert, um zu bestimmen, welche zusätzlichen semantischen Informationen für Empfehlungen von Lernressourcen genutzt werden können. Hierzu werden mehrere neue hybride graphbasierte Empfehlungsalgorithmen, die zusätzliche semantische Informationen ausnutzen, entwickelt und evaluiert. Als zusätzliche Informationen werden hierbei Aktivitäten, Aktivitäts-Hierarchien, semantische Tag-Typen, die semantische Verwandtschaft zwischen Tags sowie kontextspezifische Informationen der Folksonomie verwendet. Die entwickelten Empfehlungsalgorithmen werden anhand mehrerer Evaluierungsszenarien und verschiedener Datensätze evaluiert. Die Evaluationsergebnisse zeigen, dass das Einbeziehen zusätzlicher semantischer Informationen bei der Bereitstellung relevanter Empfehlungen von Vorteil ist.

Das zweite Ziel dieser Arbeit ist die Entwicklung neuer Evaluierungsansätze für Empfehlungsalgorithmen für ressourcenbasiertes Lernen. Offline-Experimente auf historischen Datensätzen haben den Vorteil, dass sie schnell durchführbar und einfach wiederholbar sind. Allerdings leiden diese Experimente generell unter dem sogenannten *Unvollständigkeitsproblem*, da die Datensätze nur die historischen Interaktionen der Benutzer abbilden und somit neue Empfehlungen nicht ausgewertet werden können. Eine solche Auswertung ist allerdings beim TEL notwendig, da das Empfehlen von für den Lernenden neuartigen und vielfältigen Lernressourcen vorteilhaft ist. Offline-Experimente werden oft durch Benutzerstudien ergänzt, bei denen die Benutzer selbst die Relevanz oder Neuheit der Empfehlungen bewerten. Allerdings sind Benutzerstudien teuer und es ist oft schwierig, eine große Anzahl an Teilnehmern zu gewinnen. Crowdsourcing stellt eine Alternative zu den schnell durchführbaren und einfach zu wiederholenden Offline-Experimenten und den teureren Benutzerstudien dar, da Crowdsourcing sowohl die Vorteile eines Offline-Experiments als auch die einer benutzerorientierten Evaluation vereint. In dieser Arbeit wird ein Crowdsourcing-Konzept für die Evaluation von TEL Empfehlungsalgorithmen präsentiert und auf einen der in dieser Arbeit entwickelten Algorithmen exemplarisch angewandt. Die Ergebnisse zeigen, dass das vorgeschlagene Crowdsourcing-Konzept für die Evaluation graphbasierter TEL Empfehlungsalgorithmen verwendet werden kann.



---

## Acknowledgements

---

I am very grateful to my family, friends and colleagues who supported and encouraged me during my thesis.

Many thanks go to Prof. Ralf Steinmetz for giving me the chance to spend the last four years with very supportive colleagues at the Multimedia Communications Lab (KOM). My heartfelt thanks go to Dr. Christoph Rensing for his supervision and constructive advice over the years. Many thanks also go to Prof. Ulrike Lucke for her support and encouragement during the last year of my thesis. I am also grateful to Dr. Alejandro Fernández for his feedback and constant encouragement.

My thanks go to all my colleagues at KOM, especially the Knowledge Media Group, for their companionship and support. A huge thanks goes in particular to those whom I had the good luck to share an office room with over the years: Thanks to Phil for always taking the time to listen and give advice. Thanks to Renato for all his support and for sharing PHD comics and jokes with us ;-) Thanks to Sebastian for all the helpful discussions and feedback. Thanks to Irina for her happy-go-lucky moods and supply of chocolate :-)

Many thanks also go to all my students who inspired me with their input. In particular to Thomas for the many, many discussions and exchange of ideas, to Florian for his optimism and insightful suggestions, and to Katja for her enthusiasm and refreshingly different view on how to do things.

My thanks go to my family and friends for their love and encouragement. I thank my parents for giving me a solid foundation to grow on and always being there for me when I needed them. I am grateful to my loving sister for her steadfast faith in me. I thank my favourite brother for sharing this journey with me - it was a blessing having you to share the joy and the pain with. I thank my sister-in-law for her down-to-earth advice and encouragement. . . and for sticking it out with us through the last four years ;-)

I thank my loving husband for his support and never-ending patience.

And I thank my two little nephews for reminding me of what is truly important in life.



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Research Goals and Research Approach . . . . .	2
1.3	Main Contributions . . . . .	3
1.4	Outline . . . . .	4
<b>2</b>	<b>Personalized Recommendation of Learning Resources for Technology Enhanced Learning</b>	<b>7</b>
2.1	Overview of TEL Research Areas . . . . .	7
2.2	Resource-based Learning . . . . .	10
2.3	Social Tagging Systems . . . . .	11
2.3.1	Folksonomy . . . . .	12
2.3.2	Additional Semantic Information - Extended Folksonomy . . . . .	13
2.4	Application Scenario: CROKODIL . . . . .	13
2.4.1	Semantic Tag Types . . . . .	14
2.4.2	Pedagogical Concept of Activity Hierarchies . . . . .	14
2.4.3	CROKODIL's Extended Folksonomy . . . . .	15
2.5	Summary . . . . .	16
<b>3</b>	<b>Fundamentals and Related Work</b>	<b>19</b>
3.1	Recommender Systems . . . . .	19
3.1.1	Classification of Recommender Systems . . . . .	20
3.1.2	Recommender Systems for Social Tagging Systems . . . . .	22
3.1.3	Recommender Systems for Technology Enhanced Learning . . . . .	27
3.2	Evaluating Recommender Systems . . . . .	30
3.2.1	Classification of Evaluation Methodologies for Recommender Systems . . . . .	30
3.2.2	A Survey of Evaluation Methods for TEL Recommender Systems . . . . .	34
3.3	Summary and Research Goals . . . . .	42
<b>4</b>	<b>Hybrid Graph-based Recommender Approaches for TEL</b>	<b>49</b>
4.1	Hybrid Approaches Exploiting Activities and Activity Hierarchies . . . . .	49
4.1.1	AScore . . . . .	49
4.1.2	AINheritScore . . . . .	51
4.2	Hybrid Approach Exploiting Semantic Tagging (AspectScore) . . . . .	53
4.3	Hybrid Approach Exploiting Semantic Relatedness between Tags (InteliScore) . . . . .	55
4.4	Hybrid Approach Exploiting Context-Specific Information in Folksonomies (VSScore) . . . . .	57
4.5	Providing Personalized Recommendations to Support Resource-based Learning . . . . .	58
4.5.1	Hybrid Recommender Systems for Extended Folksonomies . . . . .	60
4.5.2	Implementation and Integration in CROKODIL . . . . .	60
4.6	Summary . . . . .	63
<b>5</b>	<b>Evaluation of the Hybrid Graph-based Recommender Approaches</b>	<b>65</b>
5.1	Evaluation Methods and Evaluation Metrics . . . . .	65
5.1.1	State-of-the-Art Evaluation Methods and Metrics . . . . .	65
5.1.2	Evaluation Method: LeaveRTOUT . . . . .	67
5.1.3	Evaluation Metric: Mean Normalized Precision (MNP) . . . . .	69

5.1.4	Evaluation Approach and Evaluation Datasets . . . . .	70
5.2	Evaluation Results . . . . .	73
5.2.1	Results of AspectScore, InteliScore and VSScore on the BibSonomy Dataset . . . . .	73
5.2.2	Results of AScore and AInheritScore on the GroupMe! Dataset . . . . .	79
5.2.3	Results of AspectScore, AScore and AInheritScore on the CROKODIL Dataset . . . . .	83
5.3	Summary . . . . .	88
<b>6</b>	<b>An Alternative Evaluation Approach for TEL Recommender Systems</b>	<b>91</b>
6.1	Crowdsourcing as an Evaluation Approach in Research . . . . .	91
6.2	Crowdsourcing Concept for Evaluating Recommender Systems for TEL . . . . .	92
6.2.1	The Preparation Phase . . . . .	93
6.2.2	The Execution Phase . . . . .	97
6.3	Crowdsourcing Evaluation Results . . . . .	99
6.3.1	Analysis of Results Comparing Algorithms AScore and FolkRank . . . . .	100
6.3.2	Analysis of Results Comparing Recommendations to Sub-Activities . . . . .	105
6.4	Summary . . . . .	112
<b>7</b>	<b>Conclusion and Outlook</b>	<b>113</b>
7.1	Main Contributions . . . . .	113
7.2	Outlook . . . . .	114
	<b>Bibliography</b>	<b>117</b>
	<b>List of Acronyms</b>	<b>137</b>
<b>A</b>	<b>Details of the Survey on TEL Recommender Algorithms</b>	<b>141</b>
<b>B</b>	<b>Details of the Evaluation of the Hybrid Algorithms</b>	<b>147</b>
B.1	Fundamentals of Evaluation Metrics and Statistical Significance Testing . . . . .	147
B.1.1	Accuracy Prediction Metrics . . . . .	147
B.1.2	Statistical Significance Tests . . . . .	147
B.2	Evaluation Details . . . . .	150
B.2.1	Parametrization . . . . .	150
B.2.2	Detailed Evaluation Results on the BibSonomy Dataset . . . . .	152
B.2.3	Detailed Evaluation Results on the GroupMe! Dataset . . . . .	152
B.2.4	Detailed Evaluation Results on the CROKODIL Dataset . . . . .	152
<b>C</b>	<b>Details of Crowdsourcing Experiment and Statistical Analysis of Results</b>	<b>161</b>
C.1	Crowdsourcing Experiment - Questionnaire . . . . .	161
C.1.1	Experiment Spring: Recommendations made to the Activities . . . . .	161
C.1.2	Experiment Autumn: Recommendations made to the Activities . . . . .	165
C.2	Demographics of Participants in Experiment Spring and Experiment Autumn . . . . .	166
C.3	Descriptive Statistics for Experiment Spring and Experiment Autumn . . . . .	171
C.3.1	Descriptive Statistics for AScore and FolkRank . . . . .	171
C.3.2	Descriptive Statistics for Sub-Activities . . . . .	174
C.3.3	Descriptive Statistics for Crowdworkers and Volunteers . . . . .	174
C.4	Inference Statistics - Cohen's d Effect Size . . . . .	179
<b>D</b>	<b>Details of the PageRank Algorithm and the Random Surfer Model</b>	<b>181</b>
D.1	The PageRank Algorithm . . . . .	181

---

D.2	The Random Surfer Model - Markov Chains . . . . .	185
D.3	The Intelligent Surfer Model - Personalized PageRank . . . . .	187
D.4	The Biased Surfer Model - Topic-Sensitive PageRank . . . . .	188
<b>E</b>	<b>Author's Publications</b>	<b>191</b>
E.1	Main Publications . . . . .	191
E.2	Co-Authored Publications . . . . .	192
<b>F</b>	<b>Supervised Student Theses</b>	<b>195</b>
F.1	Master Theses . . . . .	195
F.2	Bachelor Theses . . . . .	195
<b>G</b>	<b>Curriculum Vitae</b>	<b>197</b>
<b>H</b>	<b>Erklärung laut §9 der Promotionsordnung</b>	<b>199</b>





---

## 1 Introduction

---

### 1.1 Motivation

---

Today, more and more learners turn to the Web as their first source of information to solve certain tasks or for learning purposes. This form of self-regulated learning with resources found on the Web is known as *Resource Based Learning (RBL)* [94, 191]. Resources could be as diverse as videos found on YouTube<sup>1</sup>, answers to questions discussed in forums, blog posts, or articles found in Wikipedia<sup>2</sup>. Due to the ever increasing amount of resources found on the Web, learners are continuously confronted with the challenge of finding and selecting relevant learning resources that fit their needs. Thus there is an increasing need in resource-based learning to support learners in identifying and organising learning resources that are relevant to their current task or learning goal. Recommender systems can be used to provide personalized support to learners in a social tagging system by suggesting learning resources relevant to a particular task or learning goal [159]. Hence, learners no longer have to search for learning resources but rather have relevant resources, tailored to their current needs and learning goals, offered to them as recommendations.

In Technology Enhanced Learning (TEL), and in particular resource-based learning, *Social Tagging Systems* [126] have often been used to support learners in collaboratively organizing and sharing resources they have found on the Web with other learners. In social tagging systems, users attach keywords, called tags, to resources such as web pages or blogs. Through the process of collaborative tagging, a network-like structure called a *folksonomy* emerges [186]. A folksonomy comprises the users, resources, tags and the user's assignment of tags to resources. In folksonomies, recommender systems can exploit the graph structure gained from the tagging behaviour of the learners to make recommendations that go beyond just considering the similarity between resources.

Furthermore, social tagging systems that support resource-based learning often have additional features to support the learners in their learning process, such as additional semantic information attached to tags, learning goals, and learning activities [10, 15, 25, 177, 220]. The additional semantic information thus gained can be included in the folksonomy and further exploited to improve recommendations. A folksonomy enriched with additional semantic information is termed an *extended folksonomy* [1]. The folksonomy graph could be extended by including additional nodes or links or by adjusting the weights of the nodes. Related work shows that recommender approaches can provide improved recommendations by considering additional semantic information gained from extended folksonomies [97, 107, 188, 244], however not all semantic information available in a resource-based learning scenario has yet been fully investigated and exploited. Recommendations for learning purposes have different requirements compared to recommendations in other domains such as e-commerce [159]. Recommendations for learning need to be personally tailored to the learner, considering the learner's current learning goal or activity. In addition to being relevant to the learner and to the learner's current learning activity, it is a requirement for recommendations for TEL to be novel and diverse to the learner. Thus there is a need for personalized recommender approaches that provide novel and diverse recommendations relevant to the learner's current activity. This thesis aims to contribute to filling this gap.

As more and more TEL recommender algorithms are developed, it becomes increasingly important to measure their benefits in supporting learners. Offline experiments using historical datasets offer a fast and repeatable way to evaluate TEL recommender systems. However, it is a challenge to find datasets that fit the specified evaluation requirements and contain all specific information needed by the recommender algorithm. Offline experiments also face the challenge of the incompleteness problem [49] - in order for

---

<sup>1</sup> <http://www.youtube.com>, retrieved 01.06.2014

<sup>2</sup> <http://www.wikipedia.org>, retrieved 01.06.2014

---

a resource to be evaluated as relevant, the users must have interacted with or indicated an interest in the recommended resource in the past. As such, a newly recommended resource can neither be judged as relevant nor irrelevant by an offline experiment and it remains unknown whether the user would actually find this recommendation interesting and relevant or not. To solve this problem, user experiments can be conducted to complement offline experiments. In a user experiment, the users themselves are asked to judge whether they find recommendations made to a particular task to be relevant or not. They can also indicate whether they find the recommendations new and diverse. But due to the effort and costs of performing user experiments, few are conducted and even less are repeated in order to evaluate different variations and settings of a recommender algorithm. Furthermore, a survey on the evaluation of recommender systems for learning shows that on average only about 40 participants take part in a single user study. Compared to offline evaluations, this further limits the evaluation scope of user studies. Hence, there exists a gap between the fast, easy to repeat offline experiments and the more expensive user studies.

---

## 1.2 Research Goals and Research Approach

---

The first research goal of this thesis is to develop and evaluate personalized recommender algorithms that support resource-based learning. As mentioned above, there exists a need to investigate whether and how additional semantic information gained from an extended folksonomy, such as the learner's current learning activity, the learner's hierarchical activity structures, as well as the related learning resources, their tags and semantic tag types, could be exploited to provide personalized recommendations of learning resources. To this aim, graph-based recommender approaches have been investigated as a basis for creating feature-enhanced hybrid graph-based recommender algorithms [27] considering additional semantic information gained from extended folksonomies.

Unlike research on recommender systems in e-commerce or other information retrieval research where the speed, accuracy of prediction and the large amount of information available in the system are the main research challenges, this thesis rather focuses on a scenario with smaller online learning communities [232] having specific goals and consisting of only a few users and a few hundred resources. Furthermore, the recommendation requirements in a learning scenario go beyond simply recommending similar resources [45, 159]. Consequently, the evaluation results cannot be compared to the performance results of large scale recommendation scenarios nor online search engines, but rather with comparable state-of-the-art recommender algorithms used as baselines. Hence, regarding the first goal, the focus of this thesis is on exploiting domain and system specific semantic information to provide relevant, novel and diverse recommendations. Although the recommender algorithms presented in this thesis have been designed and evaluated specifically for smaller online learning communities, they have the potential to be applied to improve recommendations in larger communities as well.

As a first step, the resource-based learning application scenario is analysed, taking the CROKODIL learning environment as a concrete example. From the analysis, the additional semantic information that could be exploited in order to enrich the folksonomy are identified and an extended folksonomy model for CROKODIL is defined. Activities, activity hierarchies and semantic tag types are identified as additional semantic information that could be exploited by recommender algorithms for resource-based learning. Recommender algorithms could be content-based [222] or knowledge-based [67], however these approaches do not consider the community data found in online learning communities such as the user's collaborative tagging behaviour or interactions with learning resources or other learners. Graph-based approaches have the advantage of exploiting the transitive relations between nodes in the graph, thereby taking the learning resources, semantic tags and hierarchical activity structures of other learners in the community into consideration. Additionally, it is possible to focus on a particular user or current learning activity thereby generating personalized recommendations as well as topic-sensitive recommendations respectively [41, 103]. Hence, hybrid graph-based recommender algorithms are identified as being suitable for a resource-based learning application scenario.

---

Several concepts for hybrid graph-based recommender algorithms exploiting additional semantic information found in a resource-based learning scenario are proposed, implemented and evaluated. As a proof-of-concept, the proposed recommender algorithms are integrated into the CROKODIL learning environment. Furthermore, the recommender algorithms are evaluated on several historical datasets representing different evaluation scenarios including the resource-based learning application scenario CROKODIL. However, due to the incompleteness problem, neither the novelty nor diversity of the recommendations made to the learner could be evaluated.

The second goal of this thesis is to investigate alternative evaluation approaches for recommender algorithms for TEL. To achieve this goal, first a detailed survey of TEL evaluation methodologies is conducted in order to identify the type of evaluation methods used, the focus of the evaluations, and the effects measured by evaluations of TEL recommender systems over the years. Next an evaluation approach for hybrid graph-based recommender algorithms for extended folksonomies is presented, along with an evaluation method and evaluation metric for offline experiments on historical datasets for folksonomies. These are then applied in the evaluation of the aforementioned hybrid graph-based recommender algorithms.

As identified above, there exists a gap between the fast, easy to repeat offline experiments and the more expensive user experiments, and from the results of the survey, user-centric evaluation metrics such as novelty and diversity are rarely evaluated in TEL. Crowdsourcing embraces the advantages of offline experiments by giving access to sufficient willing users, as well as being easy and fast to conduct and repeat. But it still retains the advantages of a user study of being able to evaluate user-centric metrics. Hence, a crowdsourcing evaluation concept is proposed to evaluate the relevance, novelty and diversity of learning resources recommended to a specified learning activity by graph-based recommender algorithms for TEL. As a proof of concept, the aforementioned offline experiments are further complemented with a repeated crowdsourcing experiment in order to measure the relevance, novelty and diversity of recommendations made by AScore - one of the proposed hybrid graph-based recommender algorithms. The proposed evaluation approach using crowdsourcing does not have the ambition of replacing all other evaluation approaches. It is rather seen as a complement to the existing evaluation approaches. Ideally, the conception and implementation of recommender algorithms should be accompanied by their evaluation. The choice of evaluation methodology should fit the design stage of the recommender algorithm and go hand in hand with its development, helping to optimize its various aspects as it evolves over time.

---

### 1.3 Main Contributions

---

The main contributions relating to the two main goals of this thesis are summarized below.

#### 1. Personalized Recommender Approaches to Support Resource-based Learning

The first goal is to implement and evaluate new concepts for personalized recommender algorithms that support resource-based learning. The contributions pertaining to this goal are the following:

- A detailed analysis of the application scenario is conducted where additional semantic information that could be exploited to improve graph-based recommender approaches for resource-based learning are identified. An extended folksonomy model for a resource-based learning application scenario is defined, taking CROKODIL [19] as a concrete example.
- Several new concepts to provide personalized recommendations of learning resources in a resource-based learning scenario [17] are presented, implemented and evaluated:
  - Two new concepts for hybrid graph-based recommender approaches exploiting additional information gained from activities and activity hierarchies (AScore and AInheritScore [19]) are implemented and evaluated on the historical datasets GroupMe! and CROKODIL.
  - A new concept for a hybrid graph-based recommender approach exploiting additional semantic information gained from semantic tagging (AspectScore [203]) is implemented and evaluated on the historical datasets BibSonomy and CROKODIL.

- A new concept for a hybrid graph-based recommender approach exploiting the semantic relatedness between tags (InteliScore [204]) is implemented and evaluated on the historical dataset BibSonomy.
- A new concept for a hybrid graph-based recommender approach considering the learner's context in a folksonomy (VSScore [204]) is implemented and evaluated on the historical dataset BibSonomy.
- A new concept to integrate personalized recommender approaches into a resource-based learning application scenario is implemented, taking CROKODIL as a concrete example and as a proof-of-concept.

## 2. Alternative Evaluation Approaches for Recommender Algorithms for Resource-based Learning

The second goal is to investigate alternative evaluation approaches for evaluating recommender algorithms for resource-based learning. The contributions are the following:

- A detailed survey of TEL evaluation methodologies is conducted in order to identify the type of evaluation methods used, the focus of the evaluations, and the effects measured by evaluations of TEL recommender systems over the years.
- An evaluation concept for hybrid graph-based recommender algorithms for extended folksonomies is described.
- A new evaluation method (LeaveRTOOut [203]) and a new evaluation metric (Mean Normalized Precision [203]) for offline experiments on historical datasets for folksonomies are presented and applied to evaluate the proposed hybrid recommender algorithms as a proof-of-concept [19, 203, 204].
- A new concept of crowdsourcing for evaluating the relevance, novelty and diversity of TEL recommender algorithms [78, 79, 175] is introduced and a repeated crowdsourcing evaluation of one of the proposed hybrid graph-based recommender algorithm (AScore) is conducted as a proof-of-concept [78, 79].

---

### 1.4 Outline

---

This thesis is organized as depicted in Figure 1.1, highlighting the main contributions presented in Chapter 4, Chapter 5 and Chapter 6. In addition, a definition of an extended folksonomy model for the application scenario CROKODIL can be found in Chapter 2. Furthermore, a survey of TEL evaluation methodologies is discussed in Chapter 3.

In Chapter 2, the area of Technology Enhanced Learning (TEL) is briefly introduced as well as the resource-based learning application scenario that motivates the research goals of this thesis. Resource-based learning is explained based on the resource-based learning model, thereby highlighting the challenges faced during each step the learners perform. Next, social tagging systems and folksonomies are introduced along with the application scenario CROKODIL. The additional semantic information found in a resource-based learning scenario based on semantic tagging and the pedagogical concept of activities and activity hierarchies are identified and an extended folksonomy model is defined.

In Chapter 3, relevant fundamentals and related work pertaining to this thesis are presented. Recommender systems are introduced in detail, explaining the recommendation task as well as giving an overview of the different kinds of recommender systems. The focus is laid on graph-based recommender approaches and on recommender systems for social tagging systems and for TEL. Furthermore, existing evaluation methodologies for recommender systems are introduced focusing on the different kinds of evaluation approaches for TEL: offline experiments, user studies and real life testing. The fundamentals of information retrieval, especially aspects relevant to information retrieval on the Web and in social tagging systems, are explained, focusing on evaluation methodologies and metrics. Finally, from the analysis of a survey of evaluation methodologies for recommender systems for TEL, a gap is identified between

---

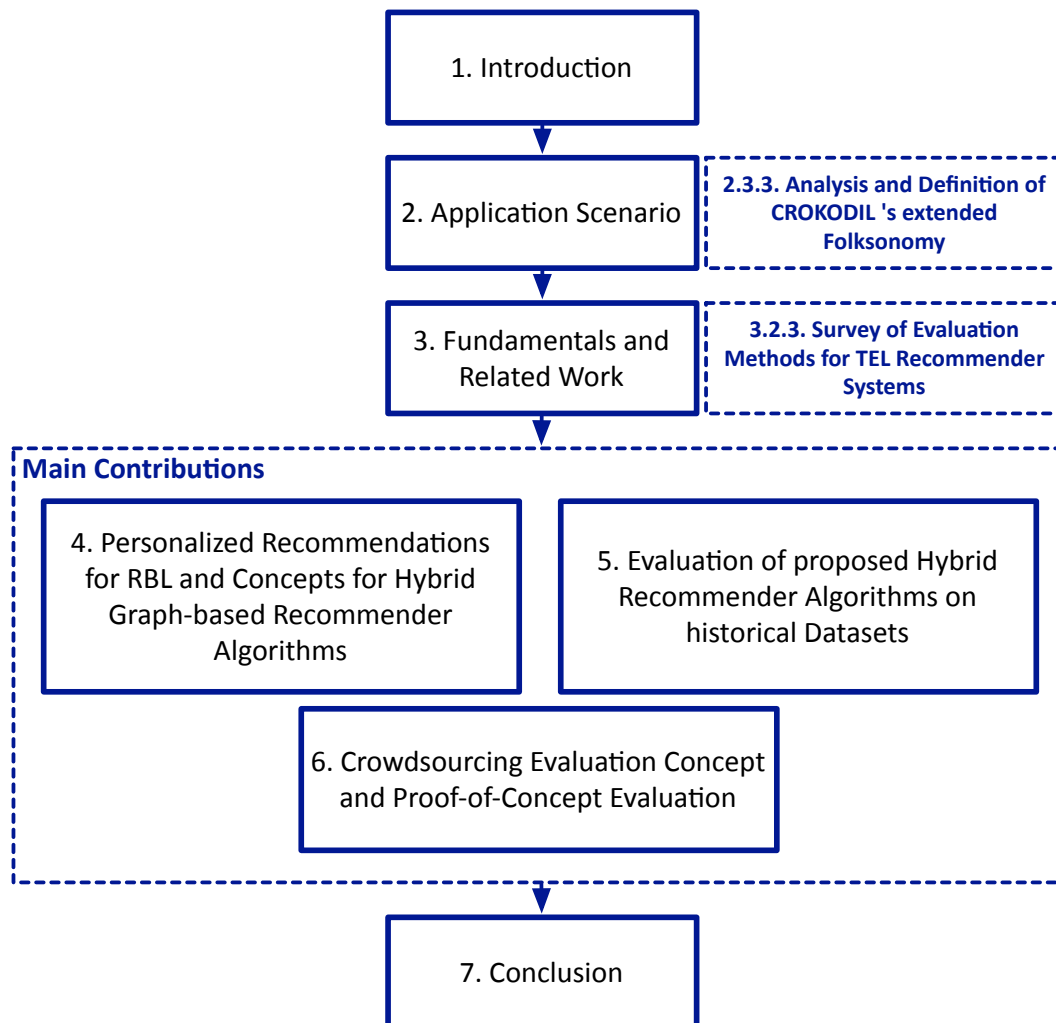
the fast and easy to conduct offline experiments and the more expensive user studies that enable the evaluation of user-centric metrics such as novelty and diversity.

In Chapter 4, several concepts for hybrid graph-based recommender approaches are proposed focusing on exploiting additional semantic information found in the extended folksonomy defined in Chapter 3. Two concepts for hybrid graph-based recommender approaches that exploit additional information gained from activities and activity hierarchies (AScore and AInheritScore) are presented. This is followed by a concept that incorporates additional semantic information gained from semantic tagging (AspectScore), and a concept utilizing the semantic relatedness between tags (InteliScore). Finally, a concept considering the learner's context in a folksonomy (VSScore) is described. As a proof-of-concept, personalized recommendations of learning resources are implemented and integrated into CROKODIL as an example of a concrete resource-based learning scenario.

In Chapter 5, the implemented hybrid graph-based recommender concepts introduced in Chapter 4 are evaluated in offline experiments. An evaluation approach for evaluating hybrid graph-based recommender algorithms for folksonomies in an offline experiment on historical datasets is described. Additionally, an evaluation method LeaveRTOOut and an evaluation metric Mean Normalized Precision are proposed to complement the existing state-of-the-art evaluation method and metric. The proposed hybrid approaches are evaluated on three historical datasets.

In Chapter 6, a crowdsourcing evaluation concept is introduced as an alternative evaluation approach to evaluate TEL recommender systems. A repeated proof-of-concept crowdsourcing evaluation applied to the AScore recommender algorithm proposed in Chapter 4 is performed to show that the proposed crowdsourcing evaluation concept can be used to effectively evaluate graph-based recommender algorithms for TEL.

Chapter 7 highlights once again the main contributions of this thesis giving an outlook on future work.



**Figure 1.1:** Outline



---

## 2 Personalized Recommendation of Learning Resources for Technology Enhanced Learning

---

This chapter presents the application scenario of this thesis. Firstly, a brief overview of TEL research areas is given, followed by a more detailed introduction to resource-based learning and social tagging systems. Thereafter the resource-based learning application scenario is presented and sources of additional semantic information to improve the recommendation of learning resources are analysed.

---

### 2.1 Overview of TEL Research Areas

---

**Technology Enhanced Learning (TEL)** aims to create technological approaches with an underpinning in social sciences to support and enhance learning [159]. TEL can also be referred to as E-learning and both terms will be used interchangeably in this thesis. Over the years, TEL has manifested itself in diverse forms, not only due to technological advancements like in new communication media, new data storage and access techniques, but also due to changes in social interactions in learning, thereby shifting from an individual form of learning to a more collaborative form in online communities.

In the early 1990s, TEL focused mainly on the creation of learning content for individual learning. As personal computers became faster, **Computer Based Trainings (CBTs)** [179] got more popular. Learning content was often distributed as courses e.g. to learn a foreign language on CD-ROMs. CBTs exploited the advantages gained from multimedia by integrating multimedia elements such as audio, video and animations into the learning content. Later on, with the advent of the Internet in personal homes, learning courses were offered online as **Web Based Trainings (WBTs)** [179]. This new possibility to access courses online led to advancements in the personalization of learning content [179]. User models built from the individual learner's level of knowledge, goals and preferences were used to adapt the learning content to the user. A new research area called **Adaptive Hypermedia** [43] thus became a prominent form of TEL, giving rise to the development of new adaptive hypermedia learning systems [60, 226]. The creation of adaptive learning content became an important research challenge where modularity played a major role [225]. However, creating high quality learning content was very time-consuming and expensive. Therefore it was important to support authors when reusing existing learning material or so called *Learning Objects (LOs)* or learning resources. These modular learning objects are commonly stored in a *Learning Object Repository (LOR)* [76, 100]. In order to promote the reuse of learning objects, a standard called *Learning Object Metadata (LOM)* [75] was developed that can be used to describe learning objects. Concepts were investigated to reuse learning objects in a modular way at distinct levels of granularity and consequently, the semi-automatic generation of metadata became an important focus in research [172]. Various authoring approaches and supporting authoring tools were developed. For example, authoring by aggregation [101], re-purposing with the support of a pattern based adaptation tool to support the authoring of adaptation processes [255] as well as the support for XML authoring [90, 170]. Many universities record lectures and offer the video recordings to students online as **E-lectures** [118]. E-lectures allow students and teachers more flexibility especially regarding conflicting schedules [118].

Learning however does not take place only on an individual level but rather as a collaborative process between several individuals or in a group [241]. Collaborative learning, though a very broad term, can simply be described as two or more persons learning or attempting to learn together [64]. The interaction between students is very important for learning as students learn more effectively when they cooperate with each other [116]. These pedagogical and psychological insights gave rise to the TEL research field known as **Computer Supported Cooperative Learning (CSCL)** [92]. CSCL investigates how to support collaborative learning amongst groups of learners. Several concepts and methods to support the learning process in a group in a virtual learning environment have been developed [249].

Collaborative learning however also takes place when interacting with Web 2.0 applications such as blogs, wikis and social networking services [77]. User-generated content plays a crucial role in this form

---

of TEL which is often termed **E-learning 2.0** [250]. As learners participate and contribute more actively online, so called **Communities of practice** [248] evolve over time. Such online communities often share a common goal of gaining more knowledge in a specific field by interacting and communicating with each other in a group. The members of the group benefit from each other by exchanging ideas and sharing information and experiences. Unlike in CSCL, where most forms of collaboration are usually instructional, even in extreme forms where interactions between learners are specified and controlled by so called social scripts [247], collaborative learning in E-learning 2.0 faces new challenges as learners have to acquire the skills and competences needed for self-regulated learning [34, 219]

Over the years, knowledge acquisition has become a lifelong process [225, 249]. Even at work, it is increasingly common to learn about job related topics by referring to information found on the Web. For example, developers constantly make use of dedicated technical forums such as Stack Overflow<sup>1</sup> to find out how to solve certain programming tasks. Often, there are no didactically prepared learning resources available to gain sufficient knowledge required to accomplish the task at hand. Therefore it is necessary to gain knowledge in a self-directed manner from learning content found on the Web. **Resource-based learning (RBL)** can be described as using resources mainly found on the Web for learning purposes [94, 191]. RBL is a special form of self-regulated learning using learning resources found on the Web, especially user-generated content from online learning communities. Learners have been shown to prefer learning through interactions with diverse learning resources [191]. These could be user-generated content such as videos found on YouTube<sup>2</sup> or presentation slides on Slideshare<sup>3</sup> as well as collaboratively constructed resources such as wikis and blogs [94]. Besides using and generating content, learners collaborate with other learners using diverse applications such as social network services to communicate with each other, discussion boards or forums where they ask questions and exchange information, social tagging systems where they share and tag web resources, as well as recommend relevant web pages about certain topics to other learners. The advantages of communication and interaction in communities and social networks can no longer be ignored as a very important part of the learning process [191]. Resource-based learning is the application scenario of this thesis and will be dealt with in more details below. A very challenging form of TEL is currently found in the application of mobile, ubiquitous, pervasive, contextualized and seamless technologies for learning, particularly in **Pervasive Learning** [150] where pedagogical strategies are adapted to the context of the learner or in **Mobile Learning** where the focus is on the mobility of learners and their interaction with mobile devices such as smart phones or tablets for learning purposes [62, 149]. Progress in this area is mainly due to the evolution of smart phones having sensors and access to the Internet. Challenges comprise the adaptation of technology to individuals, places and situations, for example offering support for location-based learning content authoring and content access [199] and for situated learning [144].

Another TEL research field called **Serious Games** or **Game based Learning** can be described as games with an educational purpose [168]. Research in this field is very interdisciplinary as game concepts and technologies are combined with multimedia technologies and pedagogical concepts [253, 254]. Game based learning can be said to be self-regulated but also adaptive and collaborative. Authoring tools for educational games offer support in dealing with the high complexity and interactivity of games. Concepts for the adaptive control of educational games and author support mechanisms have been developed specifically for adaptive single-player educational game authoring [169]. In addition, social media for peer education in single player educational games has also been investigated [134] and concepts for user-generated content exchange, game adaptation, and peer group formation have been developed [133]. The research challenge here lies in combining educational games with social media from an interdisciplinary perspective.

Recently, **Massive Open Online Courses (MOOCs)** have been gaining in importance [195] where learners register for open online courses and use open learning content from educational institutions

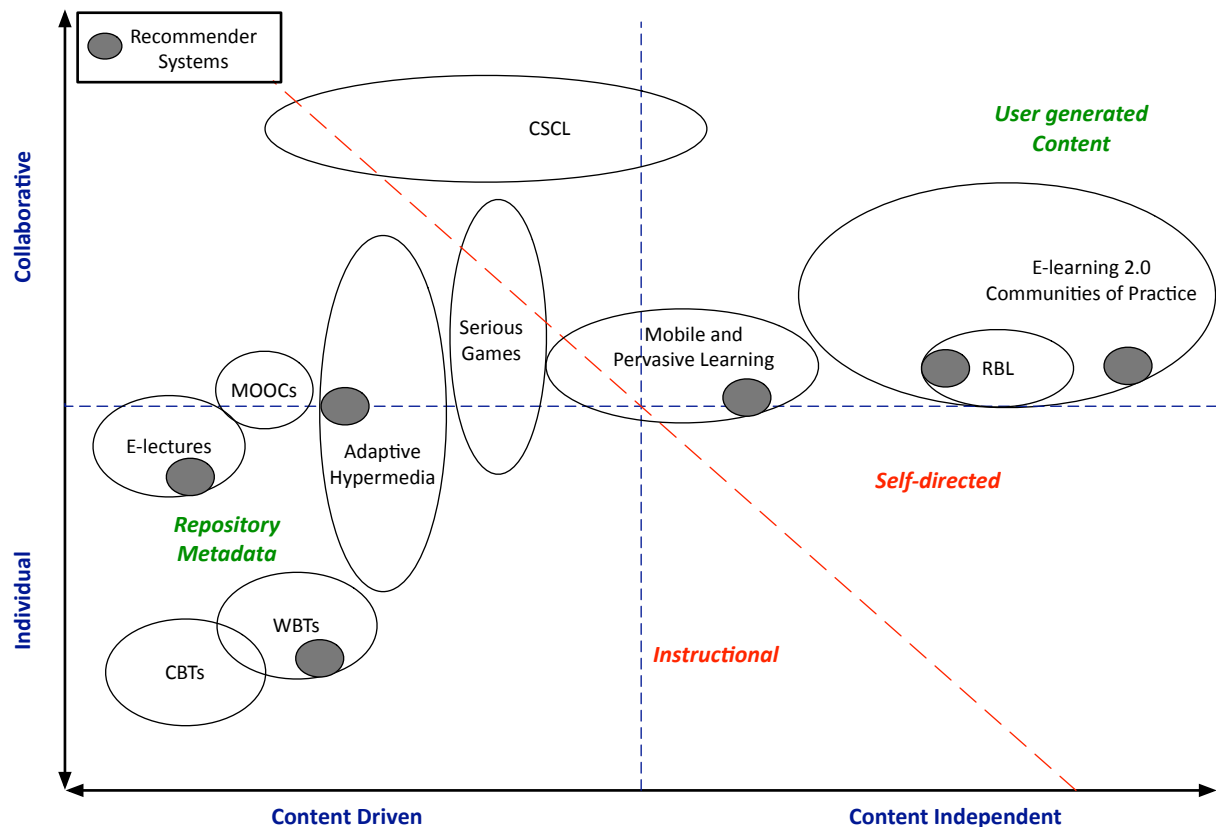
---

<sup>1</sup> <http://www.stackoverflow.com/>, retrieved 10.07.2014

<sup>2</sup> <http://www.youtube.com/>, retrieved 10.03.2014

<sup>3</sup> <http://www.slideshare.net/>, retrieved 10.03.2014





**Figure 2.1:** Overview of TEL Research Areas

like those offered by Coursera<sup>4</sup> or Udacity<sup>5</sup>. Whereas so called xMOOCs often offer prepared learning courses as videos, another type of MOOC, known as cMOOCs, demand more self-directed learning.

Figure 2.1 depicts the spectrum of these varied TEL research areas. It differentiates between individual forms of learning such as in CBTs and WBTs and other more collaborative forms of learning such as CSCL or learning in communities of practice. It also gives an overview of more content driven approaches like CBTs and WBTs and more content independent approaches such as RBL. Another aspect shown is the separation between more instructional learning like in E-lectures and a more self-directed learning approach as for example in E-learning 2.0. There is indeed a shift from instructional learning in formal institutional settings, to a more flexible self-directed learning that focuses more on problem solving and critical thinking [191].

As shown above, TEL is a broad field ranging from individual to collaborative learning, from content driven to content independent, from classroom to workplace and mobile learning, pervading almost all areas of our daily lives. As technology is used in diverse learning scenarios, so also over the last ten years, recommender systems have been created and deployed in the TEL domain to support various learning scenarios. Personalized recommender systems for TEL support learners in finding relevant learning resources, learning activities, peer learners, experts, tutors or even learning paths tailored to the learner's personal needs and learning goals [159]. Recommender systems play an important role not only in more self-directed learning scenarios such as in RBL but also in more content driven learning for example for recommending videos in LORs, WBTs or E-lectures. In collaborative, self-directed learning scenarios, especially where user-generated content prevails, recommender systems utilize the collective intelligence gained from the learning community, where every learner contributes to the overall quality of information in the system [45]. Recommender systems thereby shift the task of finding and suggesting relevant

<sup>4</sup> <https://www.coursera.org/>, retrieved 10.03.2014

<sup>5</sup> <https://www.udacity.com>, retrieved 10.03.2014

---

items from teachers or experts to fellow learners or so called peers [45]. The following section gives an introduction to resource-based learning and explains how recommender systems could support learners during the learning process.

---

## 2.2 Resource-based Learning

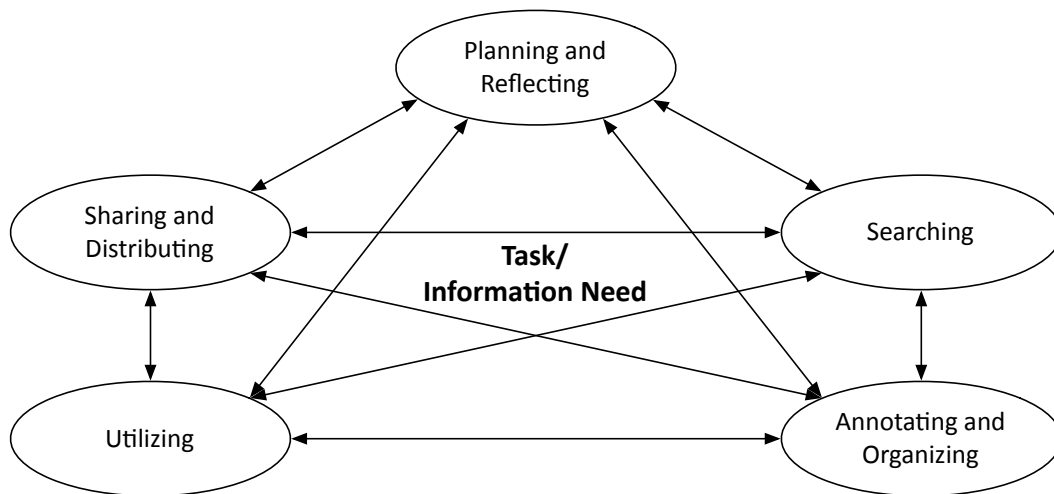
---

Resource-based learning poses challenges for learners not only from a technological perspective but also from a pedagogical perspective as well [196]. There often is no teacher who structures the learning process and prepares the learning material. Additionally, resources on the Web are rarely created by the authors with the intent of creating learning material [16]. The learners, by themselves, have to assess the trustworthiness of the resources and select a variety of these resources which are relevant to their educational goals [16, 196]. Hence, learners have to perform the entire learning process in a self-directed manner. Amongst other challenges, learners have to phrase search terms, select relevant web pages from search results, organize these web pages and structure them to be easily retrievable later on [221]. For example, *Simon* is a student taking part in a seminar about *Environmental Issues*. At the end of the semester, he is required to write up a term paper and hold a presentation on the topic *Challenges of Climate Change*. Other students taking part in the same seminar have been assigned to other related topics. The students are allowed to work individually or in teams on their topics. Most of the students refer to the Web as their primary source of information.

The **Resource-based Learning Model** [196] shown in Figure 2.2 depicts the resource-based learning process. The main tasks are planning and reflecting on the information need, searching for relevant resources, annotating, storing and organizing these found resources and finally using these resources to solve a task and sharing them with others [196]. The specific process steps are listed below. These process steps can be executed independently of one another, they can be executed in any order or not at all.

- In the *Planning and Reflecting* process step, the information need is determined and goals are defined. This could be for example searching for input for a presentation on a certain topic.
- In the *Searching* process step, relevant resources are looked for, mainly on the Web. Search engines, special libraries or specific sources could be used.
- The resources found in the previous step are annotated with notes, keywords or tags and stored in the *Annotating and Organizing* process step. This helps to categorize them and to find them later on.
- The resources are then used in the *Utilization* process step to solve the aforementioned task defined in the planning and reflecting step using the resources found in the searching, annotating and organizing steps.
- The results generated such as slides for a presentation could be shared with others, such as members of a team, in the *Sharing and Distributing* process step.

In the example mentioned above, Simon is taking part in a seminar on *Environmental Issues* and has been assigned the topic *Challenges of Climate Change*. During the planning and reflecting process step, Simon plans how he will find enough information to write up a term paper and prepare a presentation on the topic *Challenges of Climate Change*. In order to achieve his goal, Simon plans to first research generally about climate change, then he plans to make a list of the challenges of climate change and select three major challenges and research in detail about their causes, effects on the environment and possible solutions. In the searching process step, Simon searches on the Web for resources relating to climate change and the challenges of climate change. He finds several official web pages, YouTube videos, and a blog from an environmental activist. In the annotation and organizing step, he tags these resources, sorting them according to their relevance to the three previously selected challenges of climate change. He now starts to write up his term paper and prepare his slides in the utilization step. As he gets more detailed in his writing, Simon finds out he needs more information about certain aspects of his topic.



**Figure 2.2:** Resource-based Learning Model [196]

He therefore searches again for more resources and asks other students working on similar topics in the seminar for input. He realizes others could also benefit from his resources and thus shares his resources with the other students in the seminar in the sharing and distributing step.

In this example, a personalized recommender system could support Simon in finding further resources that other students have already found that are relevant to his particular topic on climate change. Hence, recommender systems could be used to offer support in the *Searching* and *Sharing and Distribution* steps of the RBL model [197]. In self-regulated learning in a RBL scenario, personalized recommender systems act as a scaffold [45], supporting and guiding the learner through the learning process [94, 241]. Thereby, the learner has full control to decide which recommended resources to accept as relevant and which ones to discard as uninteresting [45]. In addition, social tagging systems [40, 126] could be used to organize and manage the resources found in the *Annotating and Organizing* step of the RBL model. Furthermore, the tags could be used to find related resources in the *Sharing and Distribution* step, in particular, recommender systems could exploit these tags to provide recommendations to the learners. Content-based recommender systems [222] as well as knowledge-based recommender systems [67] have been investigated in a RBL scenario, however both approaches do not consider the community data found in online learning communities. The following section introduces social tagging systems and their importance for TEL and specifically for RBL.

## 2.3 Social Tagging Systems

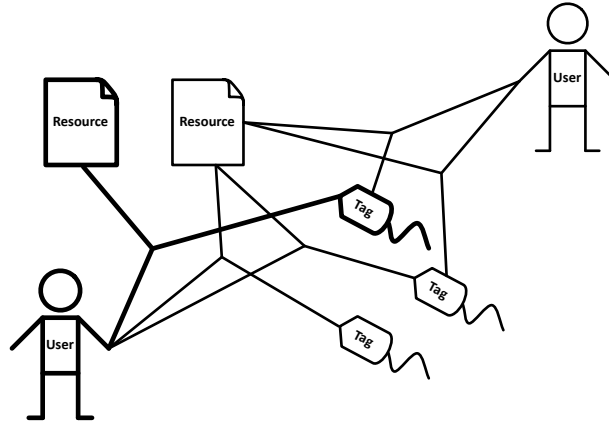
Social tagging systems, also known as social bookmarking applications or social tagging applications, are systems in which users collaboratively attach keywords or tags to resources in order to categorize and retrieve them later on [40, 126]. Examples of some popular social tagging systems are: Delicious<sup>6</sup> for sharing links on the Web, BibSonomy<sup>7</sup> for sharing bookmarks and research papers and CiteULike<sup>8</sup> for managing and sharing links to scholarly articles. A comprehensive overview of the state-of-the-art social tagging systems is given in [40].

Social tagging systems are important for TEL as they can be used to support learners in organizing their learning resources. Collaborative tagging also stimulates reflection in learners as the learners summarize different aspects of the learning resources when tagging [25]. Learners can also learn from each other by viewing other learner's tags. A learner's tagging behaviour also says a lot about the learner's activities

<sup>6</sup> <https://delicious.com>, retrieved 19.02.2014

<sup>7</sup> <http://www.bibsonomy.org>, retrieved 19.02.2014

<sup>8</sup> <http://www.citeulike.org>, retrieved 19.02.2014



**Figure 2.3:** A folksonomy with a tag assignment highlighted

which could be helpful when providing support to the learner [25]. Social tagging systems can be used to offer support to the user during the search, annotation and sharing tasks involved in resource-based learning [15]. For example, a user having found an interesting web page about Greenpeace could save the link to this web page and attach the tags *Global Warming* and *Climate Change* to it. These tags will help the user to search for and retrieve this web page later on. Tags also give a summary of the resource's content [48]. When several users collaboratively tag resources, a network-like structure called a **folksonomy** emerges [186]. In the following sections a folksonomy and an extended folksonomy are introduced.

### 2.3.1 Folksonomy

A folksonomy is formed by the collaborative creation and management of tags used to annotate and categorize content [186]. A folksonomy comprises users, tags, resources and tag assignments. Figure 2.3 depicts an example of a folksonomy with a single tag assignment highlighted: a user, tag and resource and the links between them. An *entity* refers to a user, tag or resource in the folksonomy. A formalized definition of a folksonomy is as a quadruple [102]  $F := (U, T, R, Y)$  where:

- $U$  is a finite set of **users**.
- $T$  is a finite set of **tags**.
- $R$  is a finite set of **resources**.
- $Y \subseteq U \times T \times R$  is a **tag assignment** relation over these sets of entities.

For example, the user  $Gaby \in U$  is a student who is also taking part in the seminar *Environmental Issues* that *Simon* is working on in the previous example. *Gaby* has been assigned the topic *Greenpeace*. During her research on the topic, she finds a web page on Greenpeace  $greenpeace.org^9 \in R$ , and attaches the tags *NGO*, *1971*, *global warming* and *Amsterdam*  $\in T$  to it. Thereby, the tag assignments:  $(Gaby, NGO, greenpeace.org)$ ,  $(Gaby, 1971, greenpeace.org)$  and  $(Gaby, global\ warming, greenpeace.org) \in Y$  are created. *Simon* also finds a web page on global warming  $nrdc.org/globalwarming^{10}$  and attaches the tags *global warming*, *climate change* to it, thus creating the tag assignments  $(Simon, global\ warming, nrdc.org/globalwarming)$  and  $(Simon, climate\ change, nrdc.org/globalwarming) \in Y$ .

The folksonomy structure that thus results from the collaborative tagging of resources can be used to generate tag recommendations [111] or resource recommendations [33] or even to recommend users, experts or teachers [22]. A state-of-the-art survey on social tagging systems and their benefit to recommender systems can be found in [126].

<sup>9</sup> <http://www.greenpeace.org>, retrieved 10.08.2014

<sup>10</sup> <http://www.nrdc.org/globalwarming/>, retrieved 10.08.2014

---

### 2.3.2 Additional Semantic Information - Extended Folksonomy

---

Some social tagging systems provide additional semantic information to a folksonomy, for example groups of tags, ratings for a tag or relations between tags [40]. Folksonomies enriched with additional semantic information are called **extended folksonomy** [1]. Additional semantic information provide contextual information about entities in the folksonomy [1]. For example, resources belonging to the same activity can be seen as belonging to the same semantic context of this activity. Tag types also give additional information about the context of the tag and tag assignment [17]. An example of an extended folksonomy is GroupMe! [2]. GroupMe! is a social bookmarking application supporting the tagging and grouping of resources found on the Web. Groups in GroupMe! allow resources e.g. belonging to a common topic to be semantically grouped together. Groups can also contain other groups [2]. Groups are interpreted as tags i.e. if a user adds a resource  $r$  to a group  $g$  then this is translated as a group assignment, similar to a tag assignment.

Social tagging systems for TEL often incorporate additional semantic information to support the learner, for example **PacMan (Personal Activity Manager)** is a personal learning environment that supports learners in managing their web resources and online tools by offering a simple model of learning activities. Activity structures comprise learning activities, learning resources and learning contexts e.g. at work or at home. Learners organize their web resources and online tools by creating activity structures. This semantic information is well structured and can be exploited to generate recommendations to support specific tasks or activities [177]. Another example is the **Open Annotation and Tagging System (OATS)**. OATS is an open source tool that motivates learners to collaboratively tag learning content in a learning management system [25]. It supports the collaborative creation and sharing of knowledge resources by using highlights, tags and notes that are integrated into the learning content. The **FolksAnnotations Tool Architecture (FAsTA)** is yet another example. FAsTA is a folksonomy-based automatic semantic metadata generator [10] that supports the semantic enrichment of tags by automatically extracting tags from bookmarked web pages. It aims to improve metadata representativeness, quality and validity. **SOBOLEO** is an example of a collaborative tagging system that supports work-integrated learning by encouraging the users to extend their knowledge and improve their knowledge levels [220] .

Related work shows that the exploitation of additional sources of information found in social tagging systems, such as the grouping of resources [1], the user's social network [49], the content of resources or the social relations between users [161] have led to improved recommendations. The resulting challenge therefore is how best to exploit these additional semantic information gained from an extended folksonomy to improve the recommendation of learning resources in a resource-based learning scenario. In the next section, **CROKODIL** is analysed as a concrete example of a social tagging system for RBL in online communities.

---

## 2.4 Application Scenario: CROKODIL

---

In addition to learning, learners have to manage different tasks in the overall process of resource-based learning. It therefore remains a technological challenge to support the learner in all the processes of the RBL model. **Communities, Web-Ressourcen und Kompetenzentwicklungsdienste integrierende Lernumgebung (CROKODIL)** is a platform that supports the collaborative acquisition, structuring and management of learning resources [16]. CROKODIL aims to provide support for all the process steps involved in collaborative resource-based learning [15]:

- CROKODIL is based on a pedagogical concept [198] which focuses on **activities** as the central structure for organizing learning resources in hierarchical tasks or goals [16].
- CROKODIL supports collaborative **semantic tagging** [35] where tags can be assigned tag types such as topic, location, person, event or genre thereby giving the tags added semantic information.

- CROKODIL offers social network functionality to support and encourage collaborative learning amongst the learners on the platform [16]. Groups of learners working on a common activity can be created, as well as friendship relations between learners. Communication channels such as personal and group messages and a chat are integrated into the platform.
- CROKODIL also supports resource-based learning by recommending learning resources to the learners [197]. Recommendations made are content-based [222], knowledge-based [67], and graph-based hybrid recommendations [14, 19], which are the contributions of this thesis.

The following sections introduce semantic tag types, the pedagogical concept of activity hierarchies and CROKODIL's extended folksonomy.

---

#### 2.4.1 Semantic Tag Types

---

Tags are created for different purposes and describe various aspects of a resource. Some tags give organisational information about the tagged resource, others describe the content or context of the resource and others give subjective qualities or opinions about the resource [48]. The different kinds of tags can be classified into several tag types [31]. Some tag types proposed are: person, location, event, genre, topic and goal or task [36]. The tag types can be seen as additional semantic information given to a tag. In the example above, *Gaby*  $\in U$  could give the tag *NGO* the tag type *genre* as the resource *greenpeace.org* is a web page of a non-governmental organisation. She could as well give the tag *1971* the tag type *event* as the founding of the organisation took place in 1971. The tag *global warming* hints at one of the topics of the web page, so this tag could be given the tag type *topic*. The organisation is currently situated in *Amsterdam*, as such this tag gives information about *location*. In addition, the executive director is presently *Kumi Naidoo* and as such this tag could be added and given the tag type *person*.

---

#### 2.4.2 Pedagogical Concept of Activity Hierarchies

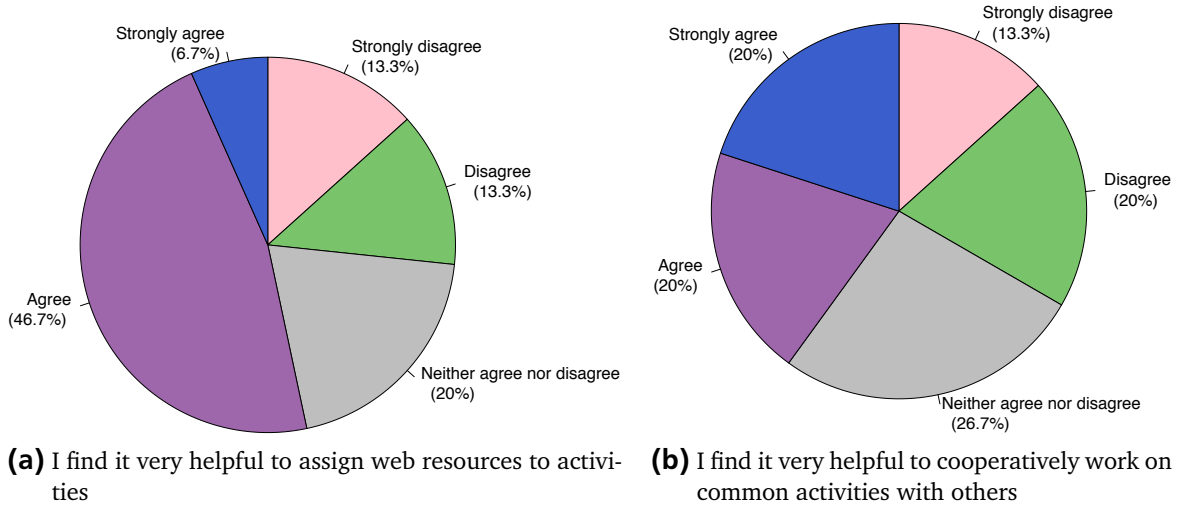
---

Collaborative resource-based learning faces the pedagogical challenge of acquiring the competences needed for self-regulated learning [34, 219]. The aim of the CROKODIL platform is to support learners to achieve these competences [198]. Activities support the learner during the learning process by organizing and structuring the learner's tasks or learning goals that the learner wishes to accomplish within a learning episode. Each learning episode should start with defining the learning goals or tasks to be accomplished by a learner or group of learners as a hierarchical activity structure. The next step is then to find relevant learning resources needed to achieve these goals and attach them to the relevant activities. Activities also support the learners afterwards when reflecting on the learning process. Activities have the following features and characteristics [198]:

- Activities have a meaningful name and description.
- Activities can be structured hierarchically, i.e. an activity can have sub-activities. This helps the learner in breaking down a larger task into smaller, more manageable sub-tasks.
- Relevant resources found during a learning activity are attached to this activity.
- Activities can be worked on collaboratively in groups of learners.
- Learners can store their own documentation of the learning outcome of the activity as a result document attached to the activity.
- Learners can describe their experiences, ask for help or discuss issues with other learners by attaching comments to the activities.

For example, *Gaby* could create an activity called *Finding out what Greenpeace works on* having two sub-activities *Analyzing issues on Climate Change* and *Discovering causes of Toxic Pollution*. She could then attach the resource *greenpeace.org* she had previously found to the activity *Finding out what Greenpeace*





**Figure 2.4:** Learners using CROKODIL find Activities helpful

works on. This way she can structure and manage her research activities and resources better. Furthermore, *Gaby* could work together with *Simon* by creating a sub-activity *Finding solutions to challenges of Climate Change* for the activity *Analyzing issues on Climate Change*. This way *Gaby* and *Simon* could benefit from resources found together on these topics.

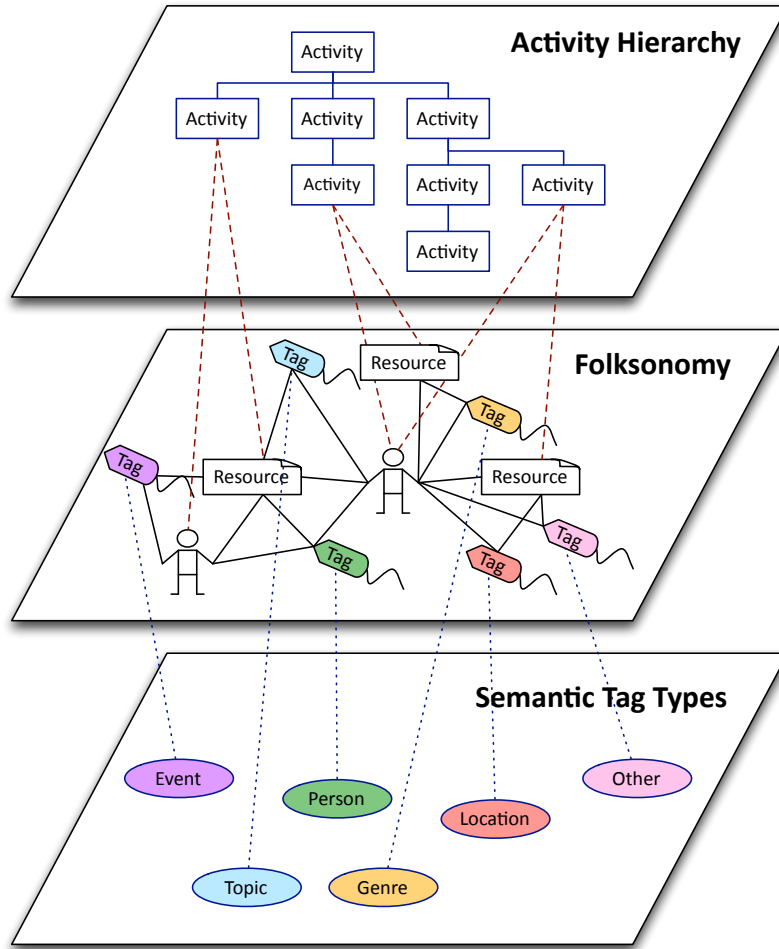
CROKODIL has been applied and evaluated in several learning scenarios such as at the university and in professional educational institutions [198, 200]. The CROKODIL application served not only as a concrete implementation of a resource-based learning scenario, but also as a platform for the requirement analysis of the resource-based learning scenario, and as an evaluation environment and evaluation dataset [66]. A survey of 15 learners using CROKODIL for a resource-based learning episode in a professional educational institution shows, in Figure 2.4 (a), that learners find it helpful to attach resources to activities, and in Figure 2.4 (b), that learners find it helpful to work together with others on common activities.

### 2.4.3 CROKODIL's Extended Folksonomy

CROKODIL offers an extended folksonomy where the additional semantic information gained from activities, semantic tag types, learner groups and friendships extend the folksonomy. CROKODIL's extended folksonomy is defined as [19]:  $F_C := (U, T_{typed}, R, Y_T, (A, <), Y_A, Y_U, G, friends)$  where:

- $U$  is a finite set of **learners**
- $T_{typed}$  is a finite set of **typed tags** consisting of pairs  $(t, type)$ , where  $t$  is an arbitrary tag and  $type \in \{topic, location, event, genre, person, other\}$
- $R$  is a finite set of **learning resources**
- $Y_T \subseteq U \times T_{typed} \times R$  is a **tag assignment relation** over the set of users, typed tags and resources
- $(A, <)$  is a finite set of **activities** with a partial order  $<$  indicating sub-activities
- $Y_A \subseteq U \times A \times R$  is an **activity assignment** relation over the set of users, activities and resources
- $Y_U \subseteq U \times A$  is an **activity membership assignment** relation over the set of users and activities
- $G \subseteq \mathcal{P}(U)$  is the finite set of subsets of learners called **groups of learners**
- $friends \subseteq U \times U$  is a symmetric binary relation which indicates a **friendship relation** between two learners

For example, a user  $u = Susana \in U$  is preparing for a quiz about global warming. To plan her research, she creates an activity *Prepare quiz about global warming* having a sub-activity *Collect historical facts*.



**Figure 2.5:** Additional Semantic Information included to create an extended folksonomy

This means  $A = \{\text{Prepare quiz about global warming, Collect historical facts}\}$  and  $\text{Collect historical facts} < \text{Prepare quiz about global warming}$ . In addition,  $(\text{Susana}, \text{Prepare quiz about global warming}) \in Y_U$  and  $(\text{Susana}, \text{Collect historical facts}) \in Y_U$ . She finds the web page *globalwarming.org*, to which she attaches the tag *U.S.* with tag type *location*, hence  $(\text{Susana}, (\text{U.S.}, \text{location}), \text{globalwarming.org}) \in Y_T$ . She then attaches this resource to the activity *Prepare quiz about global warming*,  $(\text{Susana}, \text{Prepare quiz about global warming}, \text{globalwarming.org}) \in Y_A$ . Susana creates a group *Global warming experts*  $\in G$  and invites *Lucas*  $\in U$  and her friend *Angela*  $\in U$  to help her gather more facts about global warming. It would be helpful for *Susana* to know about the resource *greenpeace.org* that *Gaby* had found earlier on and perhaps had already attached to another activity called *Investigate the impact of greenpeace*. This is where recommender systems can assist in recommending such relevant resources found in the folksonomy to other users working on similar or related activities.

Information in folksonomies are valuable for recommending resources in social tagging systems as well as on the Web [186]. As new Web 2.0 applications emerge providing additional semantic information, it becomes promising to incorporate this semantic information to improve recommender approaches [6]. In the CROKODIL application scenario, the additional semantic information gained from semantic tag types, activities and activity hierarchies is considered in this thesis to extend the folksonomy as depicted in Figure 2.5.



---

## 2.5 Summary

---

In this chapter, the application scenario for this thesis is introduced, focusing on resource-based learning as a special form of self-directed learning with resources found on the Web. Learning resources found on the Web can be tagged collaboratively in social tagging systems, thereby creating a folksonomy structure that can be used by recommender systems to generate recommendations. Additional semantic information found in social tagging systems could be used to create an extended folksonomy. Additional semantic information, gained for example from semantic tag types and activity structures, could be exploited in order to improve the recommendation of learning resources in a resource-based learning scenario. Personalized recommender systems for TEL are the focus of this thesis and will be dealt with in more detail in Section 3.1.3 of the following Chapter 3.



---

### 3 Fundamentals and Related Work

---

This chapter on fundamentals and related work gives the background needed for a better understanding of the following chapters. It also gives an overview of the state-of-the-art recommender systems, explaining briefly the various kinds of recommender systems, their advantages and disadvantages. Following this, recommender systems for social tagging systems are presented and in particular recommender systems for TEL. An introduction is given to evaluation methodologies for recommender systems, with the focus again on evaluation methodologies for TEL recommender systems. After this, a survey of the evaluation methods for TEL recommender systems is presented. The chapter ends with an analysis of the open issues identified in related work with respect to the research goals of this thesis based on the application scenario presented in Chapter 2.

---

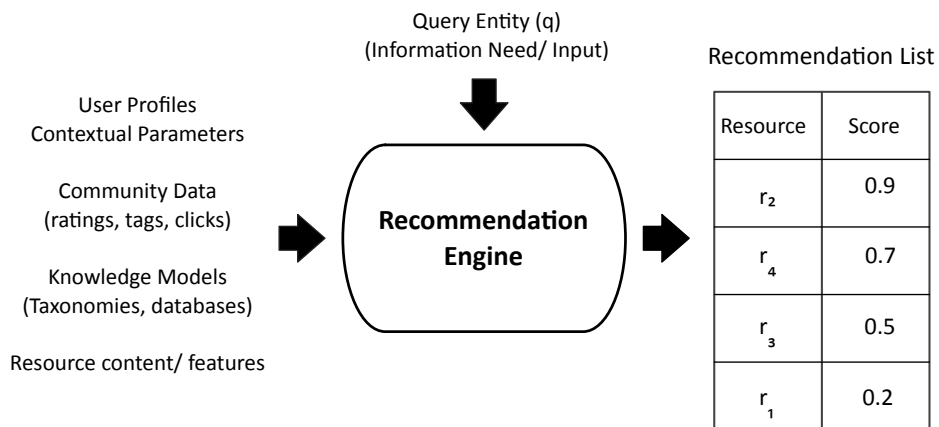
#### 3.1 Recommender Systems

---

Due to the ever increasing amount of information found on the Web, it becomes more and more difficult for a user to find suitable items to satisfy a particular information need. Recommender systems aim to meet this information overload challenge by predicting relevant resources to a user. A recommender system is basically made up of a recommendation engine that given an input, creates a list of recommended resources sorted by their relevance scores. A relevance score is the score on a scale given to a resource in accordance to its relevance to the user's *information need*. In order to provide personalized recommendations, which are recommendations tailored to a particular person, the user profile (e.g. ratings, preferences or transaction history) and user context need to be considered when generating recommendations [110]. Figure 3.1 shows the recommendation process as adapted from [110].

Given a certain user  $u$  as input, the resource recommendation task is to find a resource  $r$  which is relevant to this user. The recommendation task is also known as **ranking** in information retrieval [153]. A ranking algorithm computes for a specified user  $u$ , a score vector  $score(r)$  that contains the scores for each resource  $r$ . These scored resources are then ordered according to their score values, thus forming a *ranked list* of recommendations with the highest scored resource at the top. The top ranked resources e.g. the top ten, are recommended to the user  $u$  after filtering out resources the user already has. For example, the scores:  $score(r_1) = 0.2$ ,  $score(r_2) = 0.9$ ,  $score(r_3) = 0.5$  and  $score(r_4) = 0.7$  create a ranked list:  $r_2, r_4, r_3$  and  $r_1$ . Therefore the top recommendation for user  $u$  would be resource  $r_2$ .

The input or information need of a recommendation or ranking task is also known as a *query entity* ( $q$ ). For a recommendation task, there can be several queries  $q \in Q$ , where  $Q$  is a set of query entities.



**Figure 3.1:** The Recommendation Process [110]

---

Query entities can also be of different types. In a folksonomy this could be the users, tags or resources of the folksonomy [160]. In this thesis, the query entities will be query users  $q_u \in Q_u$ , who are all users  $u \in U$  found in the folksonomy  $F$ .

---

### 3.1.1 Classification of Recommender Systems

---

There are various types of recommender systems, each having their advantages, disadvantages and target domains. The main types are: content-based, knowledge-based, collaborative filtering and hybrid recommender systems.

---

#### Content-based Recommender Systems

---

Content-based approaches determine the similarity between resources by comparing their textual content. The fundamental principle of content-based approaches is the identification of common features of resources that have received positive feedback, for example in the form of ratings from a user, in order to recommend resources that have similar features [110, 148]. Text documents or resources with textual descriptions are often recommended using content-based approaches. Content-based approaches do not consider community data [110] and are faced with the over-specialization problem as they tend to recommend only similar resources to a user and rarely any new, serendipitous resources [148].

---

#### Knowledge-based Recommender Systems

---

Knowledge-based approaches use external knowledge models or data sources such as ontologies, taxonomies or knowledge bases such as Wikipedia to determine relationships between resources and to provide recommendations based on these relationships [110]. There are two main types of knowledge-based recommender approaches [110]:

- **Constraint-based** approaches which are based on recommendation rules.
- **Case-based** approaches which are based on identifying resources having similar requirements.

A major drawback of knowledge-based approaches is the cost of building adequate knowledge models and knowledge bases by domain experts. Knowledge-based recommender systems do not consider community data [110].

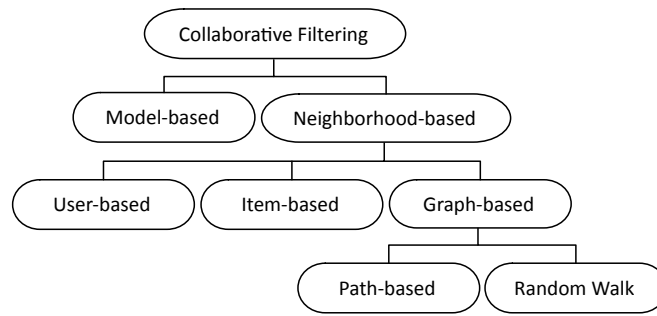
---

#### Collaborative Filtering Recommender Systems

---

Collaborative filtering, also known as social filtering, exploits the collaborative activities of users in a community to determine which resources to recommend. Collaborative filtering approaches do not rely on the textual content of the resources nor on external sources of knowledge but rather on community data such as feedback, ratings, tags or click history to make recommendations to similar users [110]. The fundamental assumption made by collaborative filtering approaches is that users with similar preferences in the past, will also have similar preferences in the future [110]. Collaborative filtering approaches can be grouped into two major classes [61]. A classification of collaborative filtering approaches are shown in Figure 3.2 [7, 110, 201].

- **Model-based** approaches take ratings from users in the system in order to train a predictive model. After training, the predictive model is used to determine ratings of users for new resources. Examples of such predictive models are: Bayesian Clustering, Latent Semantic Analysis (LSA) and Support Vector Machines (SVMs) [21]. These models are complex and the recommendations cannot be intuitively explained to the user [61].



**Figure 3.2:** Classification of Collaborative Filtering Recommender Systems

- **Neighborhood-based** approaches use feedback or ratings from users in the system to predict ratings for new resources. The advantages of neighborhood-based approaches lie in their intuitiveness and simplicity to implement. There are no expensive training phases, predictions can be precomputed offline and relative little memory is required. They are very stable as new resources have a minimal effect on predictions. Additionally, they have the advantage of being able to recommend very different resources to what the user is used to [61]. Neighborhood-based approaches can further be classified into user-based, item-based and graph-based approaches [61]:
  - **User-based** neighborhood recommender approaches predict the rating of a user for a new resource by using the ratings given by users most similar to this user, called nearest-neighbors. The key idea is that the rating of a user for a new resource is likely to be similar to that of another user, if these users have rated other resources in a similar way.
  - **Item-based** neighborhood recommender approaches are analogous to user-based but the ratings to similar resources are considered instead.
  - **Graph-based** approaches exploit the transitive relations between nodes that are not *directly connected*. Thus they avoid problems like data sparsity and limited coverage which are problems faced by user-based and item-based neighborhood collaborative filtering approaches [61].

### Graph-based Recommender Approaches

Graph-based recommendation techniques are also known as weight-spreading ranking techniques and have their origins in the field of Web Search and Link Analysis [234]. This is why graph-based approaches are also known as ranking algorithms. Link analysis and the structure of the Web have led to effective information retrieval methods. Link analysis on the Web is based on citation analysis, where the hyperlinks between web pages are considered as a citation between pages [153, 234]. Graph-based recommender systems [1, 48] consider the graphical structure when recommending resources. In social tagging systems, the users, tags and resources are represented as nodes in a graph and the transactions or relationships between them are represented as edges. The following two characteristics are common for most graph-based approaches [61]:

- *Propagation*: Graph-based approaches enable nodes that are not directly connected to still have an influence on each other by propagating information over other nodes and edges between them in the graph. The greater the weight of an edge, the more information can be propagated through it.
- *Attenuation*: The influence of one node on another should be smaller if the two nodes are further away from each other in the graph.

Graph-based approaches are also suitable for integrating data from different systems as the graph can be used as a common basis for interconnection, for example when connecting vocabularies, ontologies, social networks, technologies from the social semantic Web, linked open data [82], in personal learning environments [97] or for generating recommendations across several learning platforms [18, 81]. Graph-based approaches can be classified into the two following groups [61]:

- **Path-based** approaches utilize path-based similarity which the distance between two nodes of the graph by considering the number of paths connecting the two nodes, as well as the length of these paths. An example is Shortest Path [61].
- **Random Walk** approaches are based on random walk similarity. A probabilistic framework defines transitive associations where the similarity between nodes is determined as a probability of getting to these nodes in a so called *random walk* [61]. A random walk is often depicted as a weighted graph with a node representing each state. The weight of the edge between two nodes determines the probability of jumping from one node to the other. An example is the state-of-art approach *PageRank* [41, 185]. The main idea in PageRank is that a web page is important if many other web pages link to it, especially if these web pages are important themselves. The PageRank algorithm and the *Random Surfer Model* are explained in detail in Appendix D.

---

### Hybrid Recommender Systems

---

Hybrid recommender systems combine the aforementioned techniques thereby utilizing the advantages of the individual recommender approaches [46]. There are several ways to create hybrid recommender systems [47]:

- **Weighted** hybrid recommender systems are created when the results of two or more recommender systems are combined to a new list of resource recommendations by creating a union or an intersect of the recommended resources. The scores of the recommended resources in the new list are weighted and combined and the final recommendation is made.
- **Mixed** hybrid recommender systems are created when two or more recommender systems calculate ratings for the given list of resources and the best ranked for each system are recommended to the user after duplicates are filtered.
- **Switching** hybrid recommender systems switch between different individual recommender systems thereby utilizing their advantages and avoiding their disadvantages. This is achieved by analyzing the recommendation situation and selecting, based on predefined criteria, which basic recommender system is to be used and when.
- **Cascade** hybrid recommender systems comprise an initial recommender system that calculates candidates to be recommended and their scores. These recommendation results are then given on to a second recommender system that re-rates them. The combined score from both recommender systems is finally used to generate recommendations for the user.
- **Feature Combination and Feature Augmentation** hybrid recommender systems do not combine different recommender systems together, but rather one recommender system (for example a graph-based approach) combines different knowledge sources together. These knowledge sources could be the output from another recommender system, for example a collaborative filtering or content-based recommender system [27]. A recommender system could improve the database of resources, ratings and user profiles by including additional information and after this another recommender system creates recommendations for the user using this improved database [47].

---

#### 3.1.2 Recommender Systems for Social Tagging Systems

---

In social tagging systems, users may be interested in finding resources, tags, or even other users, hence recommender systems need to recommend various types of items. Usually, a recommender system generates resource recommendations, tag recommendations or user recommendations. Recommender systems that recommend more than one entity type (or mode) are called *multi-mode recommender systems* [161], and are also known as multidimensional recommendations or context-aware recommendations [161]. Multi-mode recommender systems could have a single input or multi-input [161]. Table 3.1 gives an overview of multi-mode recommendation tasks adapted from [33] and extended with multi-input tasks from [59, 161, 192, 202].

Query Entity	User Recommendation	Resource Recommendation	Tag Recommendation
User	User related users, user browsing, more users like me, users with same interests (my neighbours)	User related resources, resource recommendation, interests match	User related tags, get to know users, user profiling, tag recommendation
Resource	Resource related users, resource experts	Resource related resources, more (resources) like this	Resource related tags, index suggestion, tag recommendation, tag assignment
Tag	Tag related users, tag experts, domain experts	Tag related resources, personalized search, guided search	Tag related tags, depth browsing, semantic relations
User and Resource	User and resource related users, resource experts in my neighbourhood	User and resource related resources, personalized more (resources) like this	User and resource related tags, personalized tag recommendation
User and Tag	User and tag related users, domain experts in my neighbourhood	User and tag related resources, personalized search	User and tag related tags, personalized semantic relations
Resource and Tag	Resource and tag related users, resource experts in a specific domain	Resource and tag related resources, contextual more (resources) like this	Resource and tag related tags, semantic relations in a defined context

**Table 3.1:** Single input and Multi-input recommendation tasks adapted from [33, 59, 161, 192, 202]

- **Single Input:** Given a query entity of class  $i$ , predict an entity of class  $j$ , where  $i, j \in U, R, T$ .
- **Multi-input:** Given a query entity of class  $i$  and an entity of class  $j$ , predict an entity of class  $k$ . Usually,  $i \neq j$ .

Recommender systems for social tagging systems are often called *tag-based recommender systems* or *folksonomy-based recommender systems*. Various kinds of recommender techniques have been applied to social tagging systems. The kind of recommender technique applicable depends on the way the folksonomy model is represented. The state-of-the-art recommender techniques are based on factorization models or graph-based models [161].

---

### Graph-based Recommender Systems for Folksonomies

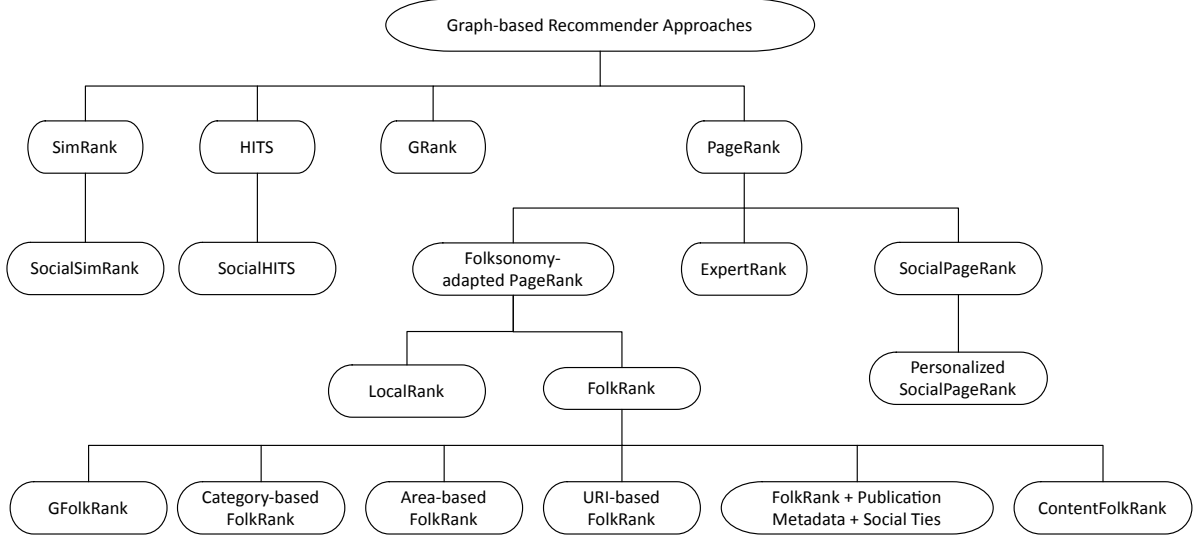
---

Graph-based recommendation methods for folksonomies require the folksonomy to be modeled as an undirected, weighted graph. The nodes represent the users, tags and resources in the folksonomy and the edges the tag assignment relations between them. Below is a formal definition of a folksonomy graph [1].

#### Folksonomy Graph

In the following, three sets are defined that will be used below to determine the weights of the edges in the folksonomy graph  $G_F$  below. For a given user  $u \in U$ , tag  $t \in T$  and resource  $r \in R$ :

- Let  $U_{t,r} = \{ u \in U \mid (u, t, r) \in Y \} \subseteq U$  be the set of all users that have assigned a tag  $t$  to resource  $r$
- Let  $T_{u,r} = \{ t \in T \mid (u, t, r) \in Y \} \subseteq T$  be the set of all tags that user  $u$  assigned to resource  $r$



**Figure 3.3:** Graph-based Recommender Approaches

- Let  $R_{u,t} = \{ r \in R \mid (u, t, r) \in Y \} \subseteq R$  be the set of all resources that user  $u$  assigned a tag  $t$

Given a folksonomy  $F$ , the folksonomy graph  $G_F$  [1, 103] is defined as an undirected, weighted graph  $G_F := (V_F, E_F)$  where:

- $V_F = U \cup T \cup R$  is the set of nodes
- $E_F = \{ \{u, t\}, \{t, r\}, \{u, r\} \mid u \in U, t \in T, r \in R, (u, t, r) \in Y \} \subseteq V_F \times V_F$  is the set of undirected edges
- Each of these edges is given a weight  $w(e), e \in E_F$  according to their frequency within the set of tag assignments:
  - $w(u, t) = |R_{u,t}|$  the number of resources that user  $u$  assigned the tag  $t$
  - $w(t, r) = |U_{t,r}|$  the number of users who assigned tag  $t$  to resource  $r$
  - $w(u, r) = |T_{u,r}|$  the number of tags that user  $u$  assigned to resource  $r$

$\vec{w}$  is a weight vector having a weight value for each node in the folksonomy graph [111]. The weights of the nodes are defined thus:

- $w(u)$  is the weight of a user  $u \in V_F$
- $w(t)$  is the weight of a tag  $t \in V_F$
- $w(r)$  is the weight of a resource  $r \in V_F$

The initial weights of the nodes in a folksonomy are usually uniformly set, for example  $\vec{w}(u) = \vec{w}(t) = \vec{w}(r) = 1$ . The recommendation task of a graph-based recommender algorithm is thus to calculate the final weights or so called scores *score* (as introduced in Section 3.1) for each node in the folksonomy.

There are many graph-based recommender approaches for folksonomies. An overview is shown in Figure 3.3. Many of these approaches are based on link-analysis algorithms for the Web [234]. This is due to the intuition that the links between entities in a folksonomy can be seen as analogous to the links between web pages on the Web. These approaches however have to be adapted to the folksonomy graph.

### FolkRank

One of the most popular graph-based recommender approaches for folksonomies is **FolkRank** [103]. FolkRank is based on the folksonomy-adapted PageRank [41, 103, 185] computation. The FolkRank computation can be illustrated with the random surfer model of PageRank [41, 185] where the folksonomy



graph  $G_F$  is created from a folksonomy  $F$  as described above. The intuition from PageRank is translated to folksonomies, whereby a resource becomes increasingly important the more often it is tagged with important tags by important users. The same holds for tags and users. The surfer probabilistically traverses the graph, taking into consideration the edge weights of the undirected edges. The more often a node is visited by the surfer, the higher it appears in the final ranking. The visit rate is said to be the node's final weight  $\vec{w}$  or score *score*. To support having a query node  $q \in Q$ , which is usually a user  $q_u \in U$  in a folksonomy, the surfer jumps with a certain probability to the nodes that represent the set of query entities  $Q$ . This corresponds to the biased surfer model of PageRank [96], see Appendix D for more details. In order to downplay the high scores of nodes that are scored high due to the structure of the folksonomy graph, the scores obtained from the random surfer model are subtracted from the scores of the biased surfer model in the last calculation step of FolkRank [103].

1. Calculate a global PageRank  $\vec{w}^{(0)}$  where the personalization vector is not used.
2. Calculate the personalized PageRank  $\vec{w}^{(1)}$  where some nodes are given priority (biased surfer model).
3. Calculate the FolkRank i.e. the difference  $\vec{w} = \vec{w}^{(1)} - \vec{w}^{(0)}$ .

The final weights of the nodes  $\vec{w}$  are called the folkrank scores  $score_{FolkRank}^{\vec{w}}$  and are used to form a ranked list of recommendations. FolkRank can recommend users, tags or resources for any specified query entity. There have been several extensions made to FolkRank:

- **GFolkRank** extends FolkRank by integrating additional semantic information gained from the grouping of resources in GroupMe! into the folksonomy graph [1]. In GroupMe!, resources can be tagged and can also be assigned to a group. Groups are interpreted as tags; this means if a user adds a resource to a group, then GFolkRank translates this as a (tag) group assignment. The folksonomy graph is thus extended with additional group nodes and group assignments as new edges. These edges are all given a constant weight, as a resource is usually added only once to a particular group. This constant weight is determined as the maximum weight of all tag assignments involving this resource. This is based on the assumption that a group assignment is considered more semantically valuable than a tag assignment. FolkRank is then run on this extended graph [1].
- **Category-based FolkRank** extends FolkRank by exploiting additional information gained from tag categories. In the social tagging system TagMe! [5], a category dimension is provided in order to classify tag assignments. A user can freely create and assign one or more categories to each tag assignment. The folksonomy is extended with these categories. Categories are seen as tags and a category assignment is treated similar to a tag assignment. The intuition is that resources sharing the same category are related. The folksonomy graph is extended with category nodes and category assignments that relate the category with the resource and tag. The weights of the edges represent the number of times a category is assigned to a tag assignment involving a particular resource [5].
- **Area-based FolkRank** extends FolkRank by exploiting spatial information from a tagged area in a resource [5]. Spatial tag assignments are tag assignments to a specific area of a resource. Users in TagMe! can attach an area tag to a specific section of an image, for example by specifying a rectangular area in a picture. Area-based FolkRank does not extend the folksonomy graph, but rather adjusts the weights on the edges. The larger the area tagged, the more important the tag is for that resource. For each tag assignment, the weight of an edge between a tag and a resource is increased when the tag assignment has spatial information attached to it. The increase in weight depends on the size of the area tagged and its position in the image [5].
- **URI-based FolkRank** extends FolkRank by using the additional information gained from a unique resource identifier (URI) that describes the semantic of a tag assignment [5]. A URI leads to an ontology providing semantic information about the tags. A URI can thus be seen as a tag or ontological concept. The folksonomy graph is extended with URI nodes and edges between the URI, tag and resource. This added information about the tag and tag assignment can be used to disambiguate the meaning of a tag in TagMe! [5].

- **FolkRank extended with publication metadata and social ties** for the recommendation of scientific publications [65]. This additional semantic information is gained from the metadata found in scientific publications, such as the author's name, the year and the venue of the publication. Additionally, the social ties amongst users and user groups from the social tagging system BibSonomy [102] are incorporated into the graph model for FolkRank. The year of posting of the publication was also considered as recently posted publications were seen as being more important to the user. The semantic structure among tags was also utilized by creating sets of similar tags and including these connections between tags in the folksonomy graph [65].
- **ContentFolkRank** extends FolkRank by leveraging content data of resources. The folksonomy graph is extended by replacing a resource node with its term nodes. To achieve this, each resource is broken down into its terms and each tag assignment is converted into a set of tag assignments with these terms. This enables tags to be recommended for new resources by utilizing the information gained from their textual content [141].

### LocalRank

LocalRank is a neighborhood-based tag recommendation algorithm. It optimizes the FolkRank approach by considering only a small part of the folksonomy graph for its computation [139]. Instead of considering all entities in a folksonomy, LocalRank focuses only on the relevant ones and selects only the local neighborhood of a given user and resource. Similar to FolkRank, the weight propagation and rank computation is performed, but without iterations. Like PageRank, LocalRank can indicate a higher preference towards a certain user, resource or tag in the folksonomy. No additional sources of information are considered in the computation.

### SocialPageRank and Personalized SocialPageRank

SocialPageRank [23] is adapted from PageRank and follows the same intuition as FolkRank. The more tags a user gives a resource, the more important this resource becomes to this user. The more resources a user has tagged the more important this user becomes and the more resources a tag is assigned to, the more important this tag becomes. As in PageRank, the weight spreading process converges in a SocialPageRank score. **Personalized SocialPageRank** adapts SocialPageRank to be able to give preference to a certain node. This means a node specified as a query, for example a user node or a tag node, can be given more preference in the computation, thereby making SocialPageRank personalized and topic-sensitive [6]. No additional sources of information are considered in the computation.

### ExpertRank

ExpertRank is an expert ranking algorithm based on PageRank and the tagging activity of users which determines a user's expertise in the tagging community. Tagging activity creates social networks of users around tags thereby representing the interests and expertise of users. The expertise of a user is calculated by the number of resources the user has tagged with a particular tag and the age of the tag. Tags are clustered based on how many common resources they have been assigned to. An expert with a specified set of skills is translated to be a user who has assigned a set of tags to a resource. This set of skills are mapped to the set of tags. Clusters of tags represent skill sets and indicate an overlap in expertise of the users. The ExpertRank of a user is computed as the weighted sum of the ExpertRank of the user for the various tags based on PageRank. The personalization vector is the activity of the user for each tag normalized over the user's activity across all tags [115].

### GRank

GRank is a graph-based approach that considers group structures in folksonomies. GRank calculates a ranking for all resources which are related to a query tag thereby exploiting the group structure in GroupMe!. It considers the tags of a resource, the tags of groups the resource belongs to as well as the tags of other resources in common groups with the resource. The weights of the edges between a tag and

---

a resource are determined by counting the number of users who have assigned a resource with this tag to a group. With multiple query tags, GRank accumulates the individual GRank values. The weights of directly assigned tags, tags assigned to neighbor resources, tags assigned to a group the resource belongs to, and tags assigned to resources belonging to the same group are emphasized. Directly assigned tags are given the highest preference, and tags from neighbor resources the least [1].

### **SocialSimRank**

SocialSimRank is an extension of the SimRank algorithm to fit folksonomies [23]. SimRank [112] is a ranking algorithm that calculates the relevance of a web page to a given tag. It is based on link analysis [234] and adopts the random surfer model of PageRank. SocialSimRank makes use of tags to calculate the similarity between a query and a web page. The semantic similarity is calculated between the search terms from the query and the tags of the web page [23].

### **SocialHITS**

SocialHITS is based on the HITS algorithm [129], which similar to PageRank is based on link analysis [234]. HITS ranks web pages by giving two scores to each page in the web graph - a hub score and an authority score. A hub score of a page is the number of incoming links from pages with high authority scores. An authority score is the number of incoming links from pages with high hub scores. These scores are calculated recursively until the scores converge to stable values [129]. SocialHITS adapts the HITS algorithm to folksonomies [3]. To achieve this, a directed folksonomy graph is created in order to determine the hubs and authorities. An edge between a user and a tag and between the user and the resource is created for all resources a user has assigned a tag to. An authority user is the first user to tag a popular resource that is later tagged by many other users [3].

---

## **3.1.3 Recommender Systems for Technology Enhanced Learning**

---

In Chapter 2, the diverse research areas of TEL are presented and the importance of recommender systems for TEL is explained. Personalized recommender systems for TEL support learners by suggesting relevant learning resources, learning activities, learning paths, other learners, experts and tutors [159]. Content based and collaborative filtering recommender techniques have been very useful in TEL scenarios, especially in informal learning [157]. A survey of existing TEL recommender systems can be found in [159] and of context-aware recommender systems for TEL in [239].

---

### **Requirements of Recommender Systems for TEL**

---

The main aim of recommender systems for TEL can be summarized as supporting learners during the learning process to accomplish their learning goals [159, 165]. Personalized recommender systems make sense in a TEL scenario as their characteristics can be mapped to corresponding principles in the learning sciences that are needed to facilitate learning [45]. These particularities and challenges lead to new requirements for recommender systems for TEL as compared to other domains [157]. It thus becomes increasingly important to develop and evaluate recommender systems that consider these particularities and requirements.

Learners should be confronted and challenged with unexpected content as this would encourage the learner to learn through discovery and exploration [45]. Recommending learning resources that are different to those a learner already knows could stimulate critical thinking and counter confirmation biases [45]. There is thus a need to recommend resources beyond their similarity [159]. Recommender systems for TEL should recommend novel [159] and serendipitous resources [45], for example by providing preference-inconsistent recommendations [224].

Learning is a very individual process and learners have very individual needs [165]. Learners also have specific long term learning goals [72]. Thus recommender systems for TEL need to consider the learning

---

goal [159] and learning activities of the learner when providing personalized, topic-sensitive recommendations [45]. In order to determine the learner's knowledge and learning activities, recommender systems for TEL need to be context-aware [8].

---

## TEL Recommender Systems for Folksonomies

---

There exist some recommender systems specifically for social tagging systems in TEL scenarios, for example for a personal learning environment like ReMashed [73], or other collaborative filtering approaches for folksonomies, such as [53, 128, 210]. The following are examples of graph-based TEL recommender systems for folksonomies.

### 3A ranking algorithm

The 3A contextual ranking system is a personalized context-aware recommender system for a Personal Learning Environment (PLE). It is based on PageRank and considers social relations and relations between resources. Actors, researchers, assets, work packages and group activities are recommended. A graph is created having actors, activities, assets, as well as roles and tags as nodes. The edges represent the relations between these nodes gained from the user's interactions. The edges are given weights according to the importance of the relations between the two nodes. A learner's context is defined by selecting several nodes of the graph, either as related to a query term or directly connected to a query node. The 3A ranking algorithm is then run on this extended graph. The ranking is personalized and contextualized by biasing the random surfer towards the learner's node and the selected nodes defining the learner's context [97].

### Random-Walk recommender system

A personalized context-aware recommender system for TEL based on a random-walk on a folksonomy is proposed by [107]. Learning materials, tutors or other learners with common interests are recommended. The tags found in a collaborative tagging system are combined with the user's social tags from external social networks such as Facebook<sup>1</sup>, Twitter<sup>2</sup> or LinkedIn<sup>3</sup>. The user's social tags comprise the user's profile information like age, education, location, the user's role as tutor or learner, learning goals and topics of interest. A graph is created where the nodes are learning materials, learners, tutors and tags, connected together via tags. Tags are directly connected with each other according to a set of personalized rules belonging to each user. These personalized rules define the edges between the tags and give a weight to the edges. A random walk with restarts is run on this graph. The rules are attuned to the user's feedback to make the recommendations more personalized and relevant to the user's current interests and learning goals [107].

### Social Semantic Web FolkRank

This is an extension of FolkRank as a TEL recommender algorithm for personal learning environments, where resources are ranked according to the relevance of their tags as well as the user's tag-based attention profile. The learner's attention profile is generated from the tags that are most often used by the learner and his friends. The frequency of a tag's usage is calculated based on the tag co-occurrences, tag similarities between tags, and affinities (based on the frequency of the relations) between tags and resources. The tags are clustered and an affinity between the user and a cluster of tags is computed. Additionally, the same extraction is done for the resources and social connections of the learner's friends and the resources of the learner's friends' friends as well. The extracted information is saved in a semantic repository using several interconnected vocabularies to describe them. A folksonomy graph is

---

<sup>1</sup> <http://www.facebook.com>, retrieved 28.05.2014

<sup>2</sup> <http://twitter.com>, retrieved 28.05.2014

<sup>3</sup> <http://www.linkedin.com>, retrieved 28.05.2014

---

created with this information and resources and users are ranked based on FolkRank whereby the relevance of the tags and the importance of the users are considered. The interests of the learner are represented by the importance of the tags used by the learner and his friends in the learner's tag-based attention profile. The preference vector comprises the learner's tag based attention profile, thus making the recommendations personalized and fitted to the learner's interests [188].

### **Relation Based Importance Ranking**

This approach recommends learning resources and is based on PageRank's random walk. The graph comprises nodes which are resources, users, tags, departments, and categories. Categories are the topics the resources belong to. Departments are the departments at a university to which the user belongs to. Categories are managed by departments. The edges of the graph are user-resource, user-tag, tag-resource, resource-category, category-department and department-user. Transition probabilities are determined for the edges. The assumption is that the importance of a resource can be influenced by the resource's relations to other typed objects. Thus the recommendation is based on the relations to other typed objects depending on their transition probabilities [244].

### **Usage-based Clustering**

MACE [184] is a learning object repository where tags represent user interests that help users manage their learning objects. Usage-based clustering is used to relate learning objects based on their usage. The assumption is that in a TEL scenario, learning objects used in the same learning session or time-span while solving a particular task, are most probably semantically related. A usage graph is created by connecting learning objects that have been used together in the same session. The more often two learning objects are used together in different learning sessions, the stronger their common usage and connection to a common task and thus a stronger semantic relatedness to each other. The edges between learning objects are given weights to reflect the strength of their common usage-relatedness. Edges are created between learning objects and between users and learning objects. A graph-based clustering algorithm such as the random-walk based Markov Clustering or Iterative Conductance Cutting is applied to the usage graph. The resulting clusters reflect the usage-based semantic relatedness between the learning objects [184].

### **Tag-based Prior Knowledge Recommendation System (TAK)**

TAK is part of a collaborative system for learning English as a foreign language. TAK assists learners in becoming familiar with the learning content by identifying the key concepts of articles, thus complementing their existing language skills and helping them attain higher comprehension and reading performance. TAK considers social network analysis based on PageRank, tag preferences and tag relevance [54]. A learner attaches tags that represent the topic or idea explained in that paragraph, thus building a network of paragraphs showing the paragraph relations within an article. PageRank is run on this network graph and the scores for each paragraph are determined by summing up the tag scores for each paragraph. The tag scores are modified according to the type of tag: topic, normal and ubiquitous. To determine the importance or representativeness of a paragraph in an article, a tag-item relevance is calculated as the similarity between sentences. User-tag preferences are determined by defining different types of tags such as topic tags or normal tags. This depicts which paragraphs are important to a user. The position of the tag in the article is also considered. For the paragraphs the learner is unfamiliar with or does not understand, key sentences are selected by calculating the similarity with the topic tags and recommended tags. These key paragraphs are then recommended to the learner along with the article to be read [54].



---

## 3.2 Evaluating Recommender Systems

---

In order to develop recommender systems that fit specified requirements and to be able to compare the performance of different recommender systems, experiments are conducted. Evaluation metrics are applied to determine the effectiveness of a recommender system according to certain criteria related to the specified requirements. Many of the evaluation methodologies and evaluation metrics used to evaluate recommender systems are inspired by machine learning or information retrieval [21, 153] or by the social sciences [224]. Recommender systems are generally evaluated by performing offline experiments, or by conducting user experiments or a combination of both [98].

---

### 3.2.1 Classification of Evaluation Methodologies for Recommender Systems

---

The evaluation of recommender systems can be classified into three types of evaluation approaches according to [159, 227]: offline experiments, user studies and real life testing. These approaches are explained in the following sections.

---

#### Offline Experiments

---

Offline experiments use pre-collected historic or simulated synthetic data to evaluate recommender systems. Offline experiments try to simulate the real process where recommendations are given to a user and the user acts on this by giving feedback like tagging, rating or buying the resource [227]. The simulation is basically done by hiding or withholding a part of the interactions of each user in a *dataset* and letting the recommender system predict these hidden interactions. Depending on the domain, interaction data could be ratings, tags, purchases or clicks [98]. The recommender system is then evaluated according to how good it can predict these hidden interactions. Offline experiments are also called *dataset driven evaluation* [238]. A dataset consists of synthetic or historical user interaction data.

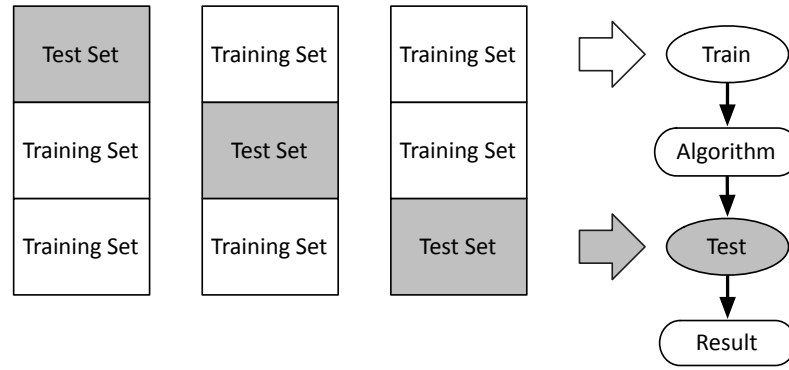
- **Natural or Historical Datasets:** A dataset can be created by collecting historical interaction data of real users in a real system over a period of time. Explicit user ratings could be collected or implicit user feedback extracted from user log data. Examples of well known publicly available datasets are the MovieLens dataset<sup>4</sup> or the Delicious or Last.fm datasets<sup>5</sup>. A list and description of some TEL datasets e.g. MACE, APOSDLE and ReMashed can be found here [238]. As datasets could be very large, they are usually pre-processed or pre-filtered to reduce their size, for example by excluding resources with low tag counts. This helps to reduce the cost of running evaluations on them, however it is important to make sure the resulting dataset still reflects the real life scenario [227].
- **Synthetic Datasets:** A dataset can be simulated or constructed manually. Parameters such as the distribution of user properties, size or rating sparsity [110] can be pre-defined and tuned in order to create datasets fulfilling specified characteristics. Synthetic datasets are usually used to test how recommender algorithms perform in constructed scenarios and under certain conditions. Evaluating TEL recommender systems using simulations on synthetic datasets have played an important role, especially when natural datasets were not available [70]. Basically, the choice is either to use an already existing dataset that might not fit the needs of the target domain, or to create a synthetic dataset that satisfies the required properties [98]. Although simulated datasets cannot fully represent a real life system and may be biased to certain algorithms or settings, they are still advantageous for testing the system in the design phase until real datasets have been collected [70].

There are various ways to conduct an offline experiment, either by conducting a cross-validation [194] or by using time-stamps to simulate the running system [227]. Evaluation frameworks exist to sup-

---

<sup>4</sup> <http://grouplens.org/datasets/movielens/>, retrieved 18.04.2014

<sup>5</sup> <http://grouplens.org/datasets/hetrec-2011/>, retrieved 18.04.2014



**Figure 3.4:** An Example of a 3-Fold Cross-Validation, adapted from [194]

port offline experiments on datasets, for example for the evaluation of folksonomy-based recommender systems [68] or for the simulation testing of multi-criteria recommender systems [154].

### Cross-Validation

This is a statistical evaluation method that splits the given dataset into two partitions. One partition is the so called *training set* used to train the recommender algorithm and to generate the recommendations. The second partition is the *test set* used to validate, test or estimate the performance of the recommender algorithm. There are different kinds of cross-validation methods [194]:

- **Re-substitution Validation** is a very simple method where the training and test set are the same. The whole dataset is used for training the algorithm and the same dataset is then used to test the algorithm. Over-fitting is a problem as the algorithm might only perform well on this dataset.
- **Hold-Out Validation** splits the dataset into two independent, non-overlapping partitions. This prevents over-fitting, however the choice of test and training partitions could adversely affect the measured performance of the algorithm. The available training and test data is also greatly reduced.
- **K-Fold Cross-Validation** splits the dataset into  $k$  equal partitions (called folds).  $K$  Validation iterations can now be performed using these  $k$  folds. In each validation run, one fold of data is used as test set and the remaining  $k - 1$  folds are used as training set.  $K$  validation estimates are obtained. Figure 3.4 shows an example of a 3-fold cross-validation.
- **Leave-One-Out Cross-Validation** is a special case of  $k$ -fold cross-validation, where  $k = n$  and  $n$  is equal to the number of instances of data. This means nearly all the data is divided into folds and one fold is left out as test set. Results are very un-biased but this method costs a lot of time to compute.
- **Repeated K-Fold Cross-Validation** runs a  $k$ -fold cross-validation several times. For each repetition, a new fold is randomly selected as a test set. A 10-fold cross-validation is commonly performed as it offers a good balance between the cost of repeated runs and the number of validation estimates.

### Simulations using Time-Stamps

In datasets where time-stamps are available, time could be used to simulate how recommendations could have been made at the time the system was running [98]. There are several possibilities to do this [227].

- One approach is to start at the earliest time-stamp in the dataset and to step through in temporal order. For each user encountered, the user's future interactions are hidden and the recommender system attempts to predict them.

- A second approach is to select users randomly from the dataset. For each of these users, a point in time just prior to a user action is randomly chosen. After this chosen time-stamp, all other user interactions are hidden and the recommender system tries to recommend items to this user.
- Another approach is to randomly sample a set of users and to select a random time-stamp. All items after this time-stamp are hidden for each test user. Thereby, a scenario is simulated where the recommender system is created at the time of the time-stamp and recommendations do not consider any new data after the time-stamp.
- A last approach is to select a time-stamp for each user and then keep the items of the user after that time-stamp hidden. The time-stamps can be varied across all test users. Therefore, the sequence in which items are selected is important and not the selected time-stamps.

Offline experiments are relatively fast to conduct when compared to the other two evaluation methodologies. Experiments are easily repeatable and many parameter settings and algorithm variations can be tested on different datasets. It is also possible to conduct rather large evaluations, having only the costs of running the computations over the datasets. The evaluation results are however objective simulations, thus it is difficult to measure user-centric metrics like user satisfaction. Furthermore, offline experiments are limited to the historical interactions of the users collected in the dataset, therefore no new, unknown resources (not interacted with in the past) can be evaluated [167], this is also known as the *Incompleteness Problem* [49].

### Incompleteness Problem

The incompleteness problem [49] arises from the assumption that all items the user interacted with in the dataset will always be relevant to this user at any time, no matter what the user is presently working on. This assumption cannot always be true. In addition, the assumption that all items the user had not interacted with in the dataset are not relevant to the user. This also cannot always be true. Therefore offline experiments based on cross-validation are limited in this regard and alternative evaluation methods are needed to complement the evaluation results from offline experiments. This motivates user-centric evaluation approaches, as these methods can evaluate the relevance of an item to a user without being dependent on the historical interactions in a dataset. In addition, user-oriented evaluation can measure user-centric metrics that offline experiments cannot measure.

---

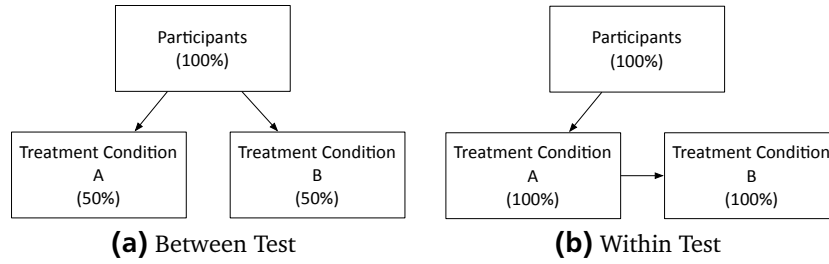
## User Studies

---

As it is important to not only measure the accuracy but also the user's satisfaction with a recommender system, it is necessary to take so called user-centric measures such as novelty or diversity into account by performing user experiments. A user study can be described as a scientific method used to find out how a recommender system influences a user's experience, perception and interactions with a system [130]. User studies are often performed by psychologists in the human sciences [87].

A user experiment is basically executed by asking users of a recommender system to perform several tasks in a controlled environment for a short period of time. The interaction behaviour of the users with the recommender system is then observed and recorded, for example the time taken to complete the task, or the quality of the results of the task [227]. Most of the time, users are asked questions before, during or after the experiment. Such questions, often prepared as a questionnaire or asked in an interview, can help to capture aspects that cannot be directly observed otherwise, such as how the user feels about using the system or taking part in the experiment [227]. Questions should be formulated simply, using neutral wording so as not to be suggestive. The participants in a user experiment should generally be unbiased users of the system and need to be selected randomly from a representative population sample. The sample size needs to be sufficiently large in order to be able to attain results that are statistically significant [130]. Research questions need to be formed into hypotheses. A hypothesis needs to be precisely defined and should be measurable [87]. Results from an experiment either validate or reject a postulated hypothesis.





**Figure 3.5:** User Experiments: Between and Within Tests

Dependent variables, also known as response or measured variables, are the aspects to be measured during a user experiment. Independent variables, also known as predictor or manipulated variables represent the different treatment conditions participants could be assigned to. In user experiments, several aspects could be manipulated: the recommender system, the recommender algorithm, the presentation of the recommendations or the interaction with the recommender system [130]. Commonly, several candidate recommender systems or algorithms are compared and tested against a reasonable alternative or so called baseline. Each of these candidates must be tested with the same experimental setting and with the same questionnaire in order to be comparable. Each of these candidate systems or algorithms could be a treatment condition. Participants are randomly assigned to treatment conditions so as to avoid participant variation or biases [227]. There are basically two ways of assigning participants to treatment conditions in a user experiment [130]:

- **Between-test or A/B test:** As shown in Figure 3.5a, half of the participants are assigned to treatment condition A, and the other half to treatment condition B. The manipulation thus remains hidden from the participants, however quite a lot of participants are needed for this type of experimental design [130].
- **Within-test:** All participants are assigned both treatment conditions A and B as shown in Figure 3.5b. The participants are aware of the experimental manipulations. The assignment order of the treatment conditions could however be randomized i.e. either A then B or B then A to avoid a bias in the order of execution of the treatment conditions [130].

There have been several user-centric evaluation frameworks proposed to evaluate recommender systems [131, 166, 190]. These frameworks guide the design and execution of user experiments. User studies are however often restricted to a small set of topics and only a relatively small set of tasks can be accomplished. It also costs a considerable amount of time and effort to plan and execute. In addition, recruiting a sufficient number of diverse and representative participants is a challenge. As a result, only a limited amount of algorithm variations and parameter settings can be tested, unlike in offline experiments. Results from user studies could be biased as the participants often know they are taking part in an experiment and if they guess at the hypothesis being tested, they may unconsciously give evidence to support it [227].

### User-Centric Evaluation Metrics

According to the requirements of recommender systems for TEL presented in Section 3.1.3, it is necessary to recommend personalized learning resources that are novel and serendipitous to the learner. Serendipity means making a relevant but new, unexpected and surprising recommendation, which the learner would probably not have found otherwise. Serendipitous recommendations are inherently novel [98]. Serendipity extends the concept of novelty by helping users find interesting items that the users would not have found on their own [61]. In addition, serendipity provides a way to diversify recommendations and diversity is often a needed characteristic of a recommender system [148].

Serendipity as a metric is difficult to measure [167], but the novel and diverse aspects could be measured in a user study. **Diversity** can be described as receiving recommendations, that are not similar

---

to other recommendations received [227]. **Novelty** can be described as the amount of relevant recommended resources that are unknown to the user [21]. Novelty can also be described as a measure of how many interesting, new and unknown recommendations a user receives from a recommender system [190].

---

### Real life testing

---

In real life testing, also known as online evaluation [227], real users use the system under normal conditions over a long period of time [159]. This might be as a field study where a large community of users is observed while using the system under realistic conditions or as a pilot study where a mature system is deployed in its real life setting. As a side benefit, large datasets can be collected for offline experiments. With real life testing, most user-centric metrics can be effectively evaluated such as user experience or user satisfaction [98]. Real life testing provides very reliable results and both qualitative and quantitative results can be measured [227]. As real life testing takes place over long periods of time, many influencing factors cannot be fully controlled nor excluded, therefore some criteria cannot be evaluated, for example, the learning success of the learners. It is also a challenge to recruit sufficient participants who remain committed for a long enough period of time. Moreover, there is a high risk to real life testing as users are easily frustrated when the system's performance is lower than expected [227]. As real life testing is accompanied by a lot of risks and costs, it is neither possible to test multiple variations of the system, nor can tests be repeated as often as by offline experiments.

---

### 3.2.2 A Survey of Evaluation Methods for TEL Recommender Systems

---

Recommender systems play an increasingly important role in Technology Enhanced Learning (TEL) and over the years, more and more recommender algorithms have been developed. With this growing importance comes a corresponding need to evaluate these recommender algorithms for TEL. Often, evaluation methods for evaluating recommender algorithms in other domains such as in e-commerce are reused in TEL [157]. However, recommender systems for TEL have differing goals in comparison to recommender systems in other domains, as they need to consider specific requirements, as explained in Section 3.1.3, in order to support the learning process [45, 157]. These differing goals also demand evaluation methods which differ from those used in other domains. As a result, there is a need to investigate these particular requirements and goals for TEL recommender systems and how they can best be evaluated.

The following survey takes a first step in this direction by giving a representative overview of how TEL recommender algorithms have been evaluated up till now. Publications on TEL recommender systems from relevant conferences, workshops and journals were analysed, focusing on educational technology between January 2006 and December 2013. Three journals were analysed, namely Elsevier Computer & Education, Journal of Computer Assisted Learning (JCAL) and IEEE Transactions on Learning Technologies (IEEE TLT). In addition, three series of conferences were investigated; two in the educational technology area, namely IEEE International Conference on Advanced Learning Technologies (ICALT), and European Conference on Technology Enhanced Learning (EC-TEL), as well as a leading conference on recommender systems, namely ACM Conference on Recommender Systems (RecSys). Finally, a series of specialized workshops were analysed: Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL) and Workshop on Social Information Retrieval for Technology Enhanced Learning Exchange (SIRTEL). Overall 107 publications were selected as being relevant to recommender systems for TEL as shown in Table A.1 in Appendix A.

---

### Classification Criteria for Survey

---

The search terms *recommend*, *recommender*, *recommendation*, *suggest* were used in a full-text search to identify potentially relevant publications. Additionally, publication titles, keywords and abstracts were

---

perused and candidate publications selected. The classification for the survey was then done manually by reading through the selected publications. It was more difficult identifying relevant publications in earlier years as the terms *recommender system* or *recommendation* were not yet widely established. Not all venues had relevant publications for each year and some venues had peaks, for example the specialized workshop on TEL recommender systems RecSysTEL took place only once every two years - in 2010 and in 2012. The selected publications were reviewed and classified according to the following classification criteria.

### Type of Recommender Algorithm

These four kinds of recommender systems were used as classification criteria in the survey: content-based, collaborative filtering, knowledge-based, and hybrid recommender systems [110] as described in Section 3.1.

### Type of Recommended Items

Recommender systems in TEL recommend a wide variety of items [45] such as learning objects in a repository, resources on the Web, test or examination items, lecture notes, complete courses or even learning partners. For the survey, the following classification was made: resources, users, metadata, activities, sequences, courses and feedback.

- **Resources:** This comprises all kinds of digital resources considered relevant for learning.
- **Users:** This category comprises users with similar interests or learning goals, who could be peer learners or teachers.
- **Metadata:** This comprises any semantic information suggested by the recommender system used to describe other items such as resources or courses.
- **Activities:** All tasks, activities and learning goals of the learner are considered in this category.
- **Sequences:** This comprises broadly the learning paths through the learning resources.
- **Courses:** Complete learning courses such as reading lessons or language learning courses, as well as a bundle of structured learning resources were considered as courses.
- **Feedback:** This widely comprises feedback from the user and interactions between the user and the system.

### Type of Evaluation Methodology

The types of evaluation methodologies applied for the evaluation of the recommender system or recommender algorithm, as described in Section 3.2, were classified in these four categories:

- **User Study:** This included mainly user experiments, expert interviews, controlled lab experiments and online surveys.
- **Real Life Testing:** The complete recommender system was used by real users in a real life setting for a long period of time (generally over a few months); also called a field study.
- **Offline Experiment:** This comprised offline experiments on historical and synthetic datasets, manually tagged datasets as well as simulation experiments.
- **No Evaluation:** This means no evaluation results were documented in the publication.

### Focus of Evaluation

The evaluation was designed for and applied to either evaluating the entire recommender system or only the recommender algorithm, or both:

- **Recommender Algorithm:** Only the recommender algorithm was evaluated.
- **Recommender System:** The evaluation considered the complete recommender system, usually as part of a larger learning platform.
- **Recommender Algorithm and Recommender System:** Here the focus of the evaluation was on the complete recommender system as well as on the recommender algorithm.

---

## Effects Measured by Evaluation

The goal of an evaluation is to measure a certain effect or property of the recommender system. The different effects measured in the survey were classified according to the following categories:

- **Accuracy:** This category contains a varied number of evaluation goals, all with the general aim of measuring the precision or quality of the recommender algorithm. Some of the identified goals aim to measure the accuracy of the predictions, to determine the performance improvements of the algorithms, to measure the perceived relevance of recommendations regarding certain learning tasks, the representativeness and quality of the recommendations, the correctness of the approach, the validity of the recommender algorithm or underlying domain model and the overall effectiveness and performance of the system.
- **Correlations:** In this category, the correlations between user activities and measured effects are investigated. Generally, the co-occurrences and correlations between different activities found in the collected dataset are analyzed. For example, the correlation between the usage logs of a learner and the performance of the learner in an examination could be investigated.
- **Knowledge Levels:** This comprises measuring the learner's individual knowledge level or expertise level in a particular topic and comparing the learner's intellectual abilities.
- **Learning Motivation:** In this category, the improvement in learning motivation of the learner is measured.
- **Learning Performance and Learning Achievement:** This category comprises mainly the perceived effectiveness of learning using the recommender system. This covers the learner's achievements and scores in tests, as well as the speed with which a learning activity is executed. Other indicators are the learner's reading or posting frequency.
- **Task Support:** This category comprises diverse means of support for the learner's current tasks, including the efficiency or speed of a recommender algorithm in generating recommendations at runtime.
- **User's Feedback and Usability:** This comprises the user's satisfaction with the system and the user's perceived usefulness of the recommendations.

---

## Results of Survey

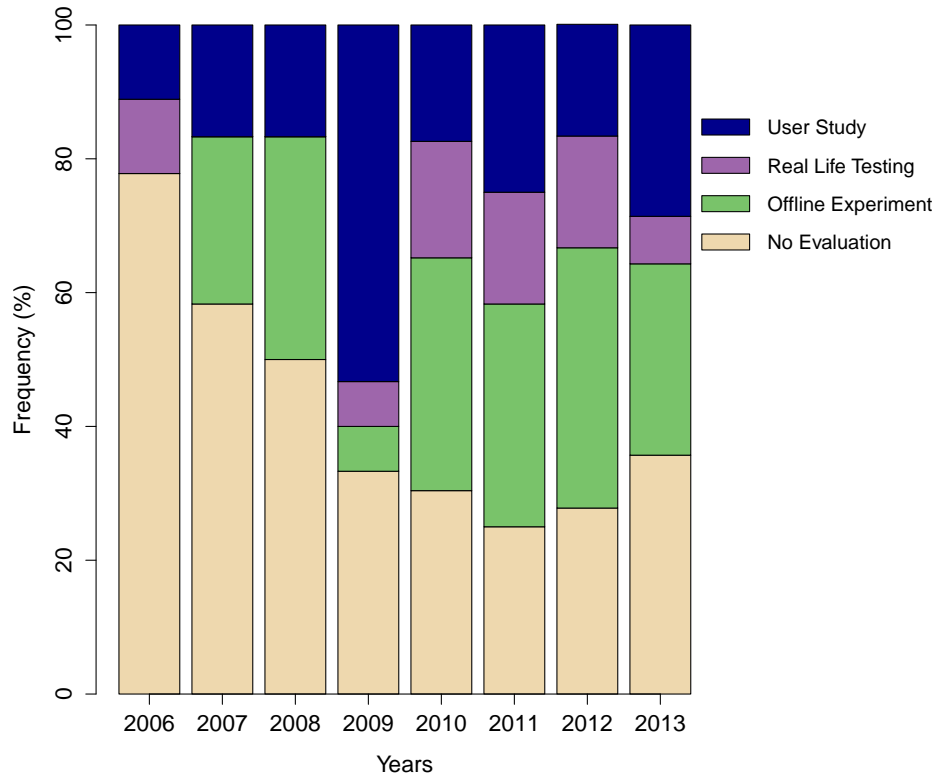
---

The 107 selected publications listed in Table A.1 in Appendix A were categorised according to the classification criteria explained above. The results of the survey are presented and discussed in the following sections.

### Type of Evaluation Methodology

The evaluation methods applied in the publications were classified as user study (27 publications), offline experiment (31 publications), real life testing (12 publications) or no evaluation (45 publications). Some short papers did propose evaluation scenarios but then referred to future work for evaluation results, e.g. [91, 140]. Figure 3.6 shows the percentage distribution of evaluation methods over the years. Some publications covered several evaluation methods. For example, an offline experiment and a complementary user study in [53] where the results of an offline experiment evaluating the accuracy of the recommender algorithm are compared with the user satisfaction evaluated in a user study. The findings confirm the claim that for recommender systems, the high accuracy measured by metrics such as precision or recall in offline experiments does not correlate to a high quality in user experience [190]. Another example can be found in [63], where the accuracy of an algorithm is first evaluated in an offline experiment and then two user studies follow to evaluate user perception and usability.

Some evaluations have real life tests followed by user studies or offline experiments. For example, a real life test is conducted in [252] to measure the performance of the recommender system and to generate a dataset, with a subsequent user study with experts to validate the recommendations. In [156], a



**Figure 3.6:** Evaluation Methodologies (Percentages per Year)

real life pilot test was run over 8 weeks in order to evaluate the teacher’s perceived usefulness and quality of the resources recommended as well as to collect a multi-attribute dataset for an offline evaluation. Other evaluations conduct a user study and afterwards analyse the data collected. For example, in [146], two user studies are complemented with a log data analysis, and similarly in [193] and [177] where the usage logs collected in a small preliminary user experiment are afterwards analysed in an offline experiment. A multi-staged evaluation was conducted in [232] where all three evaluation methodologies were covered. In the first stage, an offline experiment (a cross-validation) was executed to measure the accuracy (precision and recall) of the recommender algorithm. In the second stage, a user study (an expert questionnaire) was conducted with 4 experts evaluating the recommendations with regard to pedagogical aspects. The third stage was a real life testing to evaluate the user satisfaction with the system. Explicit feedback and click behaviour were monitored and log files analysed.

The survey shows that the evaluation of TEL recommender systems has become increasingly important over the years, with the number of publications without an evaluation having decreased substantially (from about 78% in 2006 to 36% in 2013). Overall, offline experiments were slightly more popular than user studies, although in 2006 and 2009 the focus had been more on user studies. The percentage of real life testing per year remained rather low, without any reported in 2007 and 2008.

### Offline Experiments

The survey revealed that the number of offline experiments remained steady over the years, with peaks in 2010 and 2012 due to the dedicated workshop on recommender systems in TEL (RecSysTEL). Overall, offline experiments were as popular as user studies, although in some years such as 2009, the focus has been more on user studies. Most of the offline experiments were executed on historical datasets. Historical TEL datasets that fulfil all requirements for an evaluation are however hard to find [156]. An early solution to this problem was to create synthetic datasets such as used in [70] and [155]. Since then, attempts have been made to generate datasets for TEL. For example in [156] where a dataset is generated from a real life testing evaluation. The dataTEL initiative extensively looked into issues regarding

---

collecting sharable datasets for TEL and proposed some guidelines on how best to accomplish this [74]. A few offline experiments compared evaluation results to a baseline [19, 109, 222, 210]. An overview is given in Table A.4 in Appendix A.

### **Real Life Testing**

According to the publications surveyed, real life testing often takes place with real users using a prototype implementation of the recommender system, sometimes as part of a project deliverable [26, 246]. Few publications had a real life testing evaluation; however in recent years (2010 - 2012) there has been a slight improvement, see Table A.5 in Appendix A for more details. In most real life tests, the entire recommender system was evaluated including the recommender algorithm. The testing periods lasted generally between 4 to 32 weeks and the number of participants were on average higher than for user studies, the highest amount being 1763 participants over a 32 week period [229] and the lowest 6 participants over an 8 week period [145]. Some publications stated specific evaluation goals and reported concrete evaluation results such as in [127], whilst in others, the reports on real life evaluations were very vague, no concrete results were stated, neither the number of participants, nor the duration of the testing were mentioned [125, 246].

### **User Studies**

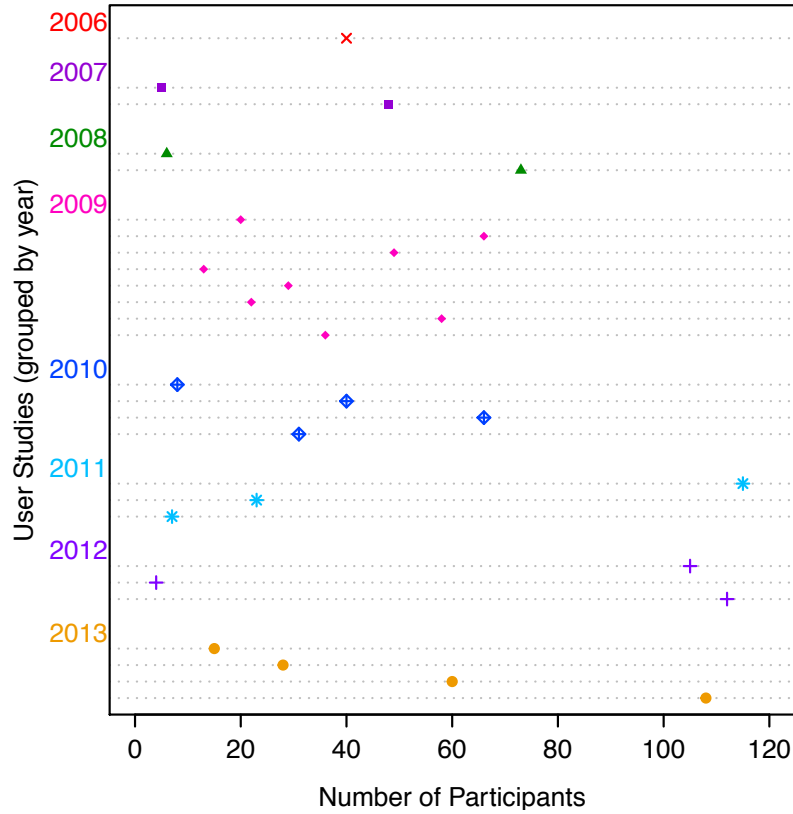
The number of user studies conducted have been quite steady over the years as can be seen in Figure 3.6. An overview of the diverse methods used in the user studies are shown in Table A.6 in Appendix A. Questionnaires are used most often in user studies. Sometimes experts are considered in the experiment, either by giving them a questionnaire [22] or by interviewing them. Some do not ask the participants direct questions but rather observe their interaction with the system. Others conduct pre-tests to measure the learner's knowledge before the experiment and a post-test afterwards [106]. User experiments, on the one hand are often conducted as lab experiments, where the participants take part in the experiment in a controlled environment under surveillance. On the other hand, some experiments have been offered as online experiments, where the participants are not in a fixed place but rather log into the system from where they are. One experiment used a crowdsourcing platform to reach participants [119]. Figure 3.7 shows the distribution of the number of participants in user studies over the years. The number of participants in user studies ranges between 4 to 115. With a total of 27 user studies, on average 44 participants took part in a single user study. However, the median value was 36 as a few studies had a lot of participants such as 115 in [63]. When several unrelated user studies are reported in one publication, the average of the number of participants is taken. For example in [224] where two user studies were performed, one as an online experiment with 121 participants and the other as a lab experiment with 89 participants.

### **Focus of Evaluation**

The focus of the evaluation was classified according to what was evaluated. The findings of the survey are shown in Table A.7 in Appendix A and the distribution of the evaluation focus across evaluation methods is shown in Figure 3.8.

- **Recommender Algorithm:** According to the survey, recommender algorithms were evaluated more often than the entire recommender system. The recommender algorithm was often evaluated by performing offline experiments on historical datasets to evaluate the precision or recall.
- **Recommender System:** To evaluate the entire recommender system, the users were often asked questions regarding overall user satisfaction, usability, learning experience and perceived support during the learning process.
- **Both:** The evaluation of both the recommender algorithm and recommender system comprised the two approaches mentioned above, sometimes as two distinct phases of the evaluation. In only 8 publications was it explicitly mentioned that both the recommender system and the recommender algorithm were evaluated.





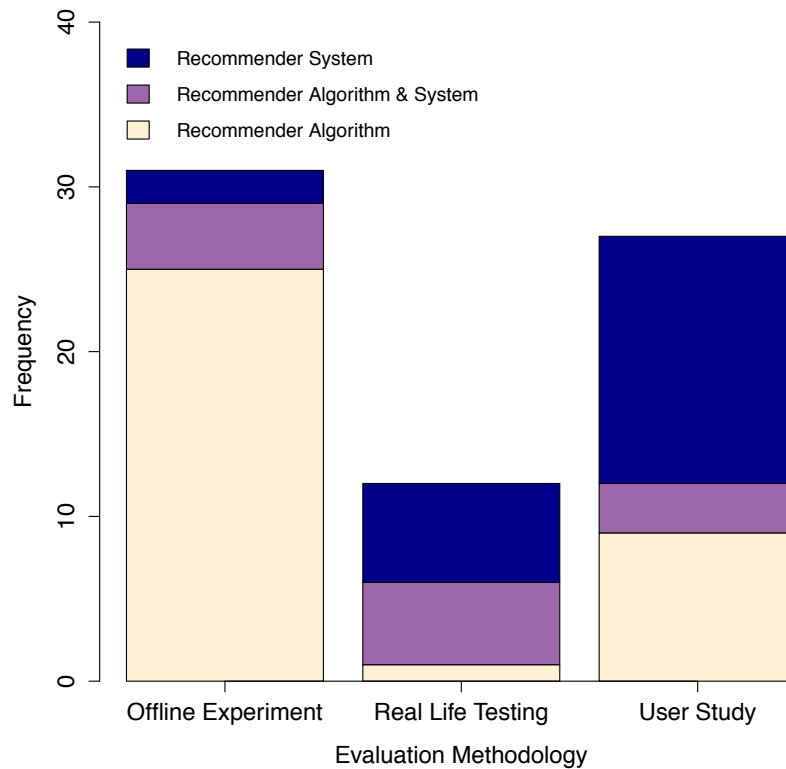
**Figure 3.7:** Number of Participants in User Studies over Time

Considering the evaluation methodologies, offline experiments focused mainly on evaluating the recommender algorithm, whereas real life testing and user studies concentrated on evaluating the entire recommender system. Rarely was the evaluation focus on both the recommender system and the recommender algorithm.

### Effects Measured

An overview of the effects measured by the evaluations in the survey are shown in Table A.8 in Appendix A. Accuracy and user feedback are the most common effects measured followed by learning performance, task support, correlations, knowledge levels and learning motivation. Figure 3.9 shows that in recent years, the types of effects measured have increased in number and become more varied. Figure 3.10 shows the effects measured across the evaluation methodologies. User studies and real life testing cover all evaluation goals. Offline experiments do not cover knowledge level and learning motivation. Some publications reported multiple evaluation goals, for example [105, 106, 127, 114].

- **Accuracy:** Accuracy is the most predominant effect measured over the years and across all evaluation methodologies, although user feedback is also dominant in user studies and real life testing. Some common metrics used to measure accuracy were precision, recall, f-measure [53, 210], Mean Average Error (MAE) [156, 210], and Root-mean-square-error (RMSE) [138].
- **Correlations:** The correlations between the rank position of a learning goal and the frequency with which a learning activity was executed were measured [146]. Some metrics used were for example, Pearson Correlation [180], Kendall's Tau Correlation [216].
- **Knowledge Levels:** Knowledge levels were measured using self- and peer-assessments and applying Knowledge Indicating Events (KIE) [145].
- **Learning Motivation:** Learning motivation was measured using a learning motivation questionnaire such as the Motivated Strategies for Learning Questionnaire (MSLQ) [105].



**Figure 3.8:** Evaluation Focus across Evaluation Methodologies

- **Learning Performance and Learning Achievement:** The perceived effectiveness of learning using the recommender system was measured in [212]. The learner's achievements and scores in tests were measured in [106, 193]. It was also measured if students using a recommender system achieve better results than students receiving random or no recommendations [173]. The learner's reading frequency per post, the learner's replying and posting frequency were also measured [243], as well as the frequency with which a learning activity was executed [146]. Evaluation tools used were pre-knowledge tests [212], pre- and post-tests [173], assignment grades [193], usage log analysis [146, 193]. Learning styles [212, 237] were measured for example with the Felder-Soloman Index of Learning Styles (ILS) questionnaire [127].
- **Task Support:** Support was given to learners in order to achieve their current learning goals [95, 146] and to accomplish their learning or authoring tasks [239]. The effectiveness and applicability of the recommendations in supporting learning tasks were also measured [53]. In addition, the runtime efficiency of a recommender algorithm was measured in [127, 152, 156].
- **User's Feedback and Usability:** The user's satisfaction with the system [55, 114], the user's perceived usefulness of the recommendations [237], the benefits of the system with respect to learning and enjoyment [95] and the ease of use for both students and teachers [106] were measured. Evaluation tools utilized were the Technology Acceptance Model (TAM) [106], General Interest Structure Test (AIST) [223], interviews [114], observations [55, 213], recordings of the participants' interactions and think aloud protocols [239].

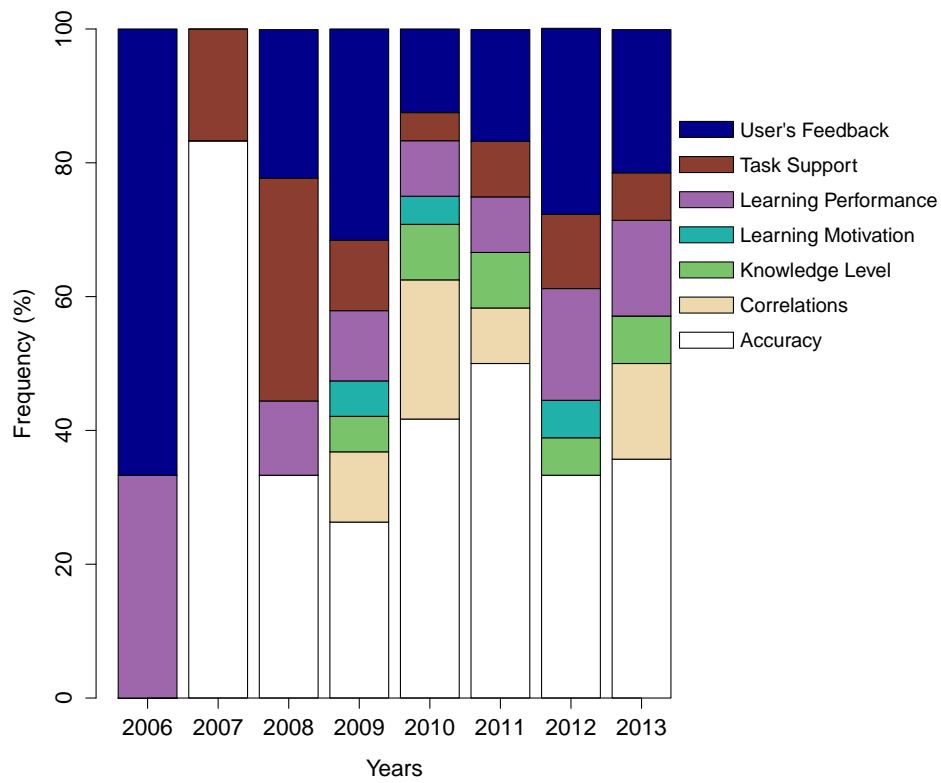
---

## Discussion of Survey Results

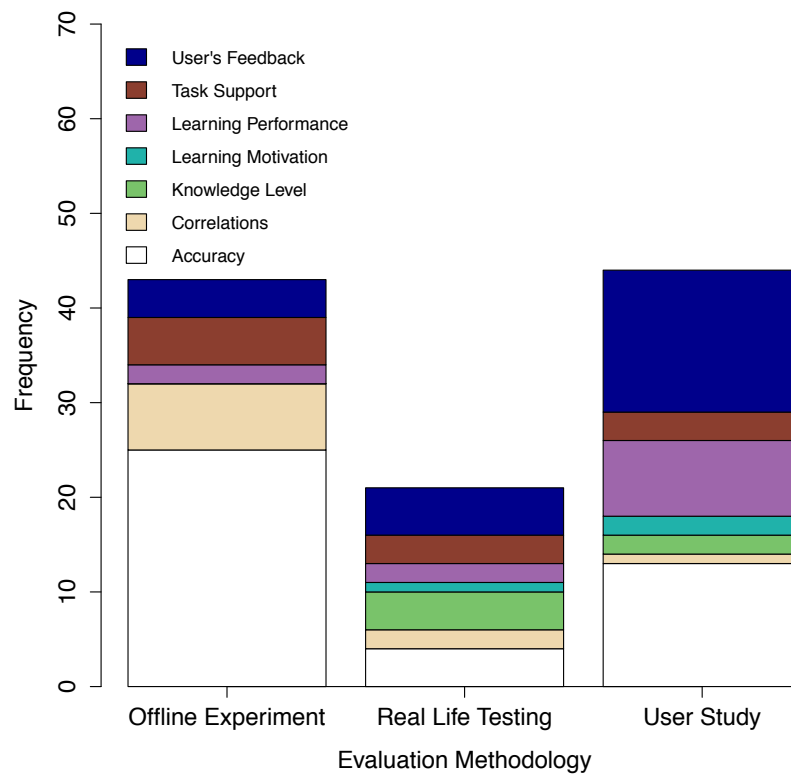
---

There are many challenges facing the evaluation of recommender systems for TEL [239]. According to the survey, few publications stated a clear evaluation focus. Evaluation goals were not always explained in detail and most of the time, the scope of the evaluation goals were much too wide. For example, the evaluation goal of measuring *learning performance* needs to be broken down into smaller, more concrete





**Figure 3.9:** Effects Measured (Percentages per Year)



**Figure 3.10:** Effects Measured by Evaluation Methodologies

---

goals. The evaluation of TEL recommender systems needs to focus more on the impact of the recommendations during learning [45]. As yet, little is known about the way users perceive recommendations and how they react to them [45]. More real life testing of TEL recommender systems over a longer period of time is needed, where the effect of the recommender system on learners, their acceptance and usage of the recommendations are measured [159].

In addition, few TEL specific evaluation goals are stated and measured. For example user-centric metrics like novelty or serendipity [167, 227] were not reported as explicit evaluation goals. Offline experiments cannot measure user-centric evaluation metrics and a high prediction accuracy does not always correlate with high user satisfaction [167, 190]. But according to the survey, the most often measured evaluation criteria are currently precision and accuracy [159]. There are however many other important properties that should be considered such as novelty and diversity [227]. User studies could complement offline experiments, but according to the survey, on average only about 40 participants take part in a user study. User studies are also rarely repeated in order to confirm results [227]. Therefore alternative approaches to evaluating recommender systems such as crowdsourcing [78, 119] as presented in Chapter 6 should be considered.

---

### 3.3 Summary and Research Goals

---

In this chapter, the fundamentals of recommender systems and their evaluation methodologies have been explained. In addition, related work on graph-based recommender systems for social tagging systems as well as graph-based recommender systems for TEL have been presented. In the following, the two main goals of this thesis are presented in more detail. The first goal is to provide personalized recommender approaches to support resource-based learning and the second goal is to provide alternative evaluation approaches for evaluating TEL recommender algorithms.

#### Personalized Recommender Approaches for RBL

The first goal is to provide personalized recommender approaches to support resource-based learning. In order to offer support to learners in a resource-based learning scenario, personalized recommender systems need to be developed that solve the information overload problem by recommending relevant learning resources to the learner. Recommender algorithms for TEL, and in particular for RBL, need to fulfil certain requirements and overcome certain restrictions. These requirements and restrictions determine the choice of recommender approaches that are most suitable for the RBL application scenario being considered in this thesis. The following points highlight the reasons for the choice of hybrid graph-based recommender approaches for the RBL application scenario.

- To support collaborative learning, the community information available in social tagging systems needs to be leveraged. It would be beneficial to generate personalized resource recommendations for a particular learner by considering the learner's tagging behaviour, the learner's current task or learning goal, the learner's and other learner's hierarchical activity structures in the community, as well as related learning resources, their tags and semantic tag types. Neighborhood-based collaborative filtering approaches are an appropriate choice of recommender approach as they are fundamentally based on utilizing community data and exploit the transitive relations between entities in the graph, thereby taking the learning resources, semantic tags and hierarchical activity structures of other learners in the community into consideration [61].
- Most TEL systems have very sparse data and this affects the kind of recommender techniques that can be applied [157]. Graph-based approaches overcome this limitation by exploiting the transitive relations between entities in a graph thereby reaching entities that are not directly connected [61]. Furthermore, graph-based approaches based on a random walk have the ability to reach entities in isolated sub-graphs by following the random surfer principle of jumping to a random entity in the graph. Incorporating additional semantic information also enhances the number of nodes and links in a graph, thereby providing additional connecting paths through the extended folksonomy [66].

- As presented in Section 3.1.3, a requirement for TEL recommender systems is to recommend resources beyond their similarity. Unlike content-based recommender approaches [148], graph-based approaches have the advantage of being able to exploit the transitive relationships between users, tags and resources to generate recommendations that the learner is not aware of. These recommendations thus have the potential of being novel, diverse and serendipitous to the learner [61].
- In order to provide learning resources relevant to a learner's current learning activity and knowledge level, recommender systems for TEL need to be context-aware [8]. The context of a learner can be determined by taking advantage of additional semantic information [157], for example as hybrid recommender systems that consider additional semantic information gained from various sources [45]. Additional semantic information found in social tagging systems could be exploited in order to recommend resources fitting the learner's learning context, current task or learning goal and also to alleviate certain challenges like concept drift and tag disambiguation [35].

Thus the focus of this thesis will be on collaborative filtering recommender approaches and in particular feature augmented hybrid graph-based recommender systems [46, 61]. In the following, graph-based approaches for social tagging systems, and in particular for TEL, are compared and analysed according to their suitability for the RBL application scenario.

The graph-based approaches for social tagging systems presented in Section 3.1.2 are compared in Table 3.2 according to the requirements for personalized recommender systems for TEL stated in Section 3.1.3 as well as the particularities of a resource-based learning application scenario as described in Chapter 2.

- **Personalized:** A recommender algorithm fitting these requirements should be personalized, meaning the recommendations generated are tailored to the needs of individual users. SocialPageRank and SocialSimRank are not personalized and thus do not satisfy this requirement.
- **Topic-Sensitive:** Recommender algorithms for RBL should be able to focus not only on a certain user, but also on a certain entity in the graph, thus making the recommendations topic-sensitive. SocialSimRank and SocialHITS are not topic-sensitive.
- **Items Recommended:** The type of recommendations made could be users, tags, resources or other items from additional sources in an extended folksonomy. However, for a resource-based learning scenario, learning resources are the most important item to be recommended. ContentFolkRank, LocalRank, ExpertRank and SocialSimRank do not focus on recommending resources.
- **Additional Semantic Information:** The additional semantic information considered by the recommender algorithm is important as the presented learning scenario has some particular semantic information that could be exploited as introduced in Chapter 2. Most of the algorithms leverage additional semantic information, but these are very specific to the learning scenario and learning environment considered. FolkRank, LocalRank, SocialPageRank, Personalized SocialPageRank and SocialHITS do not exploit additional semantic information. GFolkRank and GRank consider groups and group structures from GroupMe!, which represent a comparable concept to the pedagogical concept of activities and activity hierarchies described in Section 2.4. But groups are considered resources and can be assigned tags, furthermore, neither GFolkRank nor GRank consider the relationship between a user and a group. Neither is the structure formed by the groups of groups considered when giving weights to the edges between the groups. Category-based FolkRank exploits tag categories, which is a comparable concept to semantic tag types, but categories can be freely defined just as tags and as such are treated simply as additional tags and not used to influence the preference of certain paths through the graph. Area-based FolkRank exploits spatial tags and URI-based FolkRank URIs. ContentFolkRank considers the textual content of resources. FolkRank + Publication Metadata + Social Ties exploits publication metadata and social ties. ExpertRank considers tag clusters based on their usage frequency and the age of the tags. SocialSimRank considers the semantic similarity between a query and a tag.

Approaches	Personalized	Topic-Sensitive	Items Recommended	Additional Semantic Information
FolkRank	Yes	Yes	Users, Tags, Resources	None
GFolkRank	Yes	Yes	Users, Tags, Resources	Groups
Category-based FolkRank	Yes	Yes	Categories, Resources	Tag categories
Area-based FolkRank	Yes	Yes	Resources	Spatial tags determine new edge weights
URI-based FolkRank	Yes	Yes	URIs, Resources	URI (ontological concepts)
ContentFolkRank	Yes	Yes	Tags	Textual content of resources
FolkRank + Publication Metadata + Social Ties	Yes	Yes	Scientific publications	Publication metadata and social ties
LocalRank	Yes	Yes	Tags	None
SocialPageRank	No	Yes	Resources	None
Personalized SocialPageRank	Yes	Yes	Resources	None
ExpertRank	Yes	Yes	Experts	Tag clusters based on usage and age of tags
GRank	Yes	Yes	Resources	Groups
SocialSimRank	No	No	Tags	Semantic similarity between a query and a tag
SocialHITS	Yes	No	Resources	None

**Table 3.2:** Comparison of Graph-based recommender algorithms for Social Tagging Systems

As such, none of recommender algorithms in Table 3.2 explicitly consider semantic tag types nor hierarchical activity structures as described in the RBL application scenario in Section 2.4. Additionally, none of them consider exploiting the semantic relatedness between tags, nor the context-specific information gained from a folksonomy to create a learning context for the learner.

The TEL graph-based recommender approaches described in Section 3.1.3 are compared in Table 3.3. The learning environment the recommender system is applied to, the learning context of the learner, the items recommended and the additional semantic information considered by the recommender system are discussed.

- **Learning Environment:** The learning environment is important as it determines the aim of the recommender system. The 3A ranking algorithm and the Social Semantic Web FolkRank support personal learning environments, which have a comparable aim as RBL of supporting the learner in a collaborative learning environment. Random-Walk recommender system supports ubiquitous learning, Relation Based Importance Ranking supports formal learning, Usage-based Clustering of Learning Objects for Recommendation supports learning object repositories and TAK supports language learning. None of the approaches support the RBL scenario directly.
- **Learning Context:** Most of the approaches consider the learning context of the learner in different ways. The Random-Walk recommender system and Social Semantic Web FolkRank consider the learner's interests. TAK exploits the learner's previous knowledge. Usage-based Clustering of

---

Learning Objects for Recommendation and the Random-Walk recommender system consider the learner's learning goal. The 3A ranking algorithm has biased rankings towards the learner's context defined by nodes in the graph. But none consider the inherent context of a learner based on the strengths of the relations between entities in a folksonomy, nor the learning context derived from the semantic tag types of the tags of a learner.

- **Items Recommended:** Various types of items are recommended, including learning resources amongst other items. The 3A ranking algorithm, Random-Walk recommender system and Social Semantic Web FolkRank recommend users, TAK recommends tags, and the 3A ranking algorithm recommends group activities.
- **Additional Semantic Information:** All approaches consider some sort of additional semantic information. The 3A ranking algorithm, Random-Walk recommender system and Social Semantic Web FolkRank consider social relations and tags from other learners in the social network. Random-Walk recommender system considers feedback from the learner and Usage-based Clustering of Learning Objects for Recommendation exploits the user's interaction with learning objects. TAK exploits tag preferences and tag similarities, while Relation Based Importance Ranking defines transition probabilities of relations between objects.

None of the approaches consider the hierarchical learning activity structures of the learner as described in Section 2.4, nor do they consider the additional semantic information gained from semantic tag types, nor the semantic relatedness between tags.

The approaches analysed in Table 3.2 and in Table 3.3 do not satisfy all requirements of the RBL application scenario presented in Chapter 2 nor do they exploit the particular additional semantic information available. Thus, several concepts of hybrid graph-based recommender algorithms are presented in Chapter 4 that exploit additional semantic information gained from the learning context of a learner in a folksonomy, the semantic relatedness between tags, as well as the pedagogical concept of activities and hierarchical activity structures and semantic tag types as introduced in the RBL application scenario in Chapter 2 to recommend learning resources. *AScore* and *AINheritscore* focus on exploiting activities and activity hierarchies in order to provide personalized recommendations of learning resources to the activity the learner is currently working on [19]. *AspectScore* uses semantic tag types to recommend learning resources by giving preference to certain paths through the folksonomy determined by semantic tag types [203]. *InteliScore* utilizes the additional semantic information gained from the semantic relatedness between tags to form new paths through the folksonomy [203]. *VSScore* exploits the context gained from a folksonomy by creating a vector space with the dimensions representing the entities in the folksonomy based on graph-based rankings [204].

### Evaluation Approaches for Recommender Algorithms for RBL

The second goal is to provide alternative evaluation approaches for evaluating recommender algorithms for resource-based learning. To this aim, the existing evaluation methodologies presented in Section 3.1.3 are compared in Table 3.4 according to the requirements for TEL recommender systems. The evaluation methods of recommender systems for TEL are compared by considering the number of participants, the duration of the evaluation, the effort required to execute the evaluation and the number of variations possible. User centric metrics are also considered as well as the evaluation scope.

- **Participants:** The number of users that could take part in an evaluation is limited depending on the evaluation approach. Offline experiments can handle a lot of users as the number of users only influences the computation time. For user studies and real life testing, finding a sufficient number of willing participants is a challenge. Participants in a user study have to be recruited and motivated to take part in the experiment. According to the survey, on average, only about 40 participants take part in a user study. Finding and retaining participants for real life testing is particularly challenging as the users need to use the system earnestly for a long period of time in a real life setting.
- **Duration:** An experiment takes a certain amount of time to execute. For offline experiments, this is the computation time needed to run the experiment and is usually several hours depending on the

Approaches	Learning Environment	Learner's Context	Items Recommended	Additional Semantic Information
3A ranking algorithm	Personal learning environments	Biased rankings towards a learner's context defined by nodes in the graph	Actors, assets, researchers, work packages, group activities	Social relations and relations between resources, trust, authorship, rating and topic information
Random-Walk recommender system	Ubiquitous Learning	Learner's current interests and learning goals	Learning materials, tutors, other learners	Learner's social tags from external social networks, feedback from learner
Social Semantic Web FolkRank	Personal learning environments	Learner's interests from tag-based attention profile	Learning Resources and users	Most often used tags of learner and friends from social networks
Relation Based Importance Ranking	Formal learning at a university	None	Learning Resources	Transition probabilities
Usage-based Clustering of Learning Objects for Recommendation	Learning Object Repository	Learner's current learning goal and learning context based on usage of learning objects	Learning Objects	User interactions with learning objects, usage frequency
Tag-based Prior Knowledge Recommendation System (TAK)	Support Language Learning	Prior knowledge of learner	Tags, Articles, Key sentences	Tag preference, position of tag in the article, tag similarity

**Table 3.3:** Comparison of TEL Graph-based recommender algorithms



size of the dataset and speed of the algorithm. User studies usually require several days as it is not always possible to get enough participants to take part in the study on the same day. Additionally, there may be constraints to the number of available resources (rooms, PCs, licenses) if the user study is to be performed in a laboratory. Real life testing requires a longer time to execute as the users need enough time to experience the system, discover all functionality and generate sufficient data. Usually several weeks or even months are needed.

- **Effort:** This refers to the amount of time, energy and material costs needed to prepare and execute an experiment as well as the time needed afterwards to analyse the results. In comparison to the other two methodologies, offline experiments require a low effort, especially taking into account the fact that they can be easily repeated with different variations making it less expensive the more often an experiment is run. User studies require more effort to plan as participants usually need to be present in the laboratory and are only there for a limited amount of time. While they are present, everything needed for the experiment must be ready. Repeating user studies with different variations does not make it less expensive. Real life testing requires a lot of effort as real users need to use a mature system for a long period of time under real conditions. Deploying and maintaining a system in a real life setting is expensive as it requires sufficient resources such as enough computational capacity, adequate user support or even user training.
- **Variations:** Evaluations can be repeated in different variations by setting different parameters or adjusting certain properties. These variations lead to multiple evaluation scenarios. Offline experiments are very easy to evaluate considering many different scenarios. Different settings and parameters are simply defined for the various cases to be evaluated. For user studies, if several variations are to be evaluated, then the number of participants need to be shared out to the different treatment conditions. This reduces the number of participants per variation. If the same participants should evaluate several variations, then other issues arise, such as biases as the user gains more knowledge of the system [227]. For real life testing, it is much more expensive to have many variations of a system running. A possibility is to switch algorithms or user interfaces or other aspects during the testing. Then several variations could be run, however this limits the amount of time available to test a single variation [227]. Additionally, we have the same problem as with user studies but even amplified, as the participants get to know the system over time and any changes would be noticed and eventually affect the evaluation results [167].
- **User Centric:** User centric metrics are for example user experience and expectations [167], user's trust [227], novelty [227], serendipity [167] and user satisfaction [167]. These metrics are more easily measured by asking the users themselves [167], although some approaches to measure indications of this in offline experiments have been looked into, for example novelty and serendipity [227]. However, others like user's trust in a system are just not possible with an offline experiment [227].
- **Scope:** The scope of an evaluation refers to the number of tasks that an evaluation method can cover. An evaluation methodology may not be able to cater to all evaluation scenarios possible. Offline experiments are limited to the historical interactions recorded in the datasets. Recommendations that were not *interacted* with in the history of a user are considered not relevant. This might not be necessarily true [167], therefore the evaluation scope of offline experiments is limited due to the incompleteness problem [49] as explained in Section 3.2.1. User studies and real life testing do not face this problem. User studies cover a wide scope of evaluation questions [227], however these are very subjective judgements. Real life testing covers also a wide scope of evaluation tasks and only this method allows the user to be observed in a real life setting.

The existing evaluation methodologies analysed in Table 3.4 do not satisfy all these requirements. According to the results from the survey in Section 3.2.2, it seems that the effort and duration of user studies and real life testing deters many researchers from evaluating recommender algorithms with these methods. Repeating experiments is expensive and thus only few variations are executed. Offline experiments in comparison are relatively fast, take less effort and can be repeated easily with multiple variations.

Evaluation Methodology	Participants	Duration	Effort	Variations	User Centric	Scope
Offline Experiments	Many	Hours	Low	Multiple	No	Limited, incompleteness problem
User Studies	Limited	Days	High	Few	Yes	Very wide
Real Life Testing	Limited	Weeks and months	Very high	Very few	Yes	Very wide

**Table 3.4:** Comparison of Evaluation Methodologies for Recommender Systems

Considering the TEL evaluation scenario however, datasets are rare [74] and those available may not have the specific semantic information needed for the particular evaluation scenario as learning systems are very tailored to the particular learning scenario. Thus there remains a need for new evaluation methods to evaluate recommender algorithms for TEL, where a sufficient number of participants takes part in the evaluation, the duration and effort of executing the evaluation is acceptable and multiple variations are possible at an acceptable cost. In addition, it should be possible to evaluate user-centric metrics, the evaluation scope should be wide enough and the incompleteness problem resolved.

In Chapter 6, an evaluation concept based on crowdsourcing is presented that fulfils these requirements [78]. Crowdsourcing has the advantages of offline experiments in that experiments can be executed very fast and repeated with less effort, thereby giving access to sufficient willing users. Crowdsourcing also has similar advantages as user experiments in being able to evaluate user-centric metrics. A repeated proof-of-concept evaluation experiment, where a graph-based hybrid recommender algorithm proposed in Chapter 4 is tested for relevance, novelty and diversity, shows that crowdsourcing can be applied as an effective evaluation method to evaluate TEL recommender algorithms [78, 79].



---

## 4 Hybrid Graph-based Recommender Approaches for TEL

---

In this chapter, several concepts of hybrid graph-based recommender systems are presented. *AScore* and *AlInheritScore* [19] are presented as approaches for recommending learning resources by exploiting the additional semantic information gained from activities and activity hierarchies. *AspectScore* [203] is also described, explaining how the semantic tagging structure can be used to enhance the potential of graph-based recommendations by introducing a weighting concept based on semantic tag types [14]. The algorithm *InteliScore* is proposed to overcome limitations and drawbacks of graph-based recommender algorithms in folksonomies by incorporating semantic information to disambiguate tags. *VSScore* is also presented that exploits the context-specific information gained from folksonomies. In addition, a concept for a personalized recommender system for resource-based learning is presented. The concept is described using CROKODIL (introduced in Chapter 2) as a concrete implemented resource-based learning application scenario. The implementation and integration of hybrid graph-based recommender algorithms into CROKODIL is also described.

---

### 4.1 Hybrid Approaches Exploiting Activities and Activity Hierarchies

---

As a contribution to this thesis, the additional semantic information gained from the pedagogical concept of activity hierarchies as introduced in Chapter 2 will be considered in an extended folksonomy for two hybrid recommender approaches: **AScore** and **AlInheritscore** [19]. Activities aim to support learners during their learning process by organizing their tasks or learning goals. Activities are created describing learning goals or tasks to be accomplished by a learner or group of learners. Activities are organized in a hierarchical structure, where sub-activities specify sub-goals or sub-tasks. Relevant learning resources needed to achieve these learning goals or to solve these tasks are attached to these activities. Thus recommending learning resources from relevant activities in the hierarchical activity structure would provide the learner with learning resources relevant to the learning activity the learner is working on.

Both recommender approaches are based on the CROKODIL extended folksonomy  $F_C$  as defined in Section 3.1.2, where activities are included as new entities in the folksonomy. The activity hierarchies are depicted as new relationships between activities ( $A, <$ ), users working on activities are said to belong to activities  $Y_U \subseteq U \times A$  and users can assign resources to activities  $Y_A \subseteq U \times A \times R$  in an activity assignment. Details of the two recommender approaches *AScore* and *AlInheritScore* are described in the following sections.

---

#### 4.1.1 AScore

---

*AScore* is based on the graph-based ranking algorithm *GFolkRank* [1] as described in Section 3.1.2. *AScore* treats activities analogous to how *GFolkRank* treats groups, but with some differences as activities are not groups of resources and activity hierarchies are similar to but are not groups of groups.

- *AScore* introduces activity nodes to the folksonomy graph.
- Edges representing activity assignments between activity nodes and resource nodes are included.
- Edges between users and activities are introduced.
- Edges between the activities depicting the activity hierarchy are included.
- *AScore* considers the activity hierarchies when determining the weights of the edges.
- *AScore* does not consider activities to be resources and thus activities cannot be assigned tags.

The newly added activities and links in the extended folksonomy graph are traversed by the intelligent surfer model (explained in Appendix D) to provide personalized recommendations of learning

resources relevant to the activity the learner is currently working on. The following steps detail the AScore algorithm.

- As in Section 3.1.2, let  $G_F = (V_F, E_F)$  be the folksonomy graph of the not yet extended folksonomy  $F$ .
- Let  $G_C = (V_C, E_C)$  be the folksonomy graph of the extended folksonomy  $F_C$ .
- AScore extends the folksonomy graph  $G_C$  with new activity nodes in Equation 4.1 and new edges representing activity assignments in Equation 4.2.

$$V_C = V_F \cup A \quad (4.1)$$

$E_A$  comprises all activity assignments where a user  $u$  added a resource  $r$  to an activity  $a$ .

$$E_A = \{\{u, a\}, \{a, r\}, \{u, r\} \mid u \in U, r \in R, a \in A, (u, a, r) \in Y_A\} \quad (4.2)$$

- In addition, AScore includes edges between activities to form an activity hierarchy as shown in Equation 4.3. Where  $a_{sub}$  is a sub-activity and  $a_{super}$  is a super-activity in the graph.

$$E_H = \{\{a_{sub}, a_{super}\} \mid a_{sub}, a_{super} \in A, a_{sub} < a_{super}\} \quad (4.3)$$

- Activity membership relations between users and activities are also included in the folksonomy graph as shown in Equation 4.4.  $E_U$  comprises all assignments of a user  $u$  to an activity  $a$ .

$$E_U = \{\{u, a\} \mid u \in U, a \in A, (u, a) \in Y_U\} \quad (4.4)$$

- Finally, the edges of the graph  $E_C$  are extended as shown in Equation 4.5 to include all activity assignment edges  $E_A$ , all activity membership edges  $E_U$  and all activity hierarchy edges  $E_H$ .

$$E_C = E_F \cup E_A \cup E_U \cup E_H \quad (4.5)$$

- The newly introduced edges are given weights as depicted in Figure 4.1. The edges in  $E_A$  are given all the same weight  $activityAssign(u, r, a)$  in Equation 4.6 because, similar to GFolkRank, a resource can only be added once to an activity. Attaching additional semantic information to a resource (like assigning it to a group in GroupMe! or to an activity in CROKODIL) is seen as more valuable than simply tagging it [1], therefore  $activityAssign(u, r, a)$  is assigned the maximum number of users who assigned tag  $t$  to resource  $r$  in Equation 4.6.

$$w(u, a) = w(a, r) = w(u, r) = activityAssign(u, r, a) \quad (4.6)$$

where  $activityAssign(u, r, a) = \max(|U_{t,r}|)$

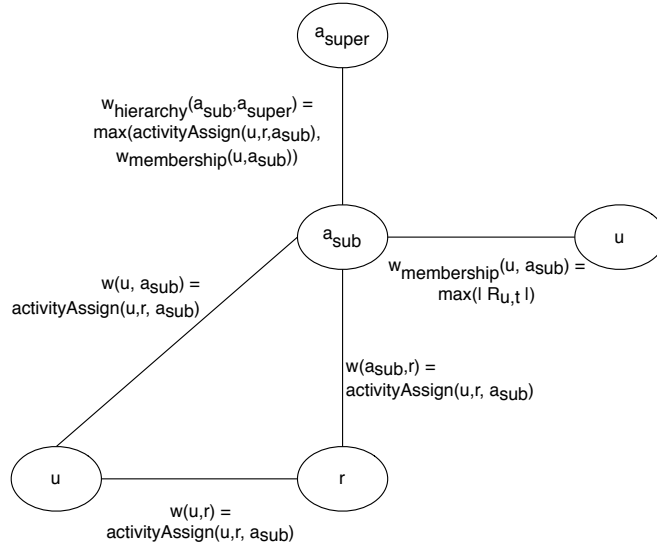
- Similarly, the edges between a user  $u$  and an activity  $a$  are given the weight  $w_{Membership}$  in Equation 4.7 which is the maximum number of resources assigned with tag  $t$  by user  $u$ , who is working on activity  $a$ .

$$w_{Membership}(u, a) = \max(|R_{u,t}|) \quad (4.7)$$

- Furthermore, AScore considers the hierarchical activity structure when determining the weights of the newly introduced edges. The edges between activities of the same hierarchy are given the weight  $w_{Hierarchy}$ . These edges are seen to be at least as strong as the connections between an activity and other nodes in the graph, therefore in Equation 4.8, the maximum weight is assigned.

$$w_{Hierarchy}(a_{sub}, a_{super}) = \max(activityAssign(u, r, a_{sub}), w_{Membership}(u, a_{sub})) \quad (4.8)$$

After the folksonomy graph  $G_C$  has been created and the weights of the edges have been determined, any graph-based ranking algorithm for folksonomies e.g. FolkRank can be applied to calculate the scores of each node. To provide personalized recommendations relevant to a learner's current learning activity, the node representing the learner in the graph as well as the node representing the current learning activity of the learner are given more emphasis according to the intelligent surfer model explained in Appendix D.



**Figure 4.1:** Edge Weight Assignments in AScore

#### 4.1.2 AlInheritScore

AlInheritScore is based on GRank [1] as described in Section 2.3, thus activities are treated similarly to how GRank treats groups but with some differences. AlInheritScore assumes that the activity hierarchies can be exploited as an additional structure to improve the spread of weights through the folksonomy graph, thereby providing the learner with relevant recommendations of learning resources from sub-activities or activities higher up in the activity hierarchy. AlInheritScore contrasts to GRank in the following points:

- AlInheritScore does not consider activities to be resources and thus activities cannot be assigned tags.
- AlInheritScore considers activity hierarchies as well as users assigned to activities when computing the scores.
- Activity hierarchies are leveraged by the inheritance of scores.
- The inherited scores are emphasized by considering the connections in the activity hierarchy.
- The distance between activities in the hierarchy are considered as well.

AlInheritScore computes for each query user  $q_u \in Q_u$  a score vector that contains the score values  $score(r)$  for each resource  $r$ , as described in Section 3.1. Each query user  $q_u \in Q_u$  however first needs to be transformed into a set of query tags  $Q_t$  representing the user  $u$ , depending on the number of tags the user  $u$  has.

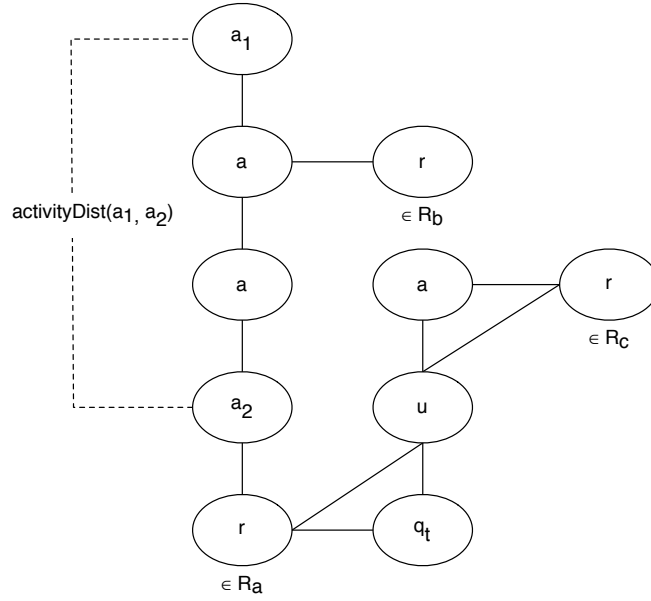
- First of all, the total number of tags  $t_{sum}$  that a user  $u$  has assigned to resources  $R_{u, t}$ , as shown in Equation 4.9 is calculated.

$$t_{sum} = \sum_{t \in T} R_{u, t} \quad (4.9)$$

Where  $R_{u, t} = \{ r \in R \mid (u, t, r) \in Y \} \subseteq R$  is the set of all resources that user  $u$  assigned a tag  $t$ .

- The user's tags  $t$  that have been assigned to resources and counted in  $t_{sum}$  are taken as query tags  $q_t \in Q_t$ .
- The weight  $w(q_t)$  of each of the query tags  $q_t$  is increased in Equation 4.10.

$$w(q_t) = w(t) + w(u) \cdot \frac{w(u, t)}{t_{sum}} \quad (4.10)$$



**Figure 4.2:** Determining Weights in AInheritScore

Where  $w(u, t) = |R_{u,t}|$  is the tag's frequency of usage by user  $u$ , i.e. the number of resources that user  $u$  assigned the tag  $t$ . User  $u$ 's initial weight in the graph  $w(u)$  is also considered.

Each query tag  $q_t \in Q_t$  has been initialized with an initial weight  $w(q_t)$ . This set of query tags  $Q_t$  is used to calculate the score values  $score$  for AInheritScore as mentioned above. The parameters  $d_a, d_b, d_c$  are defined to emphasize the *inherited* scores gained by relations in the activity hierarchy. The values used for the parameters  $d_a, d_b, d_c$  are determined in Appendix B.2.1.

- $d_a$  for resources having the query tag directly assigned to them
- $d_b$  for resources in the activity hierarchy having a resource that is tagged with the query tag
- $d_c$  for users in the activity hierarchy having assigned the query tag

The distance between activities in the hierarchy are determined as the activity distance  $activityDist(a_1, a_2)$  between two activities as shown in Figure 4.2. This is calculated as the number of hops from activity  $a_1$  to activity  $a_2$ , where  $a_1, a_2 \in A$  from the CROKODIL extended folksonomy  $F_C$  as defined in Section 3.1.2. It is also possible to calculate a lesser distance between sub-activities, or include the fan-out in the computation.

The steps in the AInheritScore algorithm are described below. For each query tag  $q_t$ , a score vector  $score$  is calculated.

- First, the score vector  $score = \vec{0}$  is initialized.
- Then  $R_q = R_a \cup R_b \cup R_c$  are determined where:
  - $R_a$  contains all resources with the query tag  $q_t$  directly assigned to them  $w(q_t, r) > 0$ .
  - $R_b$  contains all resources belonging to the same activity hierarchy as another resource  $r$ , that has the query tag  $q_t$  directly assigned to it:  $w(q_t, r) > 0$
  - $R_c$  contains all resources belonging to the same activity hierarchy as a user  $u$ , who has tagged a resource with the query tag  $q_t$ :  $w(u, q_t) > 0$
- For all  $r \in R_q$  belonging to activity  $a \in A$  execute the following steps:
  - Increase the score value of  $r$  as shown in Equation 4.11.

$$score(r) += w(q_t, r) \cdot d_a \quad (4.11)$$

- For each  $r' \in R_q$  belonging to activity  $a'$ , where  $a'$  and  $a$  are in the same activity hierarchy, increase again the score of  $r$  as shown in Equation 4.12.

$$score(r)+ = \frac{w(q_t, r')}{activityDist(a, a')} \cdot d_b \quad (4.12)$$

- For each  $u \in U$  working on activity  $a'$  i.e.  $(u, a') \in Y_U$ , increase again the score of  $r$  when  $a' \in A$  and  $a \in A$  are in the same activity hierarchy as shown in Equation 4.13.

$$score(r)+ = \frac{w(u, q_t)}{activityDist(a, a')} \cdot d_c \quad (4.13)$$

- The score vector  $score$  has the final scores for the query tag  $q_t$ .

The final scores for the original query user  $q_u$  is the aggregation (maximum or average) of all scores for each query tag  $q_t \in Q_t$ .

---

## 4.2 Hybrid Approach Exploiting Semantic Tagging (AspectScore)

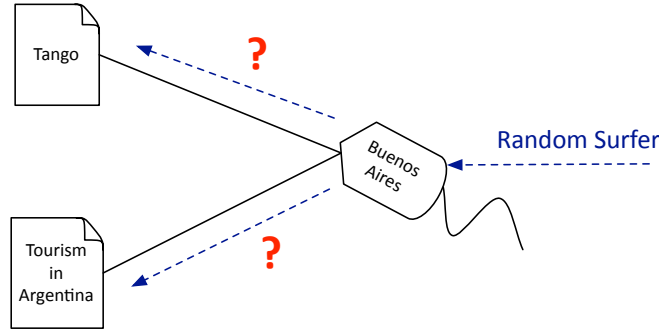
---

Graph-based algorithms for folksonomies based on the random surfer model face a major challenge of deciding which node to visit next. The surfer is dependent on following the existing links between nodes in the folksonomy graph. However these links may sometimes lead the surfer astray. This challenge is known as *Concept Drift*. Tags in many social tagging systems are often created freely without restrictions, thus they may not describe the content of a resource well, as users may use tags to express subjective opinions [35] e.g. *important* or personal goals e.g. *to read*. These non-descriptive tags could lead to irrelevant nodes and thus to concept drift. Concept drift could also be due to tag ambiguity, where the same tag could be used to mean different things. For example, as shown in Figure 4.3, the tag *Buenos Aires* could mean the topic of an Argentinian tourism web page or it could mean the location of a news article about the Argentinian dance *Tango*. Therefore if the surfer is actually looking for resources about tourism in Argentina, following the tag *Buenos Aires* could lead wrongly to the latest news about *Tango* and not about tourism in *Buenos Aires*.

A contribution to this thesis is the hybrid recommender algorithm **AspectScore (IncentiveScore)** [202, 203] that aims to alleviate the problem of concept drift by exploiting the additional semantic information gained from semantic tag types, as introduced in Chapter 2, to disambiguate tags. AspectScore is based on the CROKODIL extended folksonomy  $F_C$  as defined in Section 3.1.2 where tag types are introduced to the folksonomy giving each tag a *type*  $\in \{topic, location, event, genre, person, other\}$ . When a learner assigns a tag with its corresponding type to a learning resource, the typed tags become part of the tag assignment  $Y_T \subseteq U \times T_{typed} \times R$ , thereby giving the folksonomy additional semantic information about the tag assignment through the semantic tag types.

Learners often have a concept in mind while tagging which describes an aspect of a learning resource to a user [31, 35]. Semantic tag types describe these aspects of the learning resource and thus provide additional information about the learning resource that could be used for recommendations. Inspired by the intelligent surfer model of PageRank (see Appendix D), the graph is dynamically adapted giving priority depending on the types of tags thereby influencing the surfer's navigation through the graph. Thus the surfer can focus on links related to tags of a certain type e.g. the tag type *Topic*, as tags with this type describe the content of the resource well. The concepts or aspects of learning resources the learner is interested in can be said to be represented by the semantic tag types of the learner, thereby defining the learning context of the learner. Thus by focusing on a certain tag type, AspectScore can provide personalized recommendations of learning resources relevant to the learning context of the learner. The AspectScore algorithm is described below.

AspectScore requires a tag as query entity. Therefore each query user  $q_u \in Q_u$  needs to be transformed into a set of query tags  $Q_t$  representing the user  $u$ . Where  $Q_u \subseteq U$  and  $Q_t \subseteq T_{typed}$ .



**Figure 4.3:** Concept Drift due to Tag Ambiguity

- First of all, the total number of tags  $t_{sum}$  that a user  $u$  has assigned to resources  $R_{u,(t,type)}$ , as shown in Equation 4.14 is calculated.

$$t_{sum} = \sum_{(t,type) \in T_{typed}} R_{u,(t,type)} \quad (4.14)$$

Where  $R_{u,(t,type)} = \{ r \in R \mid (u, (t, type), r) \in Y_T \} \subseteq R$  is the set of all resources that user  $u \in U$  assigned a typed tag  $(t, type) \in T_{typed}$ .

- The user's typed tags  $(t, type) \in T_{typed}$  that have been assigned to resources and counted in  $t_{sum}$  are now taken as query tags  $q_t \in Q_t$ .
- The query tags are now weighted according to the usage frequency of the user and the tag types. The weight  $w(q_t)$  of each of the query tags  $q_t$  is increased as shown in Equation 4.15.

$$w(q_t) = w((t, type)) + w(u) \cdot \frac{w(u, (t, type))}{t_{sum}} \cdot \delta_{type} \quad (4.15)$$

Where  $w(u, (t, type)) = |R_{u,(t,type)}|$  is the typed tag's frequency of usage by user  $u$ , i.e. the number of resources that user  $u$  assigned the typed tag  $(t, type)$ . User  $u$ 's initial weight in the graph  $w(u)$  is also considered. The tag types act as an additional weighting factor defined by the parameters  $\delta_{type}$  and  $\gamma_{type_{q_t}, type_t}$ . The values of  $\delta_{type}$  and  $\gamma_{type_{q_t}, type_t}$  are determined in Appendix B.2.1.

Each query tag  $q_t \in Q_t$  has now been initialized with an initial weight  $w(q_t)$ . This set of query tags  $Q_t$  is now used to calculate the score values  $score$  for AspectScore.

- For each query tag  $q_t \in Q_t$ , an extended folksonomy graph  $G_C = (V_C, E_C)$  is created.
- $G_C$  is adapted depending on the query tag  $q_t$ :
  - For each tag node  $(t, type) \in V_C$  in the folksonomy graph, adjust the edge weights between the tag node and all other adjacent nodes in the folksonomy graph as shown in Equation 4.16.

$$w((t, type), e) = w((t, type), e) \cdot \gamma_{type_{q_t}, type_t} \quad (4.16)$$

Where  $e \in V_C$ .

- For each user node  $u \in V_C$  in the folksonomy graph, adjust the edge weights between the user node and all other adjacent nodes in the folksonomy graph as shown in Equation 4.17.

$$w(u, e) = w(u, e) \cdot \gamma_{type_{q_t}, type_t} \quad (4.17)$$

Where  $e \in V_C$ .



- For each resource node  $r \in V_C$  in the folksonomy graph, adjust the edge weights between the resource node and all other adjacent nodes in the folksonomy graph as shown in Equation 4.18.

$$w(r, e) = w(r, e) \cdot \gamma_{type_{q_t}, type_e} \quad (4.18)$$

Where  $e \in V_C$ .

- Execute FolkRank on this adapted graph.

The final scores for the original query user  $q_u$  are now the aggregation (maximum or average) of all scores for each query tag  $q_t \in Q_t$  applying the weights of the respective query tags.

---

### 4.3 Hybrid Approach Exploiting Semantic Relatedness between Tags (InteliScore)

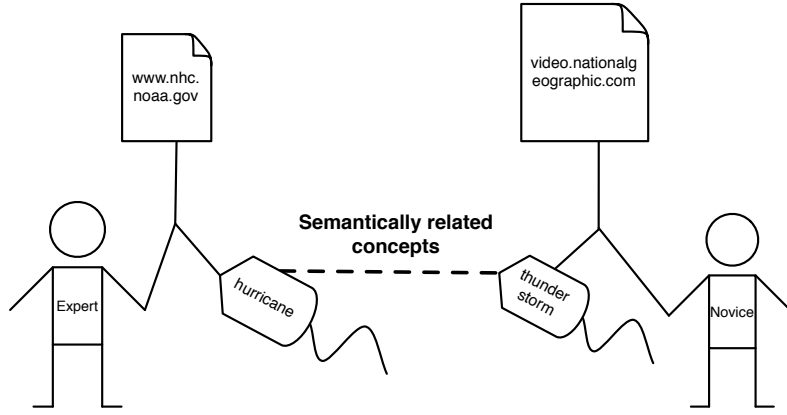
---

Another solution to overcome the challenge of concept drift of the random surfer as presented in Section 4.2 is to lead the surfer to more relevant nodes by identifying tags that are semantically more strongly related to the tag the surfer is coming from. Semantic relatedness can help to alleviate concept drift caused by the ambiguity of tags. Assume two synonym tags are connected to the same resource and imagine a surfer comes from one of these tags to the resource. As the semantic relatedness measure is ideally the maximal value between these two tags, it is thus possible to reduce concept drift by attenuating the connections to other tags connected to the resource. The multi-facetedness of entities in the folksonomy is another reason that may introduce concept drift, e.g. a resource may be about several entirely different topics. Semantic relatedness can be leveraged to reduce concept drift caused by multi-facetedness of entities. Connected entities can be attenuated depending on the semantic relatedness to a tag the surfer originated from.

Semantic relatedness between tags in a folksonomy considers more than just the textual similarity between the tags but rather the similarity between the concepts these tags represent. Recommendations of learning resources based on the semantic relatedness between their tags can help learners to find resources with similar concepts even when the learner does not know the exact terminology to describe these concepts, especially when the learners have different levels of knowledge. A novice who does not know the precise terminology of a domain would be unaware of the right terminology to use to find related resources. An expert in the domain would however use such specific terminology to tag related resources. Thus depending on the level of expertise of a learner, different aspects of learning resources are interesting to the learner either giving a broad overview of the domain or a narrow specific view [222]. For example, as shown in Figure 4.4, a novice learner working in the domain of tropical storms would more likely tag a learning resource about tornadoes with the more general tag *thunderstorm* whilst an expert would be able to be more precise and use a more specific tag like *hurricane*. The concepts of a *thunderstorm* and a *hurricane* are semantically related and thus in the example, a connection can be determined between the two learning resources via these tags, thereby providing the novice with a recommendation that gives him a new perspective on tropical storms, tornadoes and hurricanes.

As a contribution to this thesis, a hybrid graph-based recommender algorithm that exploits the semantic relatedness between tags is presented. **InteliScore** [203] aims to alleviate the problem of concept drift by extending the folksonomy graph with new relationships between tags based on their semantic relatedness. InteliScore is inspired by the intelligent surfer model of PageRank (explained in Appendix D). Semantic relatedness is used to disambiguate tags in the folksonomy and thereby have the surfer focus on links to tags that are semantically strongly related. Explicit Semantic Analysis [85] can be used to calculate the semantic relatedness between tags by exploiting their textual content and the semantic information from Wikipedia. InteliScore applies XESA [222], a method based on Explicit Semantic Analysis [85], to calculate the semantic relatedness between tags.

InteliScore requires a tag as query entity. Therefore each query user  $q_u \in Q_u$  needs to be transformed into a set of query tags  $Q_t$  representing the user  $u$ , similar to AInheritScore and AspectScore, where the tags are weighted by the usage frequency of the user.



**Figure 4.4:** Example of semantically related tags in a folksonomy

- First of all, the total number of tags  $t_{sum}$  that a user  $u$  has assigned to resources  $R_{u,t}$ , as shown in Equation 4.19 is calculated.

$$t_{sum} = \sum_{t \in T} R_{u,t} \quad (4.19)$$

Where  $R_{u,t} = \{ r \in R \mid (u, t, r) \in Y \} \subseteq R$  is the set of all resources that user  $u$  assigned a tag  $t$ .

- The user's tags  $t$  that have been assigned to resources and counted in  $t_{sum}$  are taken as query tags  $q_t \in Q_t$ .
- The weight  $w(q_t)$  of each of the query tags  $q_t$  is increased in Equation 4.20.

$$w(q_t) = w(t) + w(u) \cdot \frac{w(u, t)}{t_{sum}} \quad (4.20)$$

Where  $w(u, t) = |R_{u,t}|$  is the tag's frequency of usage by user  $u$ , i.e. the number of resources that user  $u$  assigned the tag  $t$ . User  $u$ 's initial weight in the graph  $w(u)$  is also considered.

Each query tag  $q_t \in Q_t$  has been initialized with an initial weight  $w(q_t)$ . This set of query tags  $Q_t$  is used to calculate the score values  $score$  for IntelliScore.

- For each query tag  $t_q$ , the folksonomy graph  $G = (V, E)$  is adapted to give priority to this query tag  $t_q$ . This means the weight of the tag  $t_q$  is set in the topic-sensitive probability vector as described in Appendix D.
- For each tag  $t$  in the graph  $G = (V, E)$ , adjust the weight of each incident edge  $e \in E$  with the XESA semantic relatedness value  $XESA(t, t_q)$  between this tag  $t$  and the query tag  $t_q$  as shown in Equation 4.21.

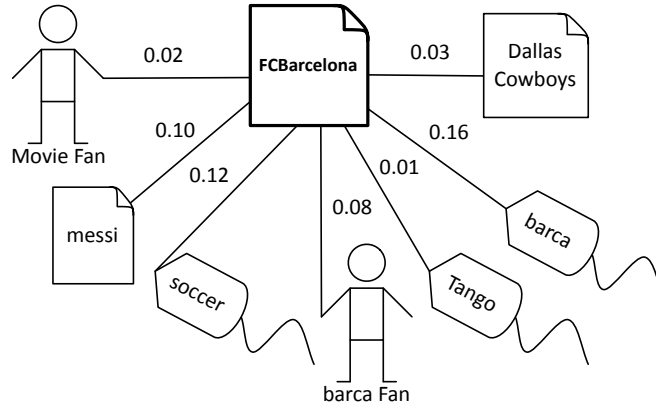
$$w(e, t) = \max(w(e, t) \cdot XESA(t, t_q)) \quad (4.21)$$

Where  $e \in E$  is an incident edge to  $t$ .

- When multiple adjustments are made to the same edge  $e$ , the maximum edge weight is taken. The tags are pre-processed by stemming, lowercase normalization and averaging the relatedness of a tag consisting of multiple tokens.
- Execute FolkRank on this adapted graph.
- The results of each query user  $q_u \in Q_u$  are accumulated (the average or maximum score), thereby applying the weights of the respective query tags  $w(q_t)$ .

The semantic relatedness measure in IntelliScore can be set to any semantic relatedness measure that calculates the semantic similarity between tags. For example, the semantic similarity based on a taxonomy could be used such as applied in [174, 175].





**Figure 4.5:** Example of the context of an entity (*FCBarcelona*) in a folksonomy

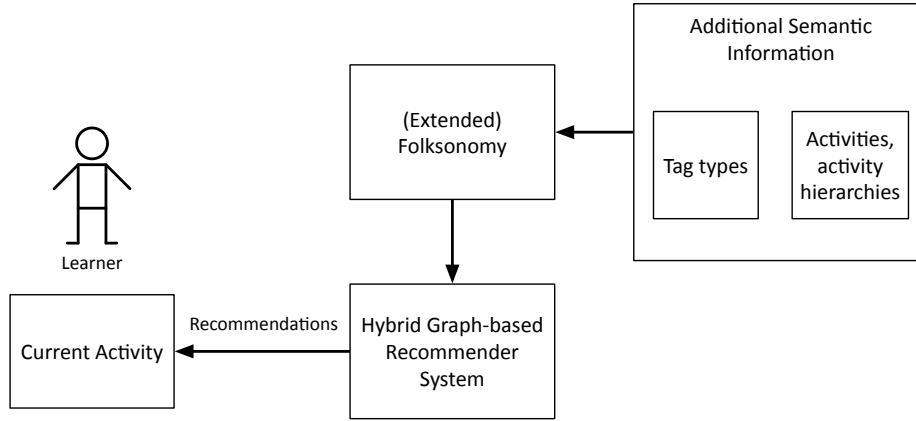
#### 4.4 Hybrid Approach Exploiting Context-Specific Information in Folksonomies (VSScore)

Folksonomies often do not contain much textual information apart from the names of the tags or users and the URL of a web resource. The content of a resource, the interests of a user or the semantic of a tag is therefore hard to determine. The context-specific information found in a folksonomy could however be used to give added semantic information to the entities of a folksonomy. The context of an entity  $e$  of the folksonomy is modeled as shown in Figure 4.5. The context of  $e$  is given by the strength of the relations between  $e$  and all other entities in the folksonomy. In the example,  $e$  is depicted as the resource *FCBarcelona*. The context of the resource *FCBarcelona* is thus given by the strengths of the relations to the other entities in the folksonomy. As can be seen, the resource *FCBarcelona* is more strongly related to the tags *barca* and *soccer* and to the resource *messi* and user *barca Fan* with relation strengths of 0.16, 0.12, 0.10 and 0.08 respectively. The resource *FCBarcelona* is however less strongly related to the resource *Dallas Cowboys*, to the user *Movie Fan* and to the tag *Tango* with relation strengths of 0.03, 0.02 and 0.01 respectively. Thus, the resource *FCBarcelona* can be concluded as having more to do with soccer than with American football.

The learning context of a user in a folksonomy could thus be represented as the context of the user or the contexts of the resources of the user in the folksonomy. As a result, the context-specific information found in a folksonomy could be exploited to provide recommendations of learning resources relevant to the learning context of the learner in the folksonomy. A hybrid recommender approach exploiting context-specific information gained from a folksonomy is presented as a contribution to this thesis.

**VSScore** [204] adapts the vector space model [153] to folksonomies where query entities and resources are represented as vectors and the distance between a query vector and a resource vector are the corresponding scores between the query and that resource. The aim of VSScore is to alleviate concept drift and improve the relevance of recommendations to a query entity. An assumption made by VSScore is that the entities in a folksonomy represent semantic concepts and the context of an entity  $e$  is given by the strength of the relations between  $e$  and all other entities in the folksonomy  $F := (U, T, R, Y)$ . In VSScore, the strength of the relations are represented by the FolkRank values calculated between the entities of the folksonomy. The relation strengths could be represented by any other ranking algorithm. The following steps describe the VSScore algorithm.

- Create a vector representation  $\vec{s}_e$  of semantic concepts for each entity  $e$  in the folksonomy  $F$ . The query entity  $q_e \in Q_e$  is also represented as a vector  $\vec{s}_{q_e}$ . FolkRank is used to calculate the relation strengths between the query entity and all other entities in the folksonomy.
- Calculate the vector distance between the query entity vector  $\vec{s}_{q_e}$  and all vectors  $\vec{s}_r$  representing resource entities in the folksonomy. The vector distances thereby represent the scores of the rec-



**Figure 4.6:** Conceptual Design of a Personalized Recommender System for Resource-based Learning (adapted from [17])

ommended resources for this query entity. The cosine-similarity [153] is used to determine the distances between the vectors in the vector space model as shown in Equation 4.22.

$$\text{cosineSimilarity}(\vec{s}_{q_e}, \vec{s}_r) = \frac{\vec{s}_{q_e} \cdot \vec{s}_r}{\|\vec{s}_{q_e}\| \cdot \|\vec{s}_r\|} \quad (4.22)$$

Any other method could be used instead of the cosine-similarity. The cosine-similarity is used in this implementation as it is a standard measure to quantify query-resource similarity of the respective vector representations in a vector space model.

#### 4.5 Providing Personalized Recommendations to Support Resource-based Learning

In order to provide recommendations of learning resources in a resource-based learning scenario, CROKODIL is chosen as a concrete implemented application scenario. As presented in Section 2.4, the additional semantic information identified in CROKODIL's extended folksonomy could be utilized to improve the recommendation of learning resources [17]. The structural recommendations existing in CROKODIL [197] were based on the structure of the semantic graph, but neither considered the learner's learning goals nor the additional semantic information found in an extended folksonomy. In Figure 4.6, a conceptual design of a personalized recommender system is presented, considering the resource-based learning scenario and based on the analysis of CROKODIL as a concrete application scenario [17]. The learner working on a particular activity is given recommendations relevant to this current activity. This current activity contributes to the additional semantic information included in an extended folksonomy. This extended folksonomy contains additional semantic information gained from the community of learners such as semantic tag types, activities and activity hierarchies. The recommendations suggested for the current activity are generated from a hybrid graph-based recommender system that runs on the aforementioned extended folksonomy, using the various hybrid recommender algorithms integrated into CROKODIL. Figure 4.7 shows a screenshot of the CROKODIL implementation of a learner's current activity. To the left is the learner's hierarchical activity structure showing all the learner's activities and sub-activities in a hierarchy. To the right, several learning resources are recommended to this learner for the current learning activity. The current activity being viewed by user *Moji* in this example is the activity *Finding out how PageRank works*. This is a sub-activity of the activity *Reading up on link-based ranking algorithms*.

Die Plattform für Ressourcen-basiertes Lernen

CROKODIL Projektwebsite

Startseite | Hilfe | Abmelden

Mein Profil | Meine Kontakte | Meine Gruppen | Meine Aktivitäten | Meine Ressourcen | Meine Tags | Nachrichten

**Suche**

Suchen

Ohne Einschränkung

**Neu erstellen... (3)**

- Wissensressource
- Aktivität
- Gruppe

**Meine Aktivitäten**

- Pervasive Learning State of the Art
- Preparing presentations, talks and posters
  - Preparing a poster
    - Finding poster templates
  - Preparing Presentations
    - Defining the contents of the presentation
    - Holding the talk
    - Planning the talk
    - Searching for cartoons
    - Searching for pictures
- Reading up on link-based ranking algorithms
  - Finding out how PageRank works**
    - Finding material explaining Link Analysis
  - Social Networks and Recommendation Systems
  - Understanding the Climate Change
  - Abgeschlossene Aktivitäten

**Zuletzt besuchte**

- Finding out how PageRank works

**Aktivität**

### Finding out how PageRank works

Wollen Sie dieser Aktivität Ressourcen, Teilaktivitäten oder Benutzer hinzufügen, nutzen Sie die Eingabefelder unten.

Wollen Sie die Informationen über die Aktivität bearbeiten, wählen Sie bitte das Stift-Symbol (Bearbeiten) oben links aus.

erstellt von

**Über diese Aktivität**

Titel	Finding out how PageRank works
Aktivität abgeschlossen?	
Zur Ansicht freigegeben für	alle CROKODIL Benutzer
Beschreibung	PageRank is a link-based ranking algorithm created in 1998 by Page and Brin. Google uses PageRank as one of its search algorithms.

**Empfehlungen (5)**

- Wie Google mit Milliarden unbekannten rechnet
- The PageRank Citation Ranking: Bringing Order to the Web
- Link Analysis and Web Search
- Introduction to Information Retrieval - lectures
- Link Analysis: An information science approach

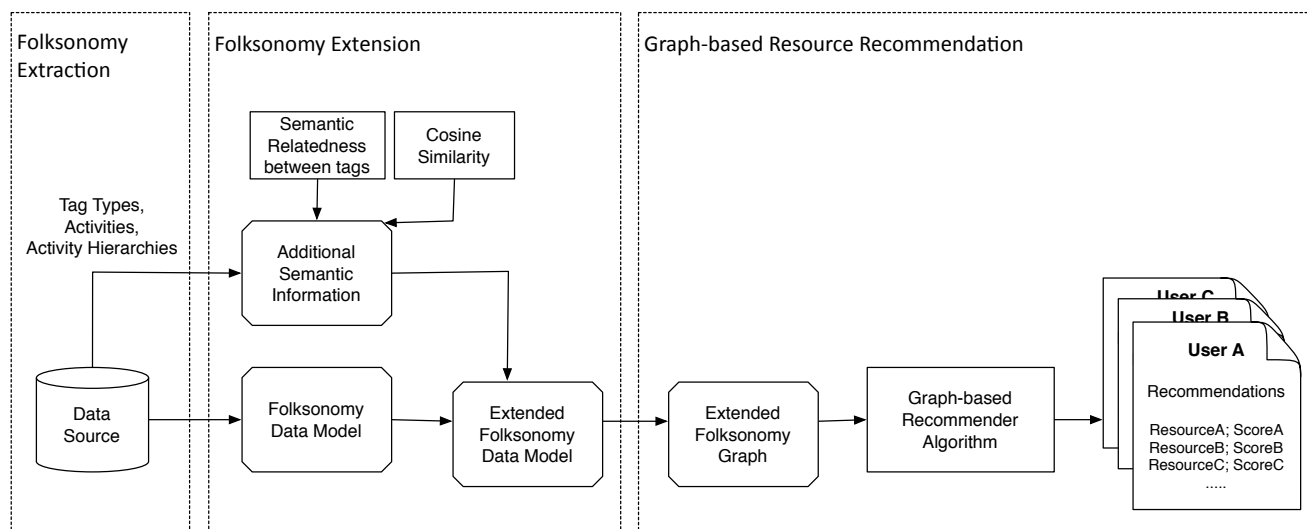
**Post**

Sie haben 12 ungelesene Nachrichten: Posteingang.

**Umfrage**

Empfehlungen bewerten

**Figure 4.7:** Screenshot showing an Activity, Activity Hierarchy and Learning Resource Recommendations



**Figure 4.8:** Hybrid Recommender Algorithm Concept

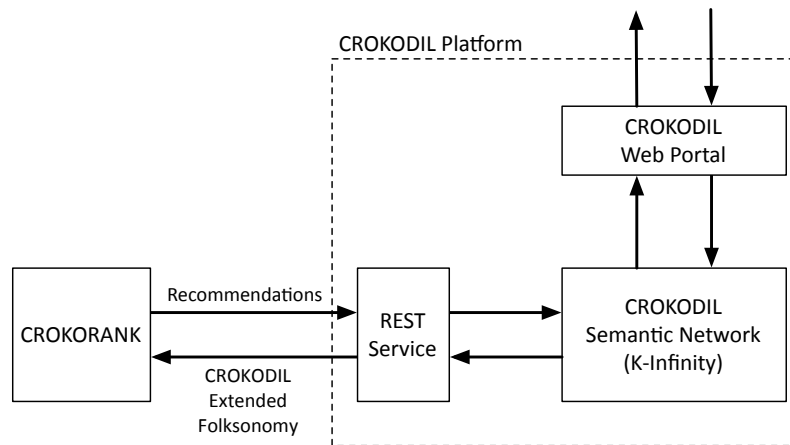
#### 4.5.1 Hybrid Recommender Systems for Extended Folksonomies

Based on the CROKODIL application scenario presented in Section 2.4, the additional semantic information gained from semantic tag types, activities and activity hierarchies are considered in this thesis to form an extended folksonomy. A feature combination hybrid is the best fitting recommender approach for such a scenario [27, 47]. Feature combination hybrids include additional information from diverse knowledge sources to enhance a dataset, in this case a folksonomy, that is then used to create recommendations [47] from a graph-based recommender algorithm running on this extended folksonomy. As shown in Figure 4.8 (adapted from [174]), a folksonomy data model from a data source is created and augmented with additional semantic information either extracted from the data source or included from other sources such as from an approach providing tag similarity measures or semantic relatedness measures between tags. The data source could be a database from a social tagging application like CROKODIL. The extended folksonomy data model presented in Section 2.4 is transformed into a graph representation and passed on to a graph-based recommender algorithm such as those presented in the following sections.

#### 4.5.2 Implementation and Integration in CROKODIL

As a proof-of-concept, AScore has been integrated into the CROKODIL scenario to provide recommendations for learners enabling them to discover relevant learning resources to a specific activity. CROKORANK [202] is a framework designed to generate resource recommendations for the application scenario CROKODIL. The CROKODIL architecture [16] and the integration of CROKORANK is shown in Figure 4.9. In the CROKODIL platform, the extended folksonomy is represented as a semantic network, where the users, resources, tags, tag types, activities and activity hierarchies are modeled as nodes and the relationships between them as edges. K-Infinity<sup>1</sup> is used as the data management platform and provides all modeling components for the creation and maintenance of the underlying semantic network [16]. The pedagogical concept presented in [16] has been implemented in CROKODIL by introducing a new node type *activity*. Activities can be organized hierarchically by the relation *is part of activity* and its inverse relation *is parent activity to*. This allows the structuring of complex activity hierarchies. The CROKODIL platform is accessible via a Web portal as shown in Figure 4.9.

<sup>1</sup> <http://www.i-views.de>, retrieved 03.06.2014



**Figure 4.9:** Integration of CROKORANK in the CROKODIL Architecture

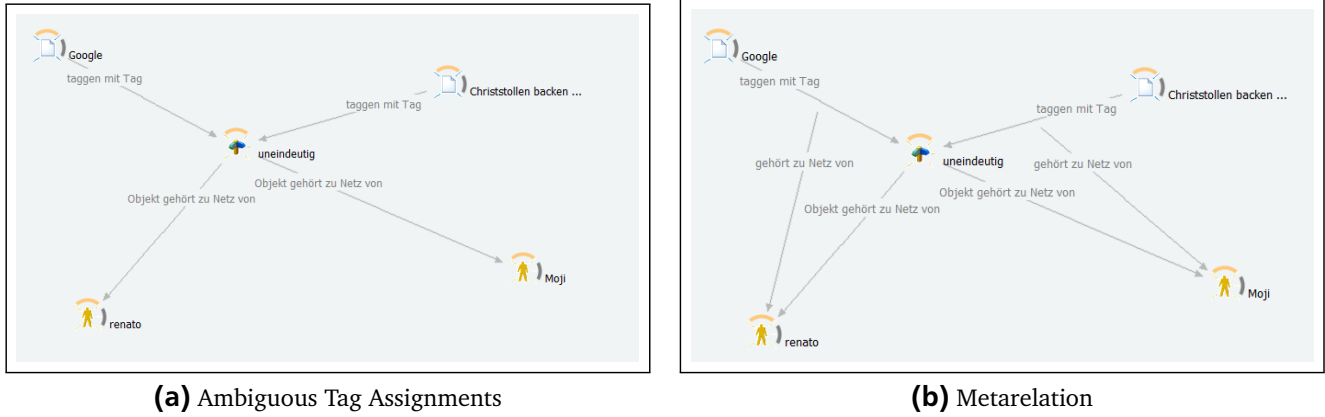
CROKORANK requires the entire CROKODIL extended folksonomy: all users, resources, tags, tag types, activities and all the relations between them, in order to generate recommendations. CROKODIL's extended folksonomy is extracted from the CROKODIL platform as XML via a REST service. Each tag assignment in the folksonomy is extracted as a resource-tag-user triple defined as an XML node *rtuTriple*. Each activity assignment is extracted as a resource-activity-user triple defined by the XML node *rauTriple*. An example of an XML *folksonomy* node is shown below having an example *rtuTriple* and an example *rauTriple*.

```

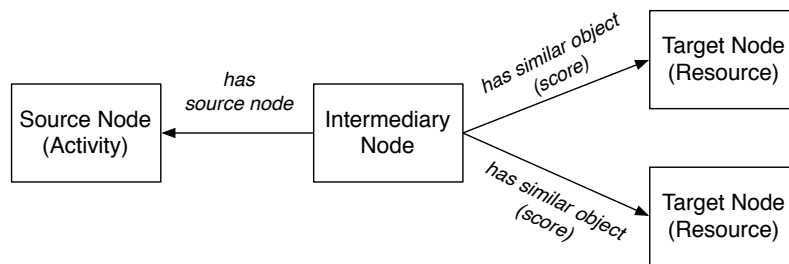
<folksonomy>
  <rtuTriple>
    <resource>ID63763_307692169</resource>
    <tag>ID62463_20750216</tag>
    <user>ID62763_90469456</user>
  </rtuTriple>
  <rauTriple>
    <resource>ID53743_307692149</resource>
    <activity>ID43733_32669236</activity>
    <user>ID53763_40969219</user>
  </rauTriple>
</folksonomy>

```

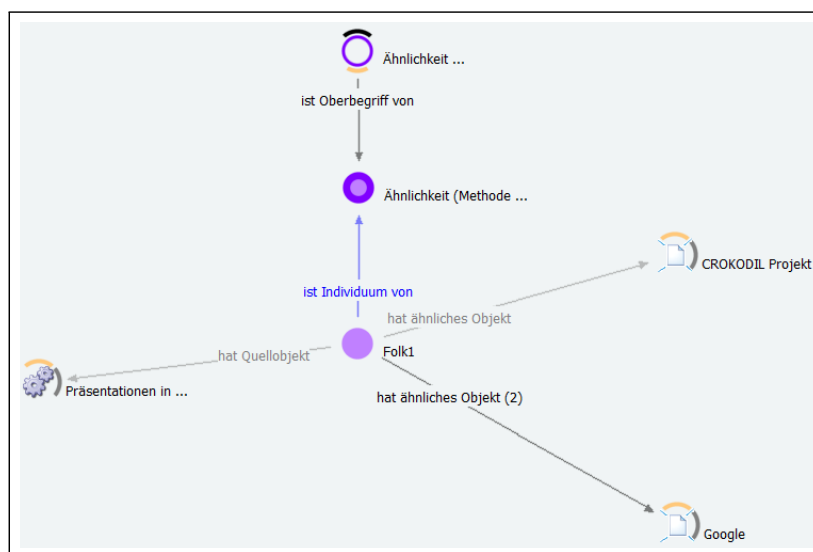
In order to be able to determine which user is responsible for creating a tag assignment or an activity assignment, a *metarelation* is required. A metarelation is a relation between a user node and the relationship between a resource and a tag or the relationship between a resource and an activity. In Figure 4.10(a), an example is shown of ambiguous tag assignments in CROKODIL's semantic network. Without a metarelation, any of the users could have attached the tag *uneindeutig* to any of the resources. In Figure 4.10(b), with the introduction of metarelations, it can now be clearly interpreted that the user named *Moji* has attached the tag *uneindeutig* to the resource *Christstollen backen* because the metarelation *gehört zu Netz von* is created between the user *Moji* and the relationship *taggen mit Tag*. Resource recommendations are generated by CROKORANK and sent to the CROKODIL platform via a REST service. CROKORANK returns the recommended resources and their scores with regard to a particular activity. In CROKODIL's semantic network, these scores are stored as attributes of the relationships between the resources and the activity as shown in Figure 4.11. The activity is given an intermediary node for each recommender algorithm to enable a more efficient traversal of the semantic network. Figure 4.12 shows an example of two resource recommendations *CROKODIL Project* and *Google* made to the activity *Präsentationen* by the recommender algorithm *AScore*. The node *Folk1* acts as an intermediary node between the activity node and the two resource recommendation nodes for *AScore*. At run-time, depending on



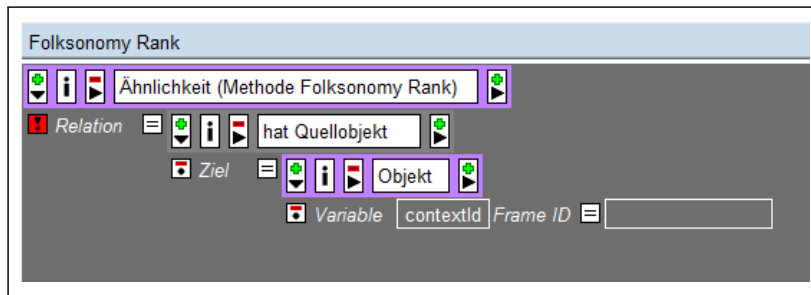
**Figure 4.10:** Screenshots showing the Modeling in K-Infinity of a Metarelation in CROKODIL's Semantic Network



**Figure 4.11:** Assigning Resource Recommendations to a particular Activity in CROKODIL's Semantic Network



**Figure 4.12:** A Screenshot of CROKODIL's Semantic Network in K-Infinity showing Resource Recommendations to an Activity via an Intermediary Node



the recommender algorithm activated, the intermediary node is traversed with a K-Infinity *Expertensuche* to determine which resources to recommend to a specific activity. The scores are retrieved from the attributes on the relationships between the resource nodes and the intermediary node and used to create a ranked list of recommended resources for the activity. An example of an *Expertensuche* is shown in Figure 4.13. Resource recommendations for the method *Folksonomy Rank* are traversed that have as source node the current activity node with the specified *contextID*.

CROKORANK [202] is a Java<sup>2</sup> implementation using the Guice dependency injection framework<sup>3</sup>. From a specified datasource, such as a file, a database or a REST interface, CROKORANK creates a data model of an extended folksonomy based on the CROKODIL scenario. Thus the entities in the extended folksonomy data model include *Activity*, *Resource*, *Tag* and *User*. A *TagType* is implemented as part of a tag and comprises one of the options *TOPIC*, *LOCATION*, *EVENT*, *PERSON*, *RESOURCETYPE*, *ACTIVITY*, or *OTHER*. A *Tag Assignment* is implemented consisting of a tag, a user and a resource entity. An *Activity Assignment* is implemented consisting of a user, a resource and an activity entity. The folksonomy graph model is created from this extended folksonomy data model. Entities are added as nodes to the graph and tag assignments and activity assignments are translated into edges of the graph. Edges consist of the two entities it connects and the weight of the edge. Depending on the algorithm, a different graph creation strategy *GraphCreationStrategy* is implemented. For the different approaches, different ranking strategies are implemented as a *GraphRanker*. Several implementations of a *Run* are created such as an *Evaluation Run* for evaluations or a *CROKODIL Run* to generate recommendations for the CROKODIL's REST interface. For each run, a *GuiceModule* file is set up consisting of the configurations for all the parameters of a Run where the implementation of the *GraphCreationStrategy*, and the *GraphRanker* to be used for the run are specified along with all other required parameters such as the datasource, result output file name, amongst others. More details can be found in [202].

## 4.6 Summary

In this chapter, several hybrid graph-based recommender algorithms for TEL were presented and are summarized in Table 4.1 giving an overview of the fundamental approaches used, the learning context considered and the additional semantic information exploited by the recommender algorithms. AScore takes the learner’s current activity as well as the activity structure in the folksonomy as the learner’s learning context. AScore is based on GFolkRank and works on an extended folksonomy consisting of additional semantic information gained from activities and activity hierarchies. AInheritScore uses the learner’s activities as the learning context of a learner. AInheritScore is based on GRank and exploits the additional semantic information gained from activities and activity hierarchies. AspectScore uses the learner’s semantic tag types to create a learning context for the learner. Based on FolkRank, AspectScore runs on an extended folksonomy enriched with semantic tag types. InteliScore calculates the semantic relatedness between tags to create a learning context for the learner. FolkRank is run on the folksonomy

---

<sup>2</sup> <http://www.java.com>, retrieved 03.06.2014

<sup>3</sup> <https://github.com/google/quice>, retrieved 03.08.2014



Approaches	Learning Context	Fundamental Approach	Extended Folksonomy	Additional Semantic Information
AScore	Learner's current activity and activity structure in the folksonomy	GFolkRank	Yes	Activities and activity hierarchies
AINheritScore	Learner's activities and activity hierarchy	GRank	Yes	Activities and activity hierarchies
AspectScore	Learner's semantic tag types	FolkRank	Yes	Semantic tag types
InteliScore	Learner's tags	FolkRank	Yes	Semantic relatedness between tags
VSScore	Learner's context in the folksonomy	FolkRank, Vector space Model	No	Context-specific information found in a folksonomy

**Table 4.1:** Proposed Hybrid Graph-based Recommender Approaches for TEL

extended with the additional semantic information gained from the semantic relatedness between tags. VSScore exploits the context-specific information found in a folksonomy to create a learning context using FolkRank and the vector space model. More details regarding these approaches can be found in [202]. The evaluations of these algorithms are presented in the following chapter.



---

## 5 Evaluation of the Hybrid Graph-based Recommender Approaches

---

In this chapter, the hybrid recommender algorithms described in Chapter 4 are evaluated on different historical datasets using offline experiments. A new evaluation method LeaveRTOut [203] and a new evaluation metric Mean Normalized Precision [203] are presented.

---

### 5.1 Evaluation Methods and Evaluation Metrics

---

The evaluation methods applied to evaluate the hybrid recommender algorithms presented in Chapter 4 are based on offline experiments using a historical dataset. The resource recommendation task for the evaluations according to Table 3.1 is a personalized resource recommendation task with a user as query entity (interests match) [33, 202]. All proposed recommender algorithms recommend learning resources to a specific user, thus the current user  $u$  is said to be the query entity or input  $q_u \in Q_u$  as defined in Section 3.1, where  $Q_u$  is the set of all query entities representing all users in the folksonomy. The recommended learning resources are evaluated per query entity, in this case per user  $q_u \in Q_u$ .

---

#### 5.1.1 State-of-the-Art Evaluation Methods and Metrics

---

In the following, the state-of-the-art evaluation method LeavePostOut for evaluating recommender algorithms for folksonomies is presented as well as an overview of evaluation metrics used for the evaluations.

---

##### Evaluation Method for Folksonomies: LeavePostOut

---

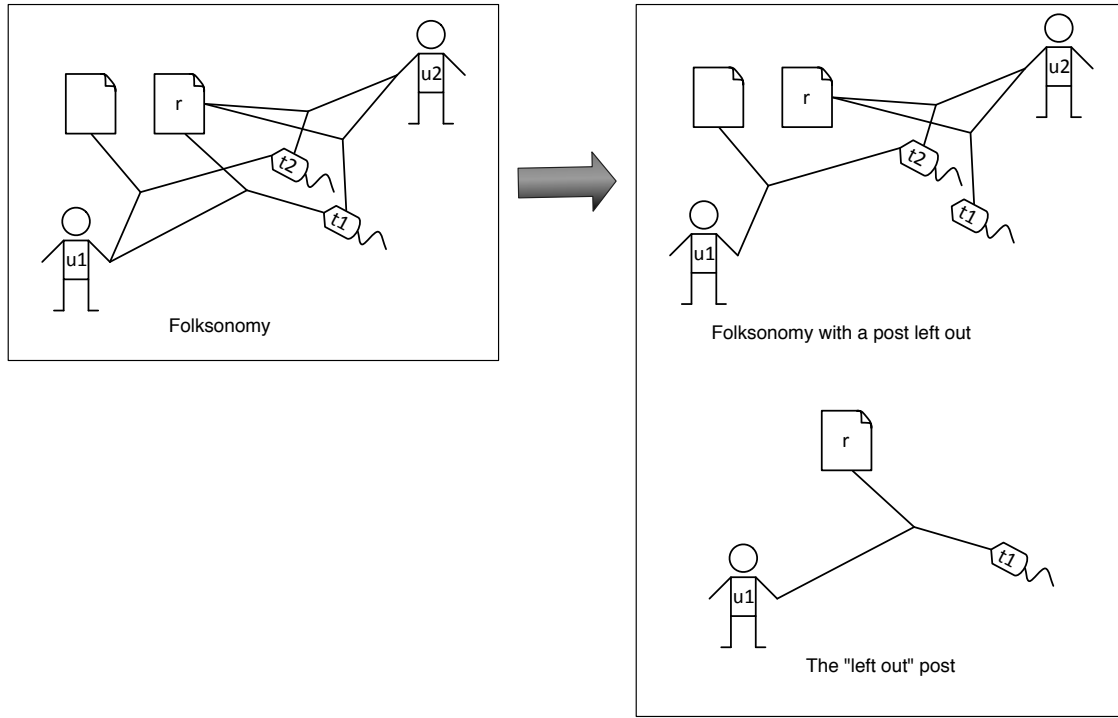
Recommender algorithms for folksonomies are often evaluated using the evaluation method **Leave-PostOut** [111]. LeavePostOut is based on the standard evaluation approach Leave-One-Out cross-validation as described in Section 3.2 but adapted to folksonomies. Analogous to a fold, a partition called a **post** [111] is defined for folksonomies. A post  $P_{u,r}$  is defined in Equation 5.1 as all tag assignments of a specific user  $u$  to a specific resource  $r$  [111].

$$P_{u,r} = \{(u, t, r) | (u, t, r) \in Y\} \quad (5.1)$$

Where  $F := (U, T, R, Y)$  is a folksonomy as defined in Section 2.3 and  $u \in U, t \in T, r \in R$ .

LeavePostOut as shown in Figure 5.1 removes a post  $P_{u,r}$  of user  $u$  with resource  $r$  from the folksonomy  $F$ . This means all tag assignments of user  $u$  to resource  $r$  are removed from the folksonomy  $F$ . Thus no connections are left in the folksonomy that could connect user  $u$  directly to resource  $r$  [111]. However, there could still be information in the folksonomy that could indirectly connect the resource  $r$  to the user  $u$  over a few hops, for example via another user or in the example in Figure 5.1 via tag  $t_2$ . The recommendation task *interests match* is now to recommend this specific resource  $r$  to user  $u$ , assuming that user  $u$  will find this resource relevant, as user  $u$  had assigned a tag to  $r$  in the removed post  $P_{u,r}$ . The user  $u$  of the post  $P_{u,r}$  is considered as input or query entity for the evaluation. This procedure is repeated for all posts of each query entity. The main assumption on which this evaluation approach is based is that the assignment of a tag by a user to a resource indicates that the user finds this resource relevant. The user and the assigned tag represent the information need for this recommendation task as explained in Section 3.1.

For a folksonomy  $F = (U, T, R, Y)$ , the LeavePostOut evaluation steps for a resource recommendation task would be the following.



**Figure 5.1:** LeavePostOut Evaluation Method

- For each user  $u \in U$ , determine all posts  $P_{u,r}$  for the user  $u$  and resource  $r \in R$ 
  - Remove one of these posts  $P_{u,r}$  from the folksonomy  $F$  as shown in Equation 5.2.

$$F = F \setminus P_{u,r} \quad (5.2)$$

- Run the algorithm to be evaluated on this modified folksonomy  $F$
- Evaluate the recommendations returned by the algorithm with regard to the current resource  $r$  from the post  $P_{u,r}$
- Add the removed post  $P_{u,r}$  back to the folksonomy  $F$  as shown in Equation 5.3.

$$F = F \cup P_{u,r} \quad (5.3)$$

- Repeat for each post of the current user.
- Repeat for each user.

---

## Evaluation Metrics

---

Evaluation results need to be quantified using standardized evaluation metrics in order to be able to report on the quality of the results and to be able to compare them to other evaluation results. In Appendix B, the basic evaluation metrics Precision, Recall and F-measure are explained and in the following Precision at  $k$ , Average Precision and Mean Average Precision are presented.

### Precision at $k$ - Precision( $k$ )

Precision at  $k$  is the precision of the top  $k$  retrieved items as shown in Equation 5.4 [153]. The number of retrieved items is limited to the top  $k$  items. This reflects a recommendation scenario where, for example,  $k = 10$ . Only the top 10 recommendations are shown in the ranked list of recommendations.

$$Precision(k) = \frac{|\text{relevant items retrieved}|_k}{k} \quad (5.4)$$

---

### Average Precision (AP)

AP is the average precision of all relevant items retrieved for a specified information need or query entity  $q$  as shown in Equation 5.5 [153].

$$AP(q) = \frac{1}{|M|} \sum_{k=1}^{|M|} P(R_k) \quad (5.5)$$

Where  $q$  is a query entity or an information need,  $M$  is the set of relevant items  $M = \{m_1, m_2, m_3, \dots, m_{|M|}\}$  with  $k \in [1, |M|]$  for the query  $q$ , which are the true positives and false negatives (tp + fn), as explained in Appendix B.  $R$  is the set of ranked recommended items made by an algorithm, consisting of true positives and false positives (tp + fp), for the query  $q$ . For example,  $R = \{rec_1, rec_2, rec_3, \dots, rec_n\}$  with  $i \in [1, n]$ . Thus,  $rec_i \in R$  is the item at position  $i$  in the ranked list of recommended items. The set  $R_k$  is a subset of  $R$  consisting of items from the top ranked recommended item  $rec_1$  till the relevant item  $m_k = rec_i$  is found in  $R$ . For the example, if  $k=1$  and  $m_1 = rec_3$ , then  $R_1 = \{rec_1, rec_2, rec_3\}$ . When a relevant item from  $M$  does not exist in  $R$ , the precision is taken as 0 in Equation 5.5.

### Mean Average Precision (MAP)

MAP takes the mean of the Average Precision (AP) for each information need  $q \in Q$  as shown in Equation 5.6 [21, 153].

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} AP(q_j) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{|M_j|} \sum_{k=1}^{|M_j|} P(R_{j_k}) \quad (5.6)$$

Where  $Q$  is the set of all information needs and  $q_j \in Q$  is a single information need or query entity with  $j \in [1, |Q|]$  and  $M_j$  is the set of relevant items for the query entity  $q_j$ .

---

## Statistical Significance Tests

Evaluation results can be analysed using statistical tools that test for significance, for example when comparing two different groups of data to determine if they are statistically different [83]. In Appendix B, the basic statistical significant tests used in this thesis are presented.

---

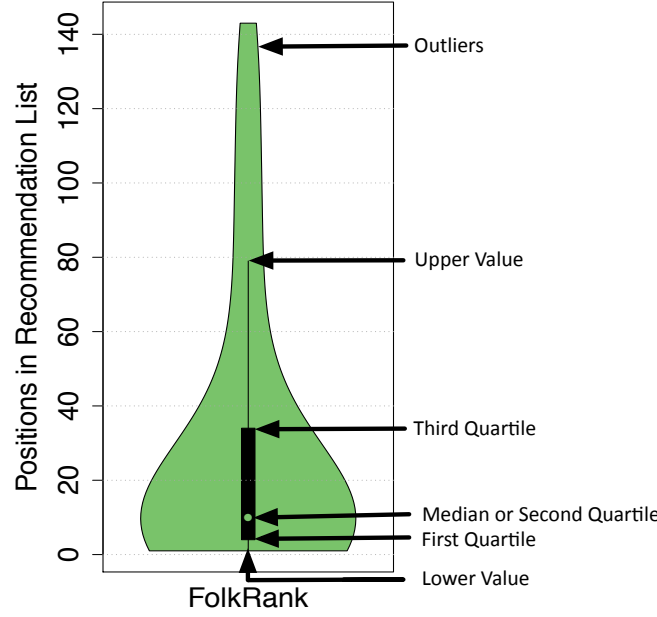
## Visualization of Evaluation Results

The ranked recommendation results are visualized as violin plots in order to have a better indication of the shape of the distribution of the recommendations. A violin plot is a combined box plot and a density trace plot [99]. Figure 5.2 shows an example of a violin plot showing the evaluation results from a recommendation task. The violin plot gives greater insight into the distribution of the evaluation results as the distributions of the positions of the recommended resources in the ranked list of recommendations can be seen. The width of the plot represents the density of the distribution at that position. Therefore the more dense the distribution is in the lower positions, the better the performance of the recommender approach, as recommendations made in lower positions in the ranked list of recommendations (like the first ten) are the most important. The violin plot also gives information about the upper and lower values of the distribution, as well as the first quartile, second quartile (or median) and third quartile. Outliers are also shown in the tail of the distribution [99].

---

### 5.1.2 Evaluation Method: LeaveRTOUT

The proposed *LeaveRTOut* evaluation method is inspired by *LeavePostOut*. In *LeavePostOut*, after  $P_{u,r}$  is removed,  $u$  and  $r$  are considered unconnected. But a tag  $t$  in a tag assignment of  $P_{u,r}$  may still be



**Figure 5.2:** An Example of a Violin Plot, adapted from [99]

connected to  $r$  and thus still indirectly connect the user  $u$  to the resource  $r$ . For example in Figure 5.1, tag  $t_2$  could indirectly connect user  $u_1$  to  $r$  via several hops.

An alternative evaluation method for folksonomies is thus the proposed LeaveRTOut method, which instead of eliminating the connection in the folksonomy  $F$  between user  $u$  and resource  $r$ , eliminates all connections between a tag  $t_1$  and a resource  $r$  as shown in Figure 5.3. Consequently, the resource  $r_1$  is no longer directly connected to the tag  $t_1$ . The resource  $r$  could however still be directly connected to the users  $u_1$  or  $u_2$ . The recommendation task is now to recommend the resource  $r_1$  to the user with the tag  $t_1$  as query entity.

In this thesis, results from LeavePostOut are complemented with results from LeaveRTOut in an effort to mitigate the incompleteness problem [49] explained in Section 3.2.1. LeavePostOut, on the one hand, sets a hard challenge for the recommendation task *interests match* with the user as query, as the user  $u$  in the left out post is no longer directly connected to resource  $r$ . On the other hand, LeaveRTOut sets a hard challenge for the task *guided search* with a tag as query, as all connections to the query tag  $t$  from the RT-post are removed from the folksonomy. Thus these two evaluation methods set two different cross-validation scenarios as the *test set* or left out post is determined in two different ways. In addition, the query entities are different, a user for LeavePostOut and the user's tags for LeaveRTOut.

Analogous to a post, an RT-post  $P_{r,t}$  will be defined as all tag assignments made to a specific resource  $r$  with a specific tag  $t$  as shown in Equation 5.7.

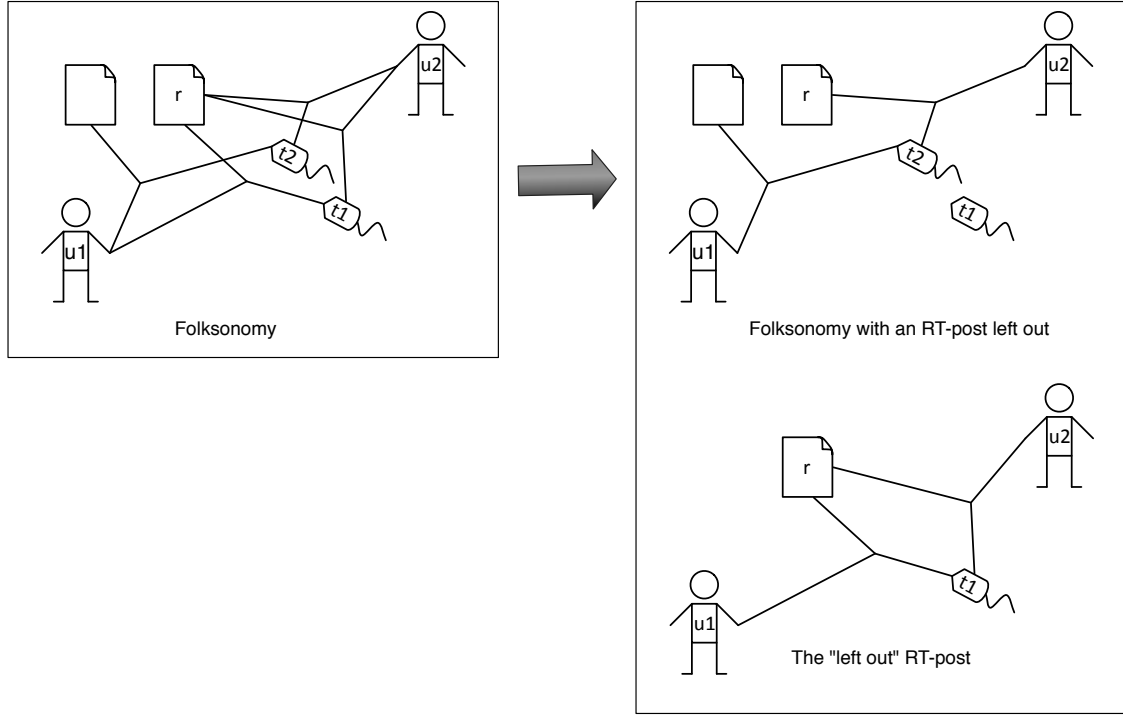
$$P_{r,t} = \{(u, t, r) | (u, t, r) \in Y\} \quad (5.7)$$

Where  $F := (U, T, R, Y)$  is a folksonomy as defined in Section 2.3 and  $u \in U, t \in T, r \in R$ .

For a folksonomy  $F = (U, T, R, Y)$  the LeaveRTOut evaluation steps for a resource recommendation task would be the following.

- For each resource  $r \in R$ , determine all RT-posts  $P_{r,t}$  between resource  $r$  and tag  $t \in T$ .
  - Remove one of these RT-posts  $P_{r,t}$  from the folksonomy  $F$  as shown in Equation 5.8.

$$F = F \setminus P_{r,t} \quad (5.8)$$



**Figure 5.3:** LeaveRTOut Evaluation Method

- Run the algorithm to be evaluated on this modified folksonomy  $F$
- Evaluate the recommendations returned by the algorithm with regard to the current resource  $r$  from the RT-post  $P_{r,t}$
- Add the removed RT-post  $P_{r,t}$  back to the folksonomy  $F$  as shown in Equation 5.9.

$$F = F \cup P_{r,t} \quad (5.9)$$

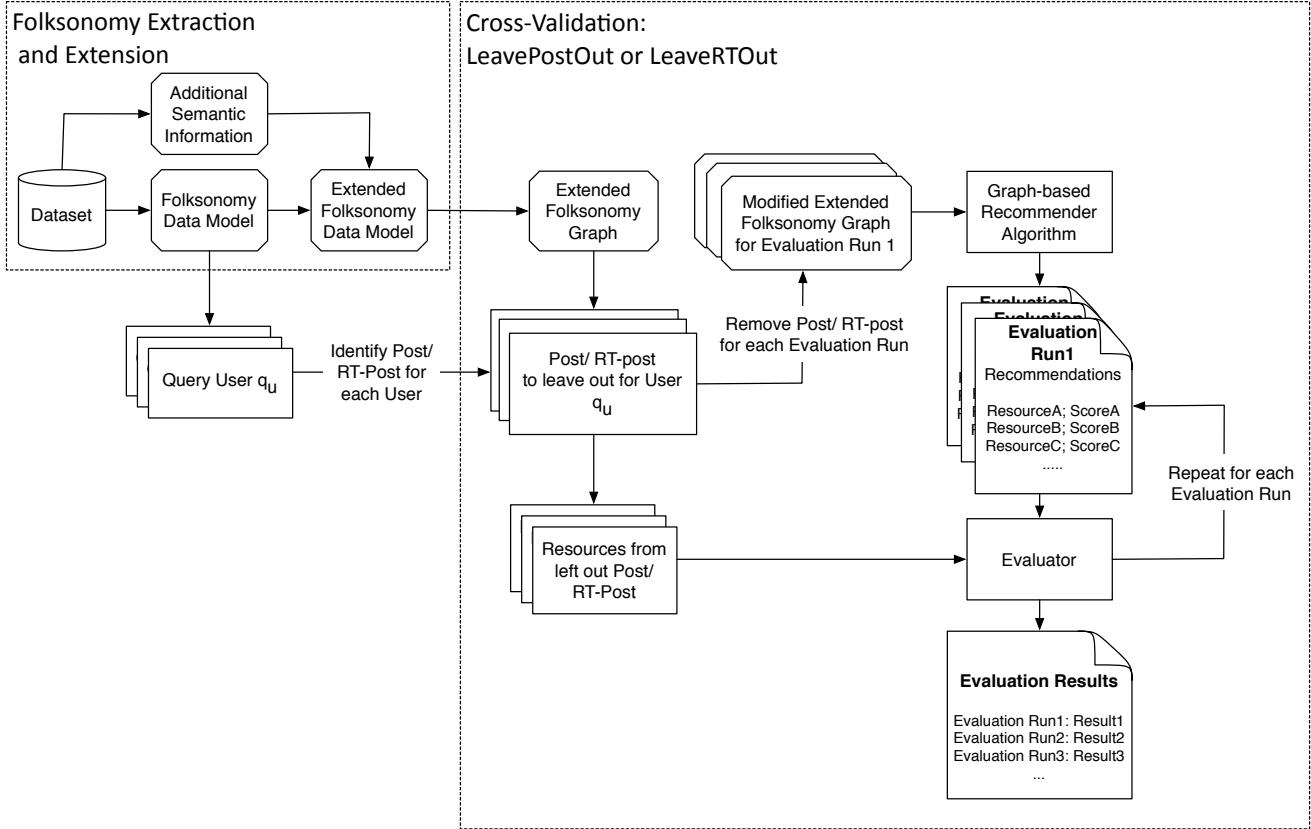
- Repeat for each RT-post of the current resource.
- Repeat for each resource.

### 5.1.3 Evaluation Metric: Mean Normalized Precision (MNP)

The evaluation metrics Mean Average Precision (MAP) and Precision at  $k$   $Precision(k)$ , as explained in Section 5.1.1, are often used to measure the recommendation accuracy of recommender algorithms. However, MAP gives an overall measure for the AP across all information queries  $Q$  and does not reflect the performance of the algorithms in the top  $k$  positions, which are crucial for a recommendation task.  $Precision(k)$  provides this focus on the top  $k$  recommended items, however it does not give an overall measure across all query entities  $Q$ . Furthermore,  $Precision(k)$  is not fitted to the evaluation methodology LeavePostOut nor LeaveRTOut where per evaluation run, only a single resource can be relevant to the query and hence the maximal achievable precision per evaluation run (where the first resource recommended is the relevant one) is  $Precision(k) = 1/k$ . This does not reflect the maximal performance of the algorithm in such a scenario as this should be 1.

Thus the new evaluation metric Mean Normalized Precision (MNP) is proposed as defined in Equation 5.10. MNP extends Precision at  $k$   $Precision(k)$  to obtain a single measure over a number of query entities  $Q$ , relative to the maximal achievable precision  $Precision_{max}(k)$ .

$$MNP(Q, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{Precision_j(k)}{Precision_{max,j}(k)} \quad (5.10)$$



**Figure 5.4:** Evaluation Approach for Hybrid Graph-based Recommender Algorithms for Folksonomies

For example, for a recommendation task where only the top ten recommendations  $k = 10$  are considered, the maximal achievable precision would be  $Precision_{max}(10) = 1/10$ . If the first resource recommended is the relevant one, then  $Precision(10) = 1/10$  and the MNP would reflect this performance by giving a  $MNP(10) = 1$ .

#### 5.1.4 Evaluation Approach and Evaluation Datasets

An approach for an offline experiment to evaluate hybrid recommender algorithms for folksonomies is shown in Figure 5.4. The folksonomy is extracted from the database and extended with additional semantic information thereby creating an extended folksonomy. An extended folksonomy graph is created from this extended folksonomy. The user query entities are extracted from the folksonomy; these could be all the users in the folksonomy. LeavePostOut or LeaveRTOut cross-validation is applied to leave out a particular post or RT-post for one evaluation run for a certain user as query entity  $q_u$ . For each user query entity  $q_u$ , all posts or RT-posts are identified and removed from the extended folksonomy graph thereby creating several modified extended folksonomy graphs for each left out post or RT-post. The graph-based recommender algorithm receives for each evaluation run a modified extended folksonomy graph with a left out post or RT-post. A query user may have several evaluation runs depending on the number of posts or RT-posts identified for the user. The graph-based recommender algorithm generates recommendations for the current query user  $q_u$  from the current modified folksonomy graph for the current evaluation run. For each evaluation run, the evaluator receives a list of resource recommendations with scores for each resource recommended. The evaluator determines the evaluation results based on this ranked list of recommendations and the left out resource for this evaluation run.

The recommendation task for the evaluations in the following sections is an *Interests Match* resource recommendation task with a user as query entity. Each user  $u$  in the folksonomy is taken as query entity

---

$q_u \in Q_u$  for the evaluation methods LeavePostOut and LeaveRTOut. For LeavePostOut, a single post is left out for each evaluation run, thus the query entity for an evaluation run is the user of the post. For LeaveRTOut, for each evaluation run, all connections between a single resource and all its tags are left out. Thus the query entity for an evaluation run is one of the users involved in the RT-post left out i.e. connected to the resource.

The parameters used for the evaluations are determined in Appendix B. The datasets used for the evaluation are described below. Each dataset has specific characteristics that make it possible to evaluate a selection of the proposed recommender algorithms. In addition, the evaluation on the different datasets gives added insights into the performance of the various recommender algorithms depending on the evaluation scenario provided by the datasets.

---

### The BibSonomy Dataset

---

The BibSonomy dataset<sup>1</sup> [132] is a folksonomy dataset from the social tagging application called BibSonomy. BibSonomy is a publication management system where scientific publications are tagged and web pages bookmarked by many users. The BibSonomy dataset was chosen as it was large enough and had enough tag assignments to be prepared as a dense folksonomy with a p-core extraction. This thus poses an evaluation scenario where the tag assignments play a central role. Therefore it would be insightful to evaluate how the algorithms AspectScore, IntelliScore and VSScore, that exploit information gained from tags and tag assignments, perform on such a dataset. Additionally, the BibSonomy dataset contains a lot of tag assignments labeled with the tag type *Topic*, thus providing a relevant evaluation scenario for AspectScore which exploits semantic tag types. To create the dataset for the evaluation, tag assignments of both publications and web pages were selected iteratively in temporal order, beginning with the oldest time stamp. For the evaluations, the dense portion of the dataset was extracted using a p-core extraction explained below.

#### P-Core Extraction

For the evaluations using the BibSonomy dataset, a p-core of level five  $l = 5$  is extracted. A p-core of level  $l$  means that an evaluation dataset comprises only of entities belonging to at least  $l$  posts [111]. A p-core extraction helps to focus on the dense part of the folksonomy and to reduce noise in a dataset. The characteristics of the BibSonomy evaluation dataset after p-core extraction are shown in Table 5.2.

#### Manually Labeling Tag Types

As the BibSonomy dataset does not have tag types, the tag assignments were manually labeled, by three researchers and a student, with the tag types required for evaluating AspectScore [202]. Table 5.1 shows the resulting distribution of tag types in the dataset. More details on the manual labeling of tag types can be found in [202].

---

### The CROKODIL Dataset

---

The CROKODIL dataset is an extraction (dated 18.06.2014) from the CROKODIL application, which is an implementation of the RBL application scenario of this thesis as presented in Section 2.4. The dataset contains activities, activity hierarchies and semantic tag types. It has the characteristics shown in Table 5.2. Furthermore, the extended folksonomy graph consists of 594 activity assignments and 221 links between activities and the users owning or working on them. The activity hierarchies introduce a further 150 links to the folksonomy. On average the activity hierarchies had a depth of 3 and a maximum fan-out of 13 with a total of 50 activities having sub-activities. The distribution of the tag types of tags

---

<sup>1</sup> Knowledge and Data Engineering Group, University of Kassel: Benchmark Folksonomy Data from BibSonomy, version of July 7th, 2011



Dataset	Topic	Genre	Person	Event	Location	Other
BibSonomy	2225	198	143	182	0	521
CROKODIL	349	103	162	25	16	652

**Table 5.1:** Distribution of Tag Types of Tags in Tag Assignments

Dataset	Users	Tags	Resources	Posts	Tag Assignments	Groups	Activities	p-core level
BibSonomy	69	179	143	1010	3269			5
CROKODIL	59	781	555	463	1307		239	1
GroupMe!	649	2580	1789	1865	4366	1143		1

**Table 5.2:** Evaluation Datasets

involved in tag assignments are shown in Table 5.1. Topic tags were the most often used tag type, followed by Person, Genre, Event and Location. The majority of tags in tag assignments had no tag types.

The CROKODIL dataset is the most representative evaluation dataset for the recommender algorithms AspectScore, AScore and AInheritScore proposed in this thesis as they were designed to exploit the activity structure and semantic tag types available in this particular RBL application scenario of which CROKODIL is a concrete implementation.

---

### The GroupMe! Dataset

---

The GroupMe! dataset is an extended folksonomy containing similar concepts to those of activities in CROKODIL. The GroupMe! dataset was collected from the GroupMe! social tagging system [2] presented in Section 2.3.2. The GroupMe! dataset has the characteristics described in Table 5.2. The concept of groups in GroupMe! is a similar concept to the activities and activity hierarchies in the CROKODIL scenario. There are however differences and a mapping of the concepts is necessary to be able to use the dataset for evaluations:

- The aim of groups in GroupMe! is to provide a collection of related resources [2]. In CROKODIL however, activities are based on a pedagogical concept to help learners structure their learning goals in a hierarchical structure. Learning resources needed to achieve these goals are attached to these activities. Therefore, the assignment of a resource to a group in GroupMe! is interpreted as attaching a resource to an activity in CROKODIL.
- Groups in GroupMe! are considered resources and can therefore belong to other groups [2]. These groups of groups or hierarchies of groups are interpreted as activity hierarchies in CROKODIL.
- Tags can be assigned to groups in GroupMe! [2]. In contrast however, tags cannot be assigned to activities in CROKODIL. These tags on groups in GroupMe! are therefore not considered in the dataset.

The GroupMe! dataset was chosen for evaluating AScore and AInheritScore as it contains structures similar to activities and activity hierarchies which are rarely found in existing folksonomy datasets. Furthermore, in comparison to CROKODIL, the GroupMe! dataset is much larger and contains much more users, tags, resources, posts, tag assignments and groups. GroupMe! thus provides a more extensive evaluation scenario for the hybrid recommender approaches requiring additional semantic information gained from activities and activity hierarchies.



---

## Baseline Algorithms

---

The recommender algorithms FolkRank, GFolkRank, GRank (all presented in Section 3.1.2) and *Most Popular* (described below) are used as baselines for the comparison of results from AScore, AInheritScore, AspectScore, IntelliScore and VSScore. For the evaluations, FolkRank has been chosen as baseline as it is the state-of-art in graph-based recommender systems for folksonomies [1, 161, 192]. Furthermore, most of the approaches are extensions of FolkRank and thus a direct comparison gives a strong verification of results. As AScore and AInheritScore are inspired by GFolkRank and GRank respectively, it is appropriate to compare their results to their respective baselines. Results are also compared to Most Popular in order to have a comparison to a simple recommender algorithm based on a very different concept than the hybrid graph-based algorithms.

**Most Popular** recommends resources based on the number of tags and users a resource is connected to as this is taken as an indication of how popular the resource is in the folksonomy [202]. The score of a resource is independent of the query as it is simply computed as shown in Equation 5.11.

$$score(r) = |T(r)| + |U(r)| \quad (5.11)$$

---

## 5.2 Evaluation Results

---

The overall aim of the evaluations is to show that the proposed hybrid recommender algorithms perform better than the specified baseline recommender algorithms, thereby showing that the exploitation of additional semantic information is beneficial to providing improved recommendations of learning resources.

The positions of the ranked recommendations are plotted as violin plots in order to show the distribution of the relevant recommendations. The Mean Average Precision (MAP) is calculated over all user query entities. The Mean Normalized Precision (MNP) is calculated for the top ten recommendation positions over all user query entities. Significance tests are conducted to determine the overall effectiveness of the algorithms based on the average precision results of LeavePostOut and LeaveRTOut, which measures the overall ranking effectiveness achieved for each user query entity.

Each evaluation run has a unique evaluation ID and each evaluation run has a unique user as query entity. Each evaluation run is repeated for all algorithms, therefore a pairwise comparison of the average precision results on a per user basis is possible. The average precision values can however not be said to be normally distributed, therefore the Wilcoxon signed-rank test [83] (as described in Appendix B) is executed on each pair of results. The null hypothesis  $H_0$  states that there will be no difference in the evaluation results between the two algorithms being compared pairwise. Hence, the alternative hypothesis  $H_1$  is that there is a difference in the evaluation results of one algorithm compared to the other algorithm. One exception however are comparisons with AspectScore. As AspectScore disambiguates tags, the number of evaluation runs is different from those of the other algorithms when evaluated with LeaveRTOut. In these cases, the comparison cannot be made pairwise as the number of evaluation runs differ from those of the other algorithms and therefore the Wilcoxon rank-sum test is conducted.

---

### 5.2.1 Results of AspectScore, IntelliScore and VSScore on the BibSonomy Dataset

---

For the evaluations, AspectScore, IntelliScore and VSScore are compared with FolkRank and Most Popular on the BibSonomy dataset. As mentioned above, BibSonomy is chosen as evaluation dataset as it offers a dense folksonomy having a p-core level of 5, meaning each resource in the dataset is involved in at least 5 tag assignments. Additionally, the BibSonomy dataset offers a lot of tag types, in particular tags of type *Topic* required by AspectScore.

---

## BibSonomy - Distribution of Ranked Recommendations

---

Figure 5.5 shows the results of positions in the ranked list of recommendations where relevant resources are found. Along the y-axis, the distribution of relevant recommendations are shown at the positions in the ranked list of recommendations, where position 0 means the top-most ranked recommendation. The width of the violin plot is proportional to the estimated density at that point. As can be seen, for LeavePostOut, most of the algorithms have the majority of resources ranked in positions  $< 40$  and for LeaveRTOut  $< 30$ , whereas Most Popular still has many resources ranked in higher positions. Table B.3 in Appendix B shows the descriptive statistics for the distribution of recommendation positions in a ranked list of recommendations for LeavePostOut and LeaveRTOut.

For each algorithm, a total of 1010 evaluation runs were executed for LeavePostOut and 3269 evaluation runs for LeaveRTOut. For LeavePostOut, Most Popular has the worst distribution of relevant recommendations with a mean position of 55, whereas VSScore, AspectScore and FolkRank have good distributions with mean positions of 30, 36 and 38 respectively. For LeaveRTOut, Most Popular also has the worst distribution of relevant recommendations having a mean position of 51. AspectScore, VSScore, FolkRank and IntelliScore have similarly good distributions with mean positions of 22, 23, 23 and 25 respectively.

---

## BibSonomy Evaluation Results: Mean Normalized Precision (MNP)

---

The Mean Normalized Precision (MNP) evaluation results of the top ten positions in the ranked list of recommendations are plotted in Figure 5.6 and shown in Table B.4 in Appendix B. For LeavePostOut, Most Popular performs worst with a MNP between 0.042 and 0.180. AspectScore and VSScore perform best with a MNP between 0.111 and 0.341 and between 0.106 and 0.365 respectively. For LeaveRTOut, Most Popular again performs worst with a MNP between 0.039 and 0.188. The MNP results of AspectScore, VSScore, IntelliScore and FolkRank are comparable.

---

## BibSonomy Evaluation Results: Mean Average Precision (MAP)

---

Figure 5.7 and Table B.5 in Appendix B show the Mean Average Precision (MAP) results of LeavePostOut and LeaveRTOut. For LeavePostOut, VSScore with a MAP of 0.197 performs best, closely followed by AspectScore with a MAP of 0.194 and FolkRank with a MAP of 0.181. IntelliScore with a MAP of 0.150 is only better than Most Popular with a MAP of 0.094. For LeaveRTOut, Most Popular again has the lowest MAP (0.099). FolkRank with a MAP of 0.203 performs best, followed by AspectScore, IntelliScore and VSScore with a MAP of 0.199, 0.198 and 0.182 respectively.

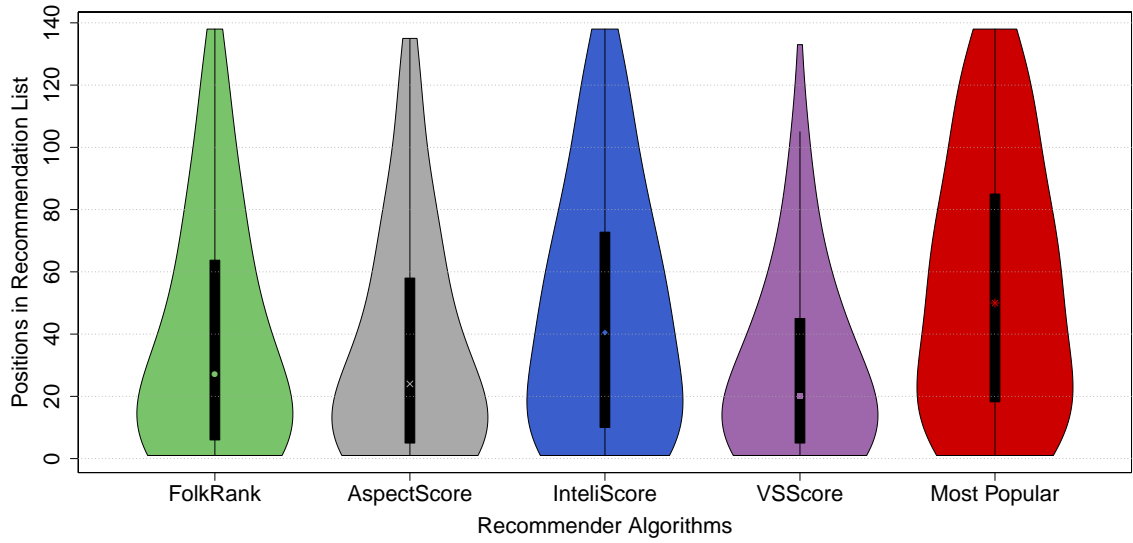
---

## BibSonomy Results of Statistical Significance Tests

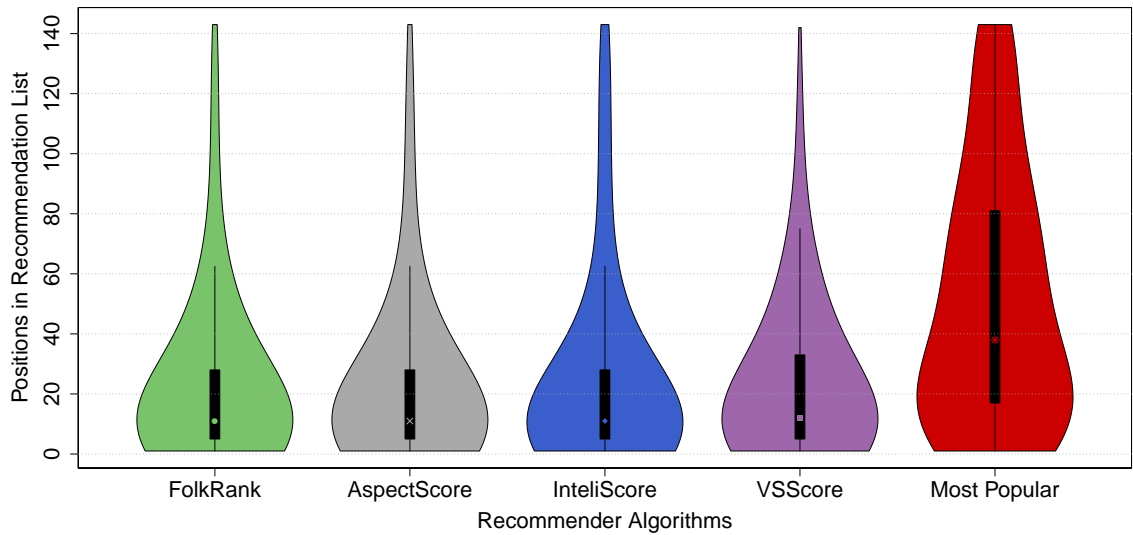
---

Statistical significance tests are performed on the average precision values from each algorithm. A summary of descriptive statistics for the average precision values is shown in Table B.6 in Appendix B. A total of 3269 evaluation runs were executed for LeavePostOut and 1010 for LeaveRTOut. The Wilcoxon signed-rank test [83] is executed on each pair of results. For comparisons with AspectScore, the Wilcoxon rank-sum test is conducted. The summarized results of all pairwise comparisons are shown in Table 5.3. For LeavePostOut, 3269 pairwise comparisons per test (3136 for comparisons with AspectScore), and for LeaveRTOut, 1010 pairwise comparisons per test were conducted. The detailed results of the significance tests stating p-values are shown in Table B.7 in Appendix B. The significance level of  $p < 0.05$  is chosen (as described in Appendix B).

For LeavePostOut, VSScore is significantly more effective than all other algorithms. AspectScore is significantly more effective than IntelliScore and Most Popular. From Table B.7 in Appendix B, FolkRank



(a) BibSonomy - Distribution of Ranked Recommendations for LeavePostOut plotted as Violin Plots

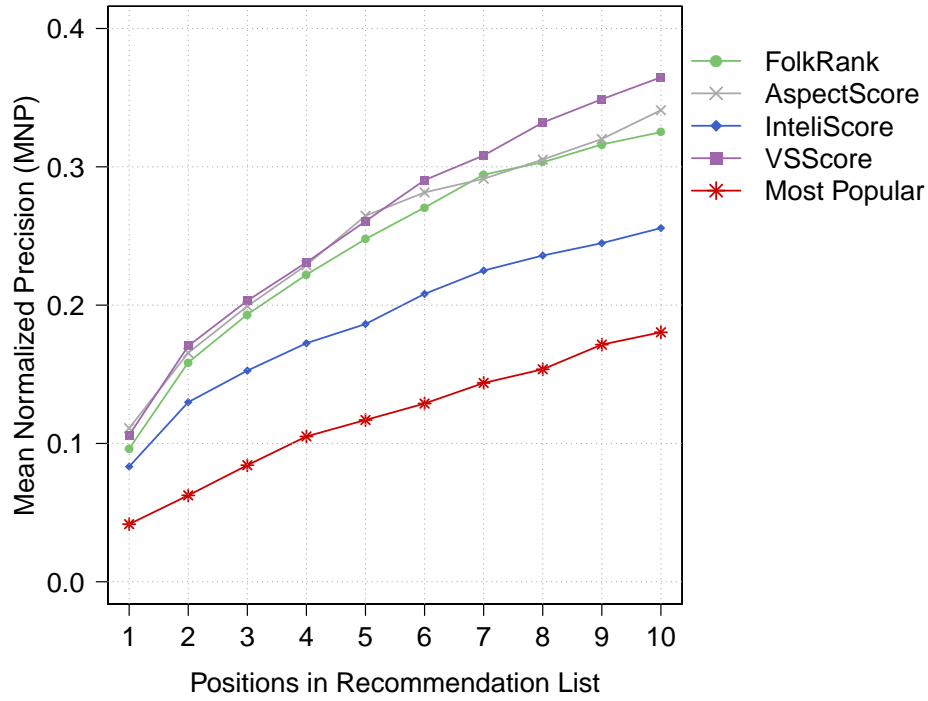


(b) BibSonomy - Distribution of Ranked Recommendations for LeaveRTOut plotted as Violin Plots

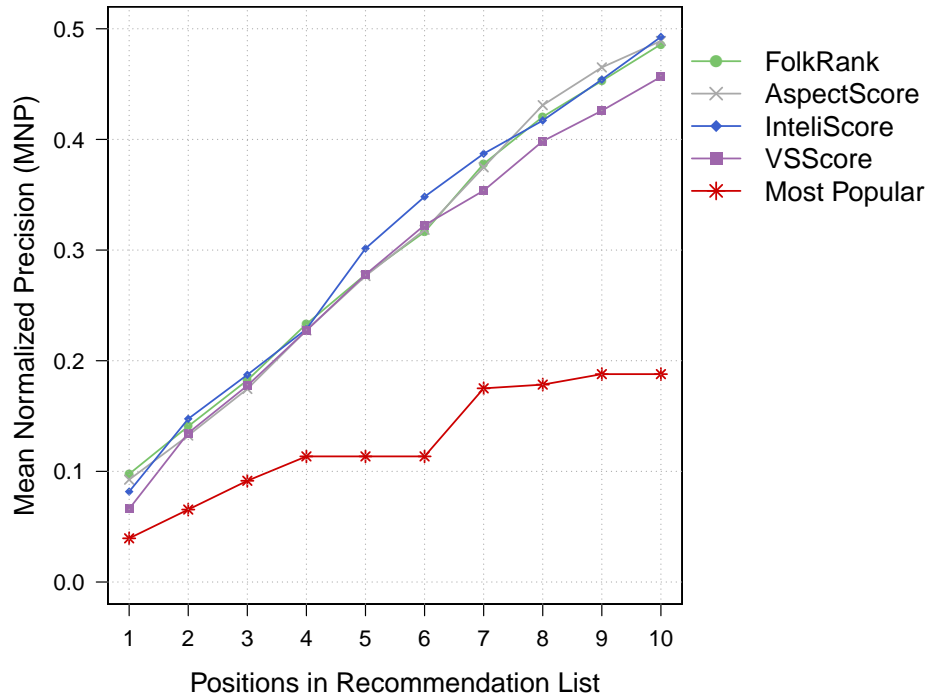
**Figure 5.5:** BibSonomy - Distribution of Ranked Recommendations for LeavePostOut and LeaveRTOut

LeavePostOut					
More effective than →	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
AspectScore	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
InteliScore	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
VSScore	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
LeaveRTOut					
More effective than →	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
AspectScore	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
InteliScore	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VSScore	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Table 5.3:** BibSonomy - Results of Statistical Significance Tests for LeavePostOut and LeaveRTOut

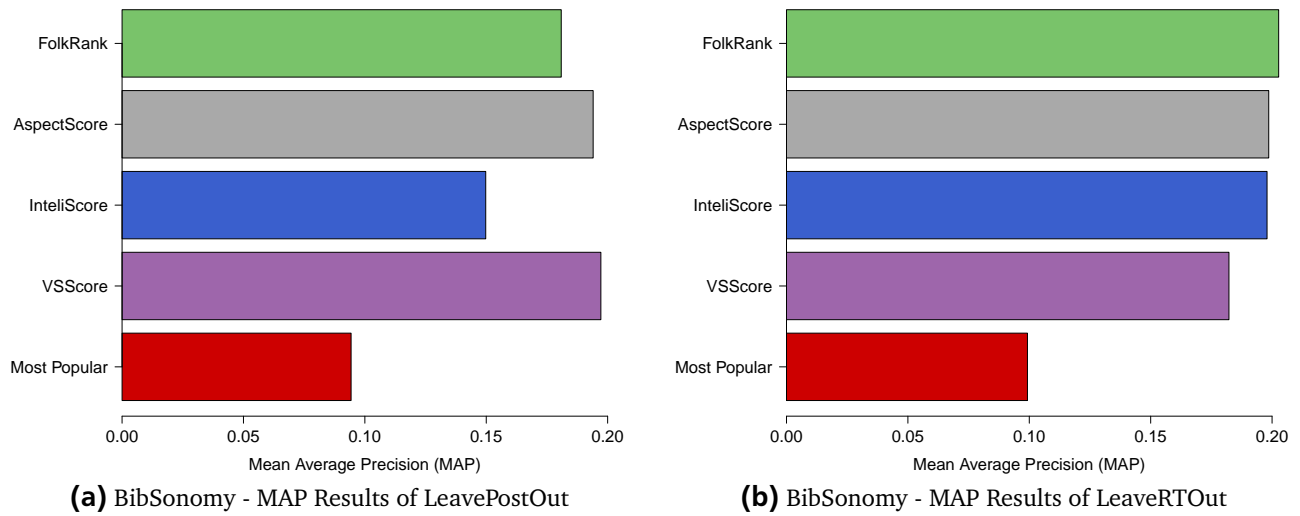


(a) BibSonomy - MNP Results of LeavePostOut



(b) BibSonomy - MNP Results of LeaveRTOut

**Figure 5.6:** BibSonomy - Mean Normalized Precision (MNP) Results of LeavePostOut and LeaveRTOut



**Figure 5.7:** BibSonomy - Mean Average Precision (MAP) Results of LeavePostOut and LeaveRTOut

is however not significantly more effective than AspectScore. InteliScore is only more effective than Most Popular. For LeaveRTOut, AspectScore, FolkRank and InteliScore are all significantly more effective than VSScore and Most Popular. FolkRank is however neither significantly more effective than AspectScore nor significantly more effective than InteliScore.

## Discussion

For LeavePostOut, MNP results show that VSScore performs best in the top positions. VSScore also achieves the highest MAP and is significantly more effective than all other algorithms regarding AP. In a dense folksonomy, a lot of information connecting the user to a resource is lost when in LeavePostOut all tag assignments in a post between the resource and the user are removed. VSScore has an added advantage over the other algorithms as it exploits the context-information from the dense folksonomy in finding a connection between the query user and the resource in the left out post. For LeavePostOut, AspectScore performs nearly as good as VSScore with regards to MAP and MNP, however AspectScore is not significantly more effective than VSScore nor FolkRank when comparing AP values. For LeavePostOut, InteliScore performs worse compared to FolkRank, AspectScore and VSScore. This is probably due to the difficulty in determining the semantic relatedness of tags [203]. For about 27% of the tags in the dataset, no semantic relatedness could be determined. In InteliScore, XESA [222] is used to calculate the semantic relatedness between pairs of tags using the English Wikipedia. Thus stop words or numbers e.g. 2006 cannot be considered and some tags do not exist in Wikipedia e.g. itegpub. In these cases, the semantic relatedness is taken as 0.0.

For LeaveRTOut, FolkRank, AspectScore, InteliScore and VSScore perform comparably across evaluation metrics. All have a MAP of approximately 0.2, however FolkRank, AspectScore and InteliScore are significantly more effective than VSScore regarding AP. FolkRank is neither significantly more effective than AspectScore nor InteliScore. MNP results also show comparable performance in the top positions. LeaveRTOut removes an RT-post, thereby all tag assignments between a resource and a tag are removed. However in a dense folksonomy, the resource and tag are most likely still connected via other tag assignments in the folksonomy as they must be involved in at least 5 tag assignments (p-core level 5). Therefore the dense folksonomy itself seems to be sufficient in providing enough information to be able to recommend the left out resource to the query tag. Thus the inclusion of additional semantic information does not seem to be advantageous in a dense folksonomy for this evaluation method.

	Baseline	Metric	LeavePostOut	LeaveRTOut
AspectScore	Most Popular	MNP	☒	☒
		MAP	☒	☒
		AP	☒	☒
	FolkRank	MNP	☒	(☒)
		MAP	☒	(☒)
		AP	(□)	(□)
InteliScore	Most Popular	MNP	☒	☒
		MAP	☒	☒
		AP	☒	☒
	FolkRank	MNP	□	(☒)
		MAP	□	(☒)
		AP	□	(□)
VSScore	MostPopular	MNP	☒	☒
		MAP	☒	☒
		AP	☒	☒
	FolkRank	MNP	☒	□
		MAP	☒	□
		AP	☒	□

**Table 5.4:** BibSonomy - Summary of Evaluation Results of AspectScore, InteliScore and VSScore

Most Popular performed worst across evaluation methods and evaluation metrics due to the fact that unlike the other graph-based algorithms, Most Popular does not take advantage of the increased interconnections in the dense folksonomy. Thus recommending popular resources to a user in a dense folksonomy is not effective.

### Limitations

Evaluations on the BibSonomy dataset are limited by the fact that semantic tag types required by AspectScore do not exist in the BibSonomy dataset. To enable an evaluation on the dense folksonomy offered by the BibSonomy dataset, the tag assignments in the dataset had to be labeled manually. Manual labelling is a challenge, as only the user who originally created and assigned a tag to a resource really knows the true meaning of the tag assignment and thus the semantic tag type relevant to the tag assignment. Furthermore, the Cohen’s  $d$  effect sizes in Table B.8 in Appendix B show that the significance test results based on the differences in AP show overall a small effect size ( $d < 0.4$ ), this indicates that the evaluation scenario on a dense folksonomy is not optimal as an evaluation scenario to compare the proposed recommender algorithms.

### Conclusion

The evaluation results are summarized in Table 5.4. Overall, the evaluation results show that an algorithm may perform better or worse depending on the evaluation method and evaluation metric applied. The LeavePostOut results differ from the LeaveRTOut results due to the fact that they set a differently hard task to solve. Hence, the results from the two evaluation methods are useful to assess the effectiveness of the algorithms in different recommendation scenarios. For example, results from LeavePostOut show on the one hand, that VSScore is more effective than AspectScore, FolkRank and InteliScore. On the other hand, results from LeaveRTOut show that AspectScore, FolkRank and InteliScore are more effective than VSScore.

For the evaluation scenario with the dense BibSonomy folksonomy, the exploitation of the context-information by VSScore, as well as the disambiguation of tags based on semantic tag types in AspectScore have shown to be effective when evaluated with the LeavePostOut evaluation method. In contrast, the exploitation of additional information in a dense folksonomy does not seem to have much added advantage for the evaluation method LeaveRTOut.

---

### 5.2.2 Results of AScore and AInheritScore on the GroupMe! Dataset

---

The performance of AScore and AInheritScore are evaluated on the GroupMe! dataset as both recommender algorithms require hierarchical activity structures similar to the group structure in GroupMe!. AScore and AInheritScore are compared with FolkRank, GFolkRank, GRank and Most Popular.

---

#### GroupMe! - Distribution of Ranked Recommendations

---

Figure 5.8 shows the ranked recommendation results of AScore and AInheritScore as violin plots. Table B.9 in Appendix B shows the descriptive statistics for the distribution of recommendation positions in a ranked list of recommendations for LeavePostOut and LeaveRTOut. A total of 1865 evaluation runs were executed for each algorithm for LeavePostOut, therefore for each run, a single resource was determined as relevant with a certain ranking or position in the ranked list of recommendations. GFolkRank and AScore had the best distribution of ranked recommendations with a mean position of 12 and 14 respectively. Most Popular had the worst distribution with a mean of 1610. For LeaveRTOut, 4332 evaluation runs were executed and thus 4332 recommendations were ranked. Again GFolkRank and AScore had the best distribution of ranked recommendations, both having a mean position of 66. The worst distribution was again Most Popular with a mean position of 836.

---

#### GroupMe! Evaluation Results: Mean Normalized Precision (MNP)

---

The results of the Mean Normalized Precision (MNP) for the top ten positions in the ranked list of recommendations for LeavePostOut and LeaveRTOut are plotted in Figure 5.9. The details are shown in Table B.10 in Appendix B. For LeavePostOut, GFolkRank performs best with a MNP in the top ten positions between 0.538 and 0.918. AScore follows with a MNP between 0.600 and 0.894 and then AInheritScore with MNP between 0.385 and 0.588. Most Popular performs worst with a MNP of 0.0. For LeaveRTOut, GFolkRank and AScore contend for the top ten MNP values. FolkRank follows and then GRank and AInheritScore. Most Popular performs worst once more.

---

#### GroupMe! Evaluation Results: Mean Average Precision (MAP)

---

Figure 5.10 and Table B.11 in Appendix B show the Mean Average Precision (MAP) results of LeavePostOut and LeaveRTOut. GFolkRank and AScore perform best for both LeavePostOut and LeaveRTOut and Most Popular performs worst. For LeavePostOut, AInheritScore with a MAP of 0.473 performs better than GRank with a MAP of 0.378. For LeaveRTOut however, GRank performs better with a MAP of 0.144 than AInheritScore with a MAP of 0.107.

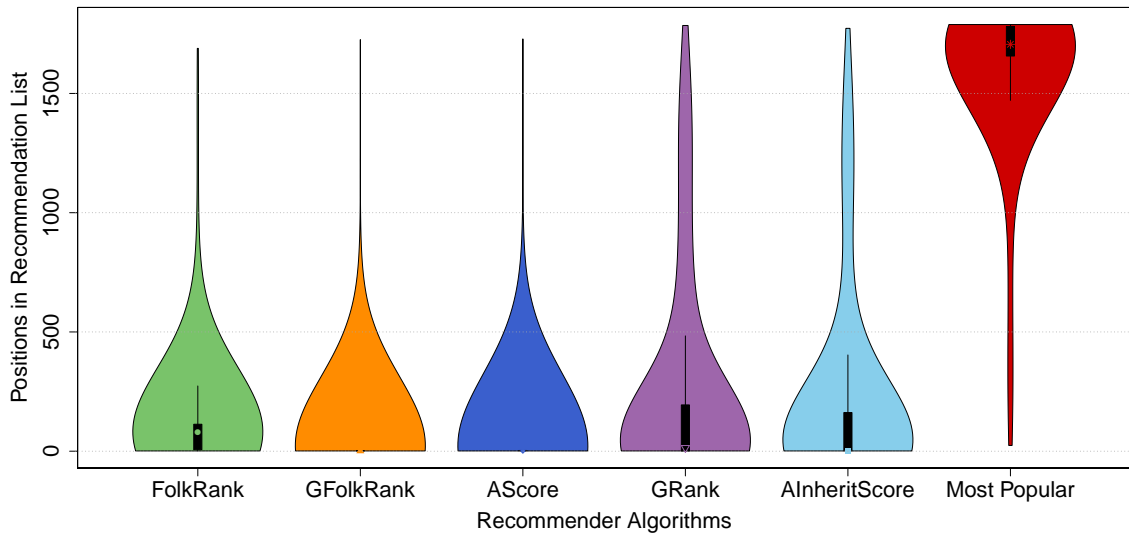
---

#### GroupMe! Results of Statistical Significance Tests

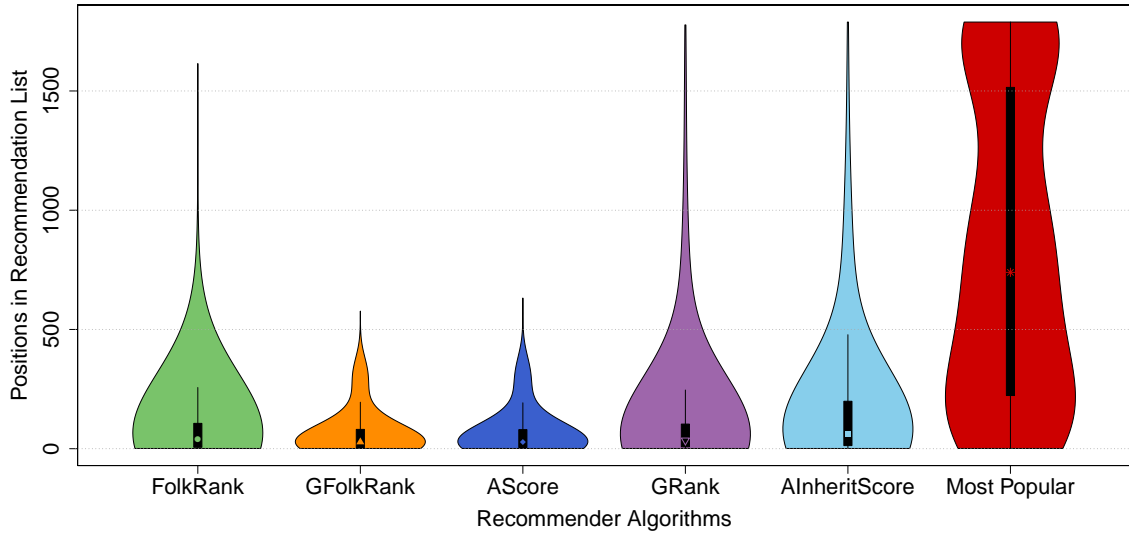
---

The average precision results of LeavePostOut and LeaveRTOut, are used for statistical significance testing. A summary of descriptive statistics for the average precision values is shown in Table B.12 in Appendix B. A total of 1865 evaluation runs were executed for LeavePostOut and 4332 for LeaveRTOut. The summarized results are shown in Table 5.5 for LeavePostOut, with 1865 pairwise comparisons per test, and for LeaveRTOut, with 4332 pairwise comparisons per test. The detailed results of the significance tests stating p-values are shown in Table B.13 in Appendix B. The threshold value of  $p < 0.05$  is applied. For LeavePostOut, AScore performs significantly better than FolkRank, GRank, AInheritScore and Most Popular regarding AP. GFolkRank is however significantly more effective than AScore. AInheritScore is significantly more effective than FolkRank, GRank and Most Popular.





(a) GroupMe! - Distribution of Ranked Recommendations for LeavePostOut plotted as Violin Plots



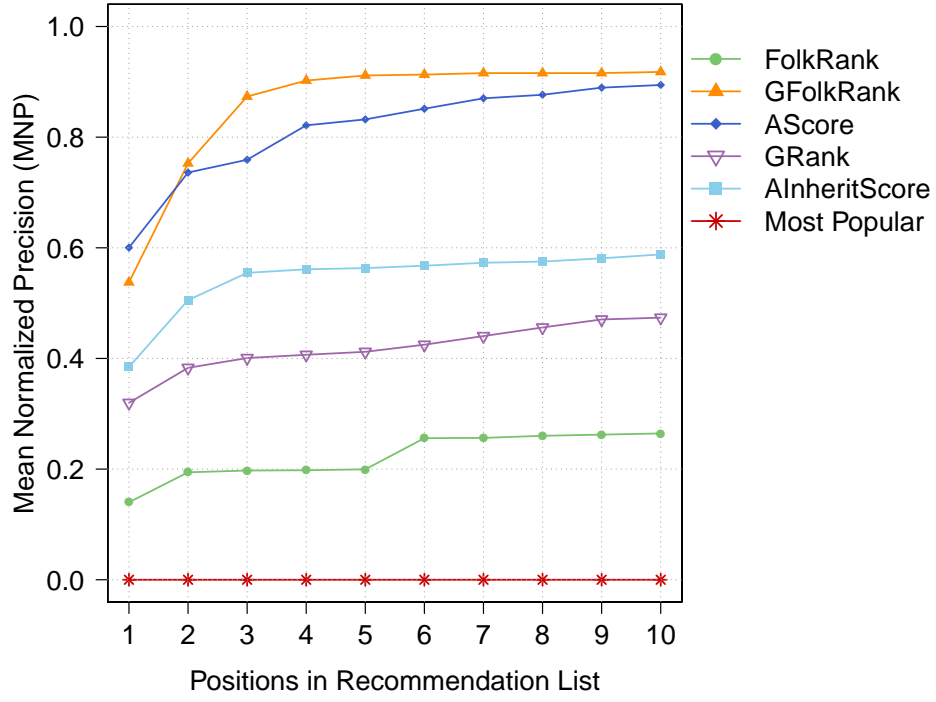
(b) GroupMe! - Distribution of Ranked Recommendations for LeaveRTOut plotted as Violin Plots

**Figure 5.8:** GroupMe! - Distribution of Recommendations for LeavePostOut and LeaveRTOut

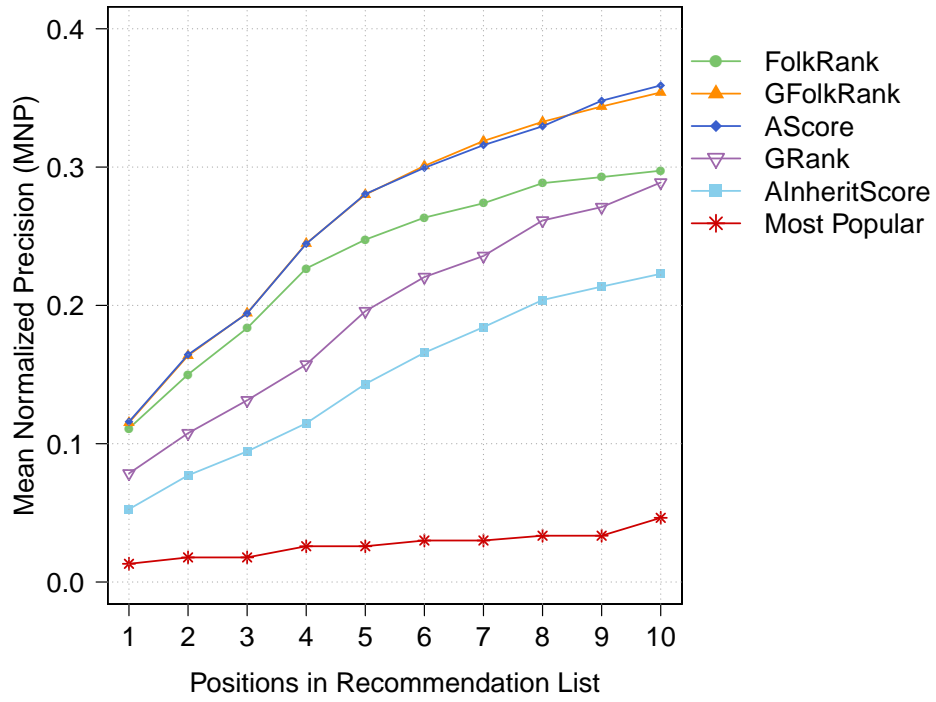
LeavePostOut						
More effective than →	FolkRank	GFolkRank	AScore	GRank	AINheritScore	Most Popular
AScore	☒	☐	☐	☒	☒	☒
AINheritScore	☒	☐	☐	☒	☐	☒
LeaveRTOut						
More effective than →	FolkRank	GFolkRank	AScore	GRank	AINheritScore	Most Popular
AScore	☒	☒	☐	☒	☒	☒
AINheritScore	☐	☐	☐	☐	☐	☒

**Table 5.5:** GroupMe! - Results of Statistical Significance Tests for LeavePostOut and LeaveRTOut



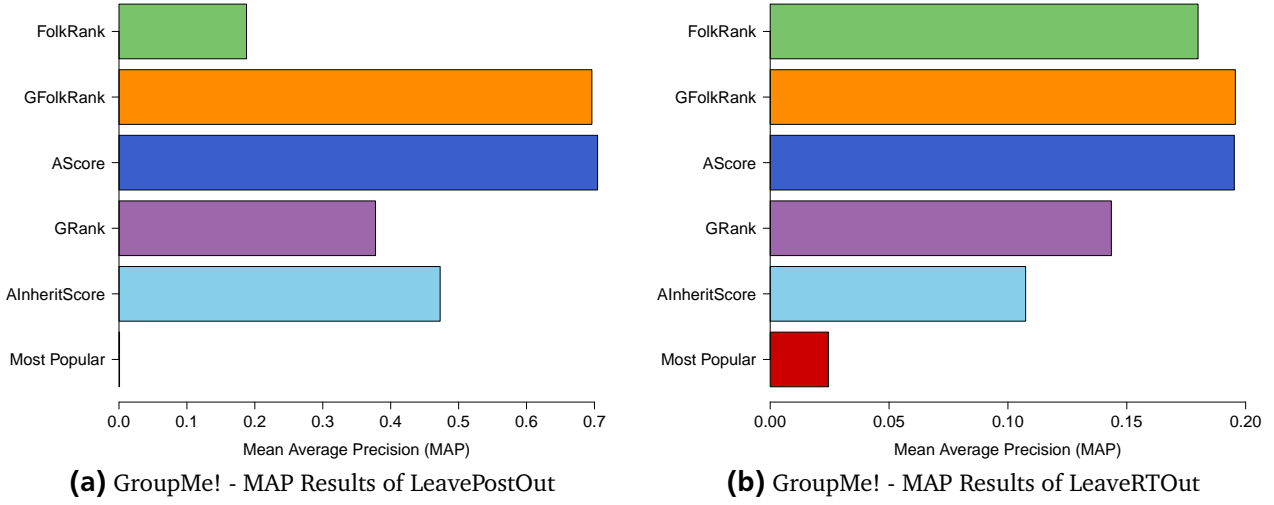


(a) GroupMe! - MNP Results of LeavePostOut



(b) GroupMe! - MNP Results of LeaveRTOut

**Figure 5.9:** GroupMe! - Mean Normalized Precision (MNP) Results of LeavePostOut and LeaveRTOut



**Figure 5.10:** GroupMe! - Mean Average Precision (MAP) Results of LeavePostOut and LeaveRTOut

For LeaveRTOut, AScore is significantly more effective than all other algorithms regarding AP. AlnheritScore is however only more effective than Most Popular.

## Discussion

For both LeavePostOut and LeaveRTOut and across all evaluation metrics, AScore performs better than the baseline algorithm FolkRank. The Cohen’s  $d$  effect size in Table B.14 in Appendix B, shows a large effect size of  $d = 1.4$  for LeavePostOut but a small effect size ( $d = 0.04$ ) for LeaveRTOut. AScore also performs better than GRank, AlnheritScore and Most Popular with an overall moderate effect size.

AScore however contends with GFolkRank across evaluation methods and metrics. The MNP values for LeavePostOut show that AScore outperforms GFolkRank in the topmost ( $k = 1$ ) position but GFolkRank is then consistently better for the remaining top positions. AScore has a higher MAP than GFolkRank for LeavePostOut, however GFolkRank is significantly more effective than AScore regarding AP but with a small effect size ( $d = 0.02$ ). For LeaveRTOut, the MNP and MAP are very similar, however AScore is significantly more effective than GFolkRank but with a small effect size ( $d = 0.001$ ).

AlnheritScore outperforms both baselines FolkRank and GRank for LeavePostOut across evaluation metrics with a moderate effect size of  $d = 0.7$  and small effect size of  $d = 0.2$  respectively for the significance tests. In contrast, for LeaveRTOut, AlnheritScore performs worse than both baselines FolkRank and GRank with regards to MNP and MAP. Significance tests also show that FolkRank and GRank are significantly more effective with regards to AP, however with a small effect size of  $d = 0.3$  and  $d = 0.1$  respectively. AlnheritScore is only more effective than Most Popular. Most Popular again performs worst across evaluation methods and evaluation metrics.

## Limitations

AScore and AlnheritScore were conceived specifically for the RBL scenario described in Section 2.4. Thus, they are fundamentally based on the exploitation of activities and activity hierarchies. The GroupMe! dataset has a comparable concept of groups and groups of groups, however there are differences (as stated in Section 5.1.4) that might have affected the evaluation of the algorithms. In addition, groups of groups or group hierarchies are unfortunately sparse in the GroupMe! dataset.

	Baseline	Metric	LeavePostOut	LeaveRTOut
AScore	Most Popular	MNP	☒	☒
		MAP	☒	☒
		AP	☒	☒
	FolkRank	MNP	☒	☒
		MAP	☒	☒
		AP	☒	☒
	GFolkRank	MNP	☐	(☐)
		MAP	☒	(☐)
		AP	(☐)	(☒)
AInheritScore	MostPopular	MNP	☒	☒
		MAP	☒	☒
		AP	☒	☒
	FolkRank	MNP	☒	☐
		MAP	☒	☐
		AP	☒	☐
	GRank	MNP	☒	☐
		MAP	☒	☐
		AP	☒	☐

**Table 5.6:** GroupMe! - Summary of Evaluation Results of AScore and AInheritScore

## Conclusion

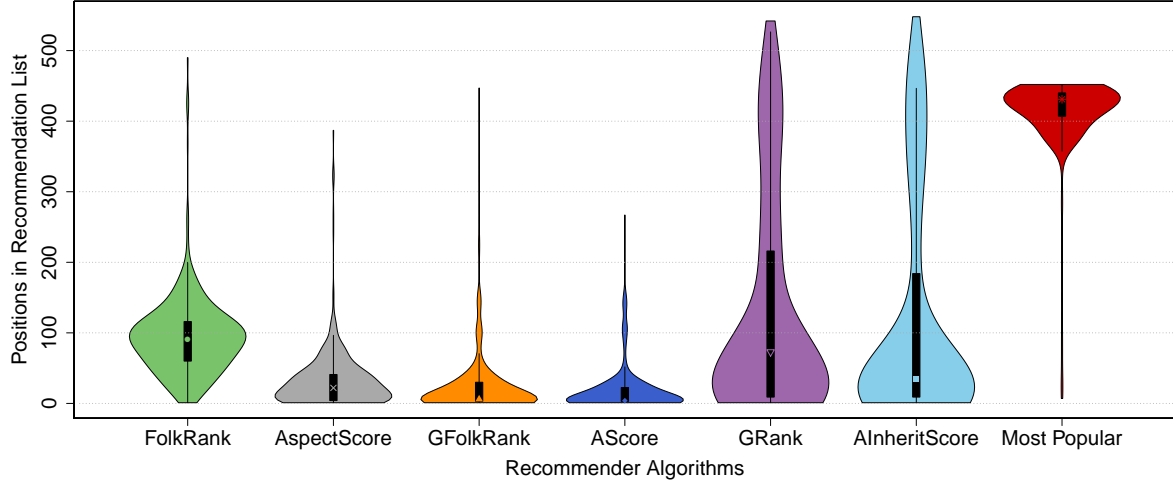
The evaluation results are summarized in Table 5.6. LeavePostOut results show that the algorithms leveraging additional semantic information are overall more effective than FolkRank and Most Popular as the hybrid algorithms have the advantage of being able to find connections between the user and the relevant resource via activities and activity hierarchies (groups and groups of groups).

For LeaveRTOut, the situation is different as the evaluation method poses a harder challenge for the algorithms GRank and AInheritScore which are based on query tags. This fact, combined with the further loss of information when all tag assignments between the query tag and a resource are removed with the RT-post, causes both algorithms to suffer in performance. AScore, GFolkRank and FolkRank do not suffer as much from this as they have the added advantage of being able to reach isolated nodes in the folksonomy via the random surfer model. Their performance is however still affected as their MAP and MNP values sink in comparison to LeavePostOut. This is due to the fact that a lot of information is lost when all tag assignments are removed between the query tag and the relevant resource in the RT-posts. The additional connections via activities and activity hierarchies (groups and groups of groups) give less of an advantage in LeaveRTOut.

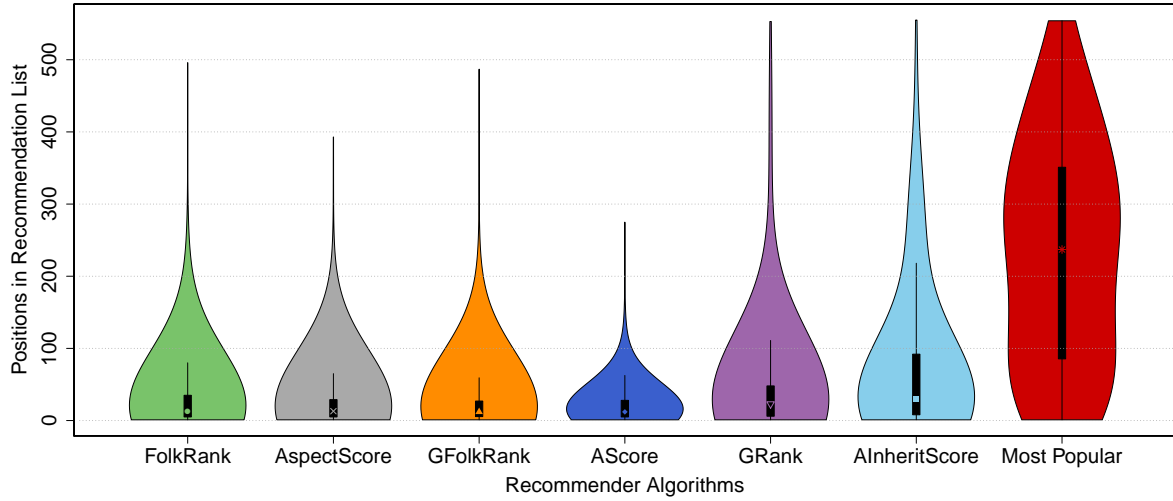
Overall for the GroupMe! dataset, the results show that although the performance depends on the evaluation method and evaluation metrics, considering additional semantic information gained from activities and activity hierarchies (groups and groups of groups) is an added advantage when recommending relevant resources to a user.

### 5.2.3 Results of AspectScore, AScore and AInheritScore on the CROKODIL Dataset

The performance of AspectScore, AScore and AInheritScore are evaluated on the CROKODIL dataset. The CROKODIL dataset offers semantic tag types for AspectScore and activities and activity hierarchies for AScore and AInheritScore. Results of AspectScore, AScore and AInheritScore are compared to FolkRank, GFolkRank, GRank and Most Popular.



(a) CROKODIL - Distribution of Ranked Recommendations for LeavePostOut plotted as Violin Plots



(b) CROKODIL - Distribution of Ranked Recommendations for LeaveRTOut plotted as Violin Plots

**Figure 5.11:** CROKODIL - Distribution of Ranked Recommendations for LeavePostOut and LeaveRTOut

---

### CROKODIL - Distribution of Ranked Recommendations

---

The positions of the ranked recommendations obtained from LeavePostOut and LeaveRTOut are shown in Figure 5.11. Table B.15 in Appendix B shows the descriptive statistics for the distribution of recommendation positions for LeavePostOut and LeaveRTOut. A total of 463 evaluation runs were executed for LeavePostOut and 1311 for LeaveRTOut. For both LeavePostOut and LeaveRTOut, AScore has the best distribution with a mean position of 19 and 21 and maximum position of 267 and 275 respectively. Most Popular performs worst with a mean position of 411 for LeavePostOut and 229 for LeaveRTOut.

---

### CROKODIL Evaluation Results: Mean Normalized Precision (MNP)

---

The MNP evaluation results of the top ten positions in the ranked list of recommendations are shown in Figure 5.12 and in Table B.16 in Appendix B for LeavePostOut and LeaveRTOut. For LeavePostOut, GFolkRank has the highest MNP at the topmost position  $k = 1$  with 0.16, however AScore achieves the highest MNP already at position 3 with a MNP of 0.4. For LeaveRTOut, GRank has the highest MNP in

LeavePostOut							
More effective than →	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritScore	Most Popular
AspectScore	☒	☐	☐	☐	☒	☒	☒
AScore	☒	☒	☒	☐	☒	☒	☒
AINheritScore	☒	☐	☐	☐	☒	☐	☒
LeaveRTOut							
More effective than →	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritScore	Most Popular
AspectScore	☐	☐	☐	☐	☒	☒	☒
AScore	☐	☐	☒	☐	☒	☒	☒
AINheritScore	☐	☐	☐	☐	☐	☐	☒

**Table 5.7:** CROKODIL - Results of Statistical Significance Tests for LeavePostOut and LeaveRTOut

the topmost position with 0.048. GFolkRank and AScore achieve the highest MNP at position 10 with a MNP of 0.467. Most Popular performs worst for both LeavePostOut and LeaveRTOut.

#### CROKODIL Evaluation Results: Mean Average Precision (MAP)

Figure 5.13 and Table B.17 in Appendix B show the MAP results of LeavePostOut and LeaveRTOut. AScore outperforms all other algorithms for LeavePostOut with a MAP of 0.304 followed by GFolkRank with a MAP of 0.291. For LeaveRTOut, AspectScore performs best with a MAP of 0.158, followed by FolkRank with a MAP of 0.153 and GFolkRank with a MAP of 0.151. AScore and AInheritScore are also very close with a MAP of 0.145 and 0.143 respectively. Most Popular has the worst MAP for both LeavePostOut and LeaveRTOut.

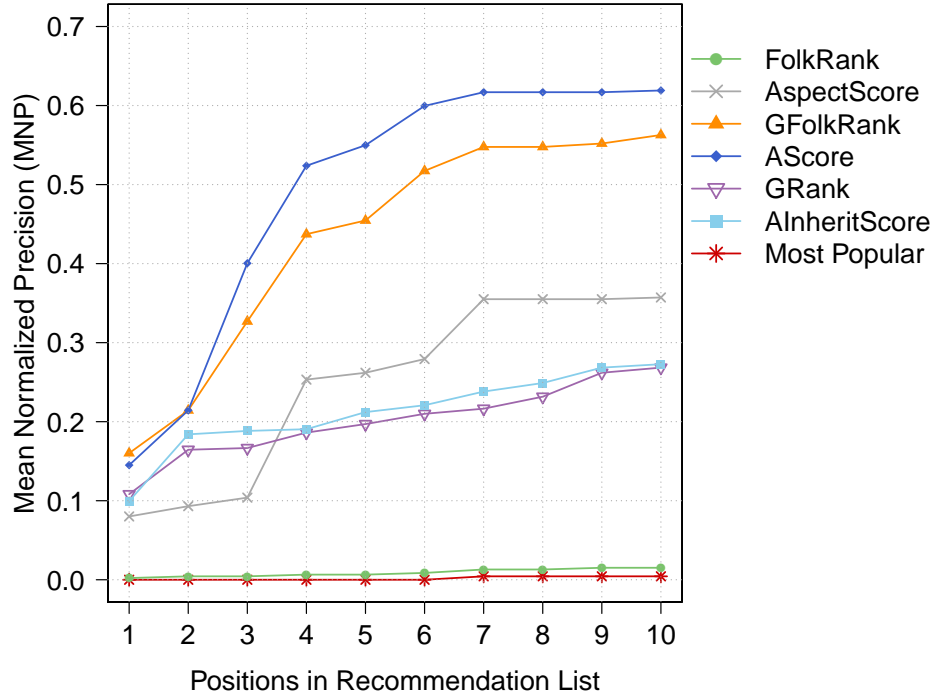
#### CROKODIL Results of Statistical Significance Tests

A summary of descriptive statistics for the average precision values is shown in Table B.18 in Appendix B. A total of 463 evaluation runs are considered for LeavePostOut and 1311 for LeaveRTOut. The summarized results are presented in Table 5.7. Detailed results of the significance tests stating p-values are shown in Table B.19 in Appendix B. For LeavePostOut, AScore is significantly more effective than all other algorithms but with varying effect sizes as shown in Table B.20 in Appendix B. AspectScore and AInheritScore are both significantly more effective than GRank, however AspectScore is more effective than AInheritScore. All algorithms are significantly more effective than FolkRank and Most Popular.

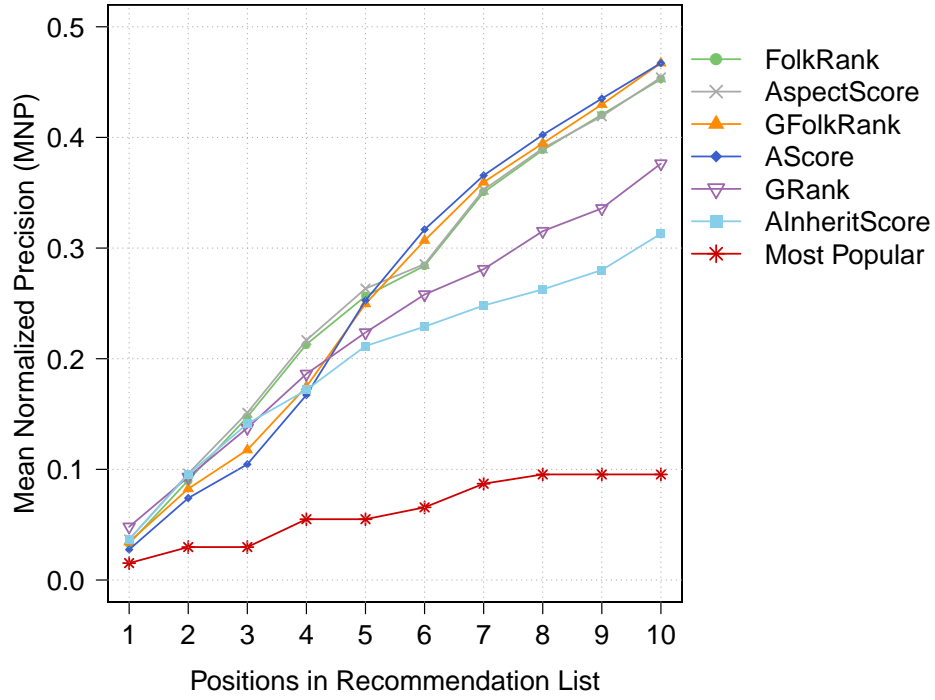
For LeaveRTOut, AScore, AspectScore, GFolkRank and AspectScore are all significantly more effective than GRank and AInheritScore. AScore is significantly more effective than GFolkRank. All algorithms are more effective than Most Popular.

#### Discussion

The LeavePostOut evaluation results show that the proposed algorithms AspectScore, AScore and AInheritScore perform consistently better than their respective baselines across all evaluation metrics. AScore outperforms all algorithms across all evaluation metrics. AScore is significantly more effective than FolkRank and Most Popular with a large effect size ( $d = 1$ ), according to the Cohen's  $d$  values in Table B.20 in Appendix B, and significantly more effective than its closest baseline GFolkRank, however with a small effect size ( $d = 0.04$ ). AScore is also significantly more effective than AspectScore, GRank and AInheritScore with a moderate effect size of approximately  $d = 0.4$ . AInheritScore also performs

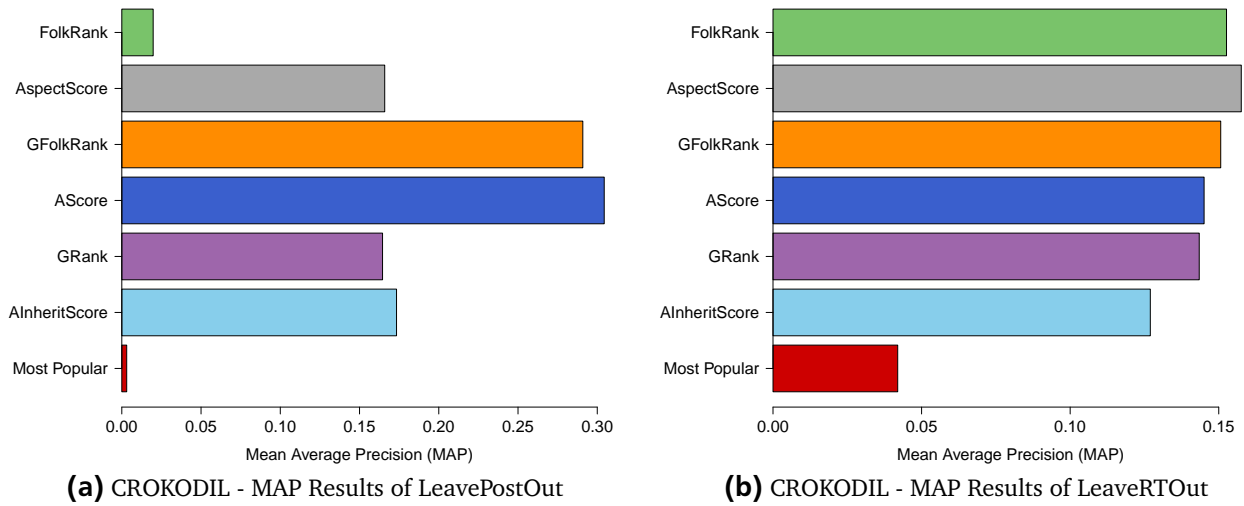


(a) CROKODIL - MNP Results of LeavePostOut



(b) CROKODIL - MNP Results of LeaveRTOut

**Figure 5.12:** CROKODIL - Mean Normalized Precision (MNP) Results of LeavePostOut and LeaveRTOut



**Figure 5.13:** CROKODIL - Mean Average Precision (MAP) Results of LeavePostOut and LeaveRTOut

better than its baseline GRank and FolkRank across evaluation metrics. It is also significantly more effective than FolkRank with a large effect size of  $d = 0.7$  and more effective than GRank but with a small effect size of  $d = 0.03$ . These results show that AScore and AlInheritScore have an added advantage compared to their respective baselines in finding relevant resources for the user for LeavePostOut through the exploitation of additional connections between the user and activities and between activities in the activity hierarchy. AspectScore also outperforms its baselines FolkRank and Most Popular across evaluation metrics and is significantly more effective with a large effect size of about  $d = 0.8$ . AspectScore is also significantly more effective than GRank and AlInheritScore but with a small effect size. Thus the exploitation of additional semantic information gained from the tag types in CROKODIL is advantageous.

For LeaveRTOut, AspectScore achieves the highest MAP. However, regarding AP, AspectScore is only significantly more effective than GRank and AlInheritScore with a small effect size of  $d = 0.1$  and more effective than Most Popular with a large effect size of  $d = 0.7$ . AScore is significantly more effective than GFolkRank, GRank, AlInheritScore and Most Popular with small effect sizes of  $d = 0.03$ ,  $d = 0.01$ ,  $d = 0.09$  and a moderate effect size of  $d = 0.6$  respectively. MNP results show that AScore is weaker than the other algorithms in the topmost positions up till position  $k = 6$ , then it achieves the highest MNP consistently for the rest of the top positions. AlInheritScore contends with GRank, AScore, GFolkRank and FolkRank for the MNP topmost positions. But similar to the results from the GroupMe! dataset, AlInheritScore is only more effective than Most Popular. Most Popular once again performs worst across evaluation methods and evaluation metrics. In contrast to the LeavePostOut results, FolkRank contends with AspectScore, AScore, GFolkRank, GRank and AlInheritScore for the MNP top positions and achieves a higher MAP than all other algorithms except AspectScore. However FolkRank is only significantly more effective than GRank, AlInheritScore and Most Popular with small effect sizes of  $d = 0.04$ ,  $d = 0.01$  and a large effect size of  $d = 0.6$  respectively.

## Limitations

The CROKODIL dataset was collected under real conditions where the CROKODIL application was used in varied RBL scenarios, for example for personal learning or as part of a group in a seminar. Thus the CROKODIL dataset is the most valid and representative dataset available for this thesis's particular RBL application scenario described in Section 2.4. The number of tag assignments and posts were however few when compared to the GroupMe! or BibSonomy datasets. The number of users was also very limited. But the results of the evaluation are still comparable to the previous evaluation results on GroupMe! and



	Baseline	Metric	LeavePostOut	LeaveRTOut
AspectScore	Most Popular	MNP	⊗	⊗
		MAP	⊗	⊗
		AP	⊗	⊗
	FolkRank	MNP	⊗	⊗
		MAP	⊗	⊗
		AP	⊗	(□)
AScore	Most Popular	MNP	⊗	⊗
		MAP	⊗	⊗
		AP	⊗	⊗
	FolkRank	MNP	⊗	(□)
		MAP	⊗	□
		AP	⊗	(□)
	GFolkRank	MNP	⊗	(□)
		MAP	⊗	□
		AP	⊗	⊗
AInheritScore	MostPopular	MNP	⊗	⊗
		MAP	⊗	⊗
		AP	⊗	⊗
	FolkRank	MNP	⊗	□
		MAP	⊗	□
		AP	⊗	□
	GRank	MNP	⊗	□
		MAP	⊗	□
		AP	⊗	□

**Table 5.8:** CROKODIL - Summary of Evaluation Results of AspectScore, AScore and AInheritScore

BibSonomy, thus showing that the proposed recommender algorithms still perform acceptably in a real RBL scenario where the dataset is small and sparse and the number of users limited.

## Conclusion

The evaluation results are summarized in Table 5.8. Comparable to the results on the GroupMe! dataset, results from LeavePostOut show that the algorithms leveraging additional semantic information are overall more effective than FolkRank and Most Popular.

For LeaveRTOut, the results are also comparable to the results on the GroupMe! dataset. The performance of all algorithms drops in contrast to LeavePostOut. In the CROKODIL dataset, the number of tag assignments is the most numerous of all relationships between the entities in the extended folksonomy, thus a lot of information is lost when all tag assignments are removed between the query tag and the relevant resource in the RT-posts. Once again, the approaches based on the random surfer model suffer less due to the fact that isolated nodes in the folksonomy are still reachable.

## 5.3 Summary

In this chapter, the evaluation method LeaveRTOut [203] and the evaluation metric Mean Normalized Precision (MNP) [203] are presented. LeaveRTOut is an alternative evaluation method used to complement results from LeavePostOut as it sets a different recommendation task. Mean Normalized Precision complements the Mean Average Precision metric as it gives a different view on the top-most ranked recommendation results of a recommender algorithm.

The performance of the recommender algorithms are dependent on the evaluation method and evaluation metric applied. Therefore it is important to evaluate with different evaluation methods in order to have a more comprehensive overview of an algorithm's performance. It is also necessary to know what is

---

important for the evaluation of the algorithm. For accurate recommendations in the topmost positions, then MNP is an effective metric, whereas if the overall precision of the recommendations is important, then the MAP can show this best.

In conclusion, the overall results of the evaluations show that incorporating additional semantic information in hybrid recommender approaches in folksonomies is beneficial to provide more relevant recommendations. The performance of individual recommender algorithms depends on the dataset, evaluation method and evaluation metric used. Thus the selection of a recommender algorithm will depend on its application scenario and what is important for making recommendations. In the following chapter, the recommender algorithm AScore is evaluated with an alternative evaluation approach based on crowdsourcing. The results from the crowdsourcing experiment complement the offline evaluation results achieved in this chapter and further counter the incompleteness problem.



---

## 6 An Alternative Evaluation Approach for TEL Recommender Systems

---

A lot of research has gone into the evaluation of TEL recommender systems based on standard methods from information retrieval which are mostly based on determining the precision of such algorithms using cross-validation on historical or synthetically created datasets as described in Section 3.2. These offline experiments are fast to conduct once the datasets exist. They can be readily repeated and easily compared to other evaluation results [157]. However, finding datasets that fulfil the exact requirements for a specific algorithm is a challenge. For example, in order to evaluate the graph-based recommender approach *AScore* described in Section 4.1, a dataset with a hierarchical activity structure is required, but such datasets are very rare [19]. Hence, offline experiments based on historical datasets are limited. This is one motivation for an alternative evaluation method to evaluate recommender algorithms for TEL. A further motivation arises with the limitation in scope of offline experiments due to the incompleteness problem [49]. Offline experiments are limited to the historical interactions of the user in the dataset. Consequently, there is no way of determining whether a newly recommended resource would be found relevant by the user or not, as the user did not have nor know this resource in the past. Furthermore, user-centric metrics like novelty or diversity are best evaluated by asking the users themselves [167]. There have been attempts to complement offline experiments by conducting user studies [157]. User studies are however very expensive with regard to the time and effort needed to plan and execute them. From the survey in Section 3.1.3, only about 40 participants take part in user studies on average. Furthermore, few variations of an algorithm can be tested and it is very expensive to repeat them. There therefore exists a gap between the fast, easy-to-conduct offline experiments and user studies. One attempt to bridge this gap is by using crowdsourcing [11, 50, 93]. A concept for evaluating TEL recommender systems using crowdsourcing is therefore proposed in this chapter. As a proof-of-concept, the *AScore* algorithm presented in Section 4.1 is evaluated in a repeated crowdsourcing experiment.

According to the requirements of recommender systems for TEL presented in Section 3.1.3, it is necessary to recommend personalized learning resources that are novel and serendipitous to the learner. From the survey in Section 3.2.2, the evaluation of novelty and serendipity are unfortunately lacking. In Section 3.2.1 it is argued that serendipity can be measured by testing for novelty and diversity. Thus for the proof-of-concept crowdsourcing experiment, relevance, novelty and diversity shall be measured. The first experiment called Experiment Spring is complemented with a repeat of the experiment called Experiment Autumn in order to verify the results from the first experiment and to validate the evaluation concept, affirming that neither the choice of activities nor the selected recommendations for the experiments directly influence the results obtained.

---

### 6.1 Crowdsourcing as an Evaluation Approach in Research

---

Crowdsourcing is an open call to online users, so called *crowdworkers*, who come from a very large community all over the world. Crowdworkers are asked to solve a problem or to perform a human intelligent task, a so called microtask, in exchange for payments, social recognition or entertainment [120]. Table 6.1 gives an overview of some popular crowdsourcing platforms. The main advantage of crowdsourcing is the fast, flexible access to a large population of willing participants at a relative low cost [13]. The quality of results gained from crowdsourcing experiments have been found to be comparable to those from traditional user experiments [120, 122], depending on the design of the task to solve [11]. The influence of payments on crowdsourcing results have also been analysed [122, 164]. The higher the payments, the less the spamming but an increase in the quality of the results could not be justified [122]. The drawbacks of crowdsourcing as an evaluation approach are the unknown participants, the artificial task, and the costs of detecting spammers [13] or so called *random clickers* [122].

Crowdsourcing Platform	Popularity	Characteristics	Year Founded
microworkers	500,000 crowdworkers worldwide	Flexible forwarding to other hosting platforms e.g. for questionnaires	2009
CrowdFlower	5 million crowdworkers in 208 countries	Access to many other crowdsourcing platforms	2007
Amazon Mechanical Turk	414,342 microtasks with crowdworkers from USA and India	Most well-known platform	2005
Clickworker	500,000 crowdworkers in 136 countries	Mostly used in Germany	2005

**Table 6.1:** Crowdsourcing Platforms (facts retrieved online as of 03.07.2014)

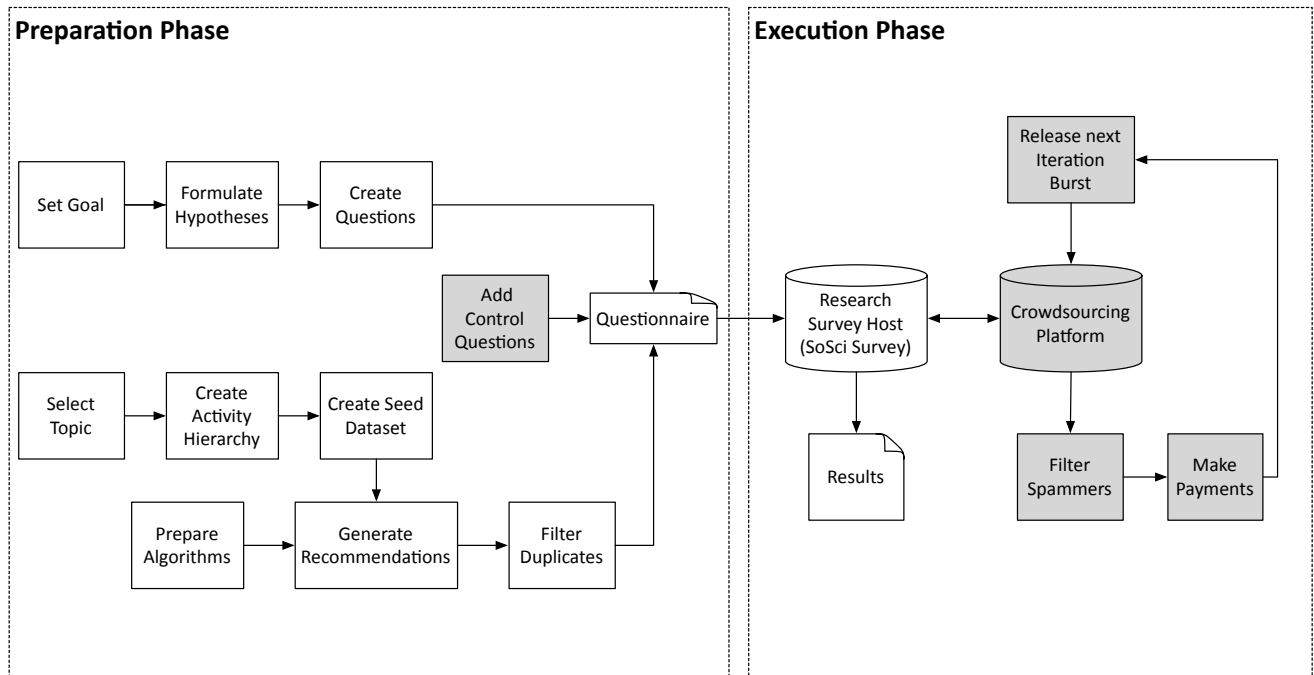
Crowdsourcing Task	Crowdsourcing Platform	Payments
Relevance evaluations of document recommendations [93]	Amazon Mechanical Turk	0.02\$
Relevance assessment of documents [12]	Amazon Mechanical Turk	0.04\$
Preference judgements of novel and diverse documents [50]	Amazon Mechanical Turk	0.80\$
Search engine evaluation [120]	Amazon Mechanical Turk	0,10\$ - 0.25\$
Repeatability of search system evaluations [32]	Amazon Mechanical Turk	0,20\$
Judgement of Wikipedia articles [124]	Amazon Mechanical Turk	0.05\$
Relevance and relatedness judgements of movies to learning objects [119]	CrowdFlower	Not stated
Select matching movies to evaluate diversity of recommendations [151]	CrowdFlower	Not stated

**Table 6.2:** Overview of Crowdsourcing in Research

Crowdsourcing has been applied in diverse domains to solve varied tasks for research. For example, to conduct surveys, usability testing, classification or translation tasks [11]. As well as for creating datasets for machine learning, annotations for natural language texts and feature selection for ranking algorithms [21]. In information retrieval, there is a particular example called TERC - Technique for Evaluating Relevance by Crowdsourcing [11], developed to test the effectiveness of modern information retrieval systems. Recommender strategies have also been evaluated using crowdsourcing to determine the relevance of the recommendations made [93]. User-centric metrics such a novelty and diversity have been evaluated with crowdsourcing [50]. Table 6.2 gives an overview of related work where crowdsourcing has been applied to perform various evaluation tasks.

## 6.2 Crowdsourcing Concept for Evaluating Recommender Systems for TEL

The proposed crowdsourcing concept for evaluating recommender systems for TEL [78, 79, 175] is presented in the following sections. The concept is made up of two phases as shown in Figure 6.1:



**Figure 6.1:** Evaluation Design and Execution Process highlighting the Steps specific to Crowdsourcing

the preparation phase and the execution phase. The preparation phase covers the same steps needed to prepare a standard user study (excluding adding control questions) [87, 108, 130]. The execution phase is specifically for a crowdsourcing experiment. The steps specific to a crowdsourcing experiment in Figure 6.1 are highlighted in grey.

### 6.2.1 The Preparation Phase

The preparation phase is shown in Figure 6.1, where the goals of the experiment are first determined and then hypotheses are specified. A topic is selected and questions are created for the questionnaires. The recommendations are generated after a seed dataset has been created and an activity hierarchy. The following sections explain these steps in the preparation phase in detail.

#### Set Goal

The goal of the experiment needs to be specified as precisely as possible. The aim of the proof-of-concept crowdsourcing experiment is the evaluation of the graph-based recommender algorithm AScore. The first goal of the experiment is to determine whether the recommendations made by AScore are more relevant to a specified activity as well as being more novel and diverse to the learner when compared to recommendations made by the baseline algorithm FolkRank. This presumption should also hold for recommendations made by AScore to sub-activities and to super-activities. A second goal is to find out if recommendations made by AScore to sub-activities lower down in the activity hierarchy (*A\_Sub*) are more relevant, novel and diverse than those made by AScore to activities higher up in the hierarchy (*A\_Super*).

#### Formulate Hypotheses

Based on these goals, hypotheses are defined. For the first goal, where AScore is compared to FolkRank for relevance, novelty and diversity, the following three hypotheses are made.

- **Hypothesis 1A: Relevance** - learning resources recommended by AScore are more relevant to a specified topic than learning resources recommended by FolkRank. This also holds for sub- and super-activities. Recommendations made by AScore to sub-activities (A\_Sub) are more relevant than recommendations made by FolkRank to sub-activities (F\_Sub). Recommendations made by AScore to super-activities (A\_Super) are more relevant than recommendations made by FolkRank to super-activities (F\_Super).
- **Hypothesis 2A: Novelty** - learning resources recommended by AScore are more new or unknown to the learner than those recommended by FolkRank. This also holds for sub- and super-activities. Recommendations made by AScore to sub-activities (A\_Sub) are more novel than recommendations made by FolkRank to sub-activities (F\_Sub). Recommendations made by AScore to super-activities (A\_Super) are more novel than recommendations made by FolkRank to super-activities (F\_Super).
- **Hypothesis 3A: Diversity** - AScore recommends more diverse learning resources than FolkRank. This also holds for sub- and super-activities. Recommendations made by AScore to sub-activities (A\_Sub) are more diverse than recommendations made by FolkRank to sub-activities (F\_Sub). Recommendations made by AScore to super-activities (A\_Super) are more diverse than recommendations made by FolkRank to super-activities (F\_Super).

For the second goal, the recommendations made to sub-activities and the recommendations made to super-activities are to be compared, thus the following hypotheses are made.

- **Hypothesis 1B: Relevance** - AScore recommends more relevant learning resources to sub-activities lower down in the hierarchy (A\_Sub) than to activities higher up in the hierarchy (A\_Super).
- **Hypothesis 2B: Novelty** - AScore recommends more novel learning resources to sub-activities lower down in the hierarchy (A\_Sub) than to activities higher up in the hierarchy (A\_Super).
- **Hypothesis 3B: Diversity** - AScore recommends more diverse learning resources to sub-activities lower down in the hierarchy (A\_Sub) than to activities higher up (A\_Super).

These hypotheses should not necessarily hold for FolkRank (F\_Sub and F\_Super) as the activity hierarchy is not considered.

---

### Create Questions for the Questionnaire

---

From these hypotheses, questions are then formulated for the questionnaire as shown below. To measure each hypothesis, three questions are created. Each questionnaire contains a total of 10 questions: 3 questions for each hypothesis and one control question to detect spammers. The questionnaire for the survey is shown in Appendix C. Basically, each questionnaire contains questions to only one topic from the activity hierarchy created in the initial research mentioned above. To each topic, 5 resources were recommended either from the algorithm AScore or from FolkRank. A topic is either a sub-activity or an activity higher up in the hierarchy (a super-activity) as shown in Figure 6.2. For the experiments, the questionnaires were hosted on the scientific research survey platform SoSci Survey<sup>1</sup>. The following questions are formulated for each hypothesis and as control questions [78, 79].

#### Hypothesis 1: Relevance

- Question 1: The given Internet resource supports me very well in my research about the topic.
- Question 2: If I could only use this resource, my research would still be very successful.
- Question 3: Without this resource just by using my own resources, my research about the given topic would still be very good.

---

<sup>1</sup> <http://www.soscisurvey.de>, retrieved 10.06.2014



---

### Hypothesis 2: Novelty

- Question 4: The Internet resource gives me new insights and/ or information for my task.
- Question 5: I would have found this resource on my own/ anyway/ during my research.
- Question 6: There are lots of important aspects about the topic described in this resource that lack in other resources.

### Hypothesis 3: Diversity

- Question 7: This Internet resource differs strongly from my other resources.
- Question 8: This resource informs me comprehensively about my topic.
- Question 9: This resource covers the whole spectrum of research about the given topic.

### Control Questions

- Control Question 1: How many pictures and tables that are relevant to the given research topic does the given resource contain?
- Control Question 2: Give a short summary of the recommended resource above by giving 4 keywords describing its content.
- Control Question 3: Describe the content of the given resource in two sentences.

---

### Select Topic

---

Next, a topic needs to be chosen in order to create an activity structure for the initial research to create a seed dataset to generate recommendations on. The topic needs to be a currently well-known topic so most participants of the survey can understand and better judge the resources recommended to the topic. For these experiments, the topic *Understanding Climate Change* was chosen as it is a very current topic having sufficient web resources online [117].

---

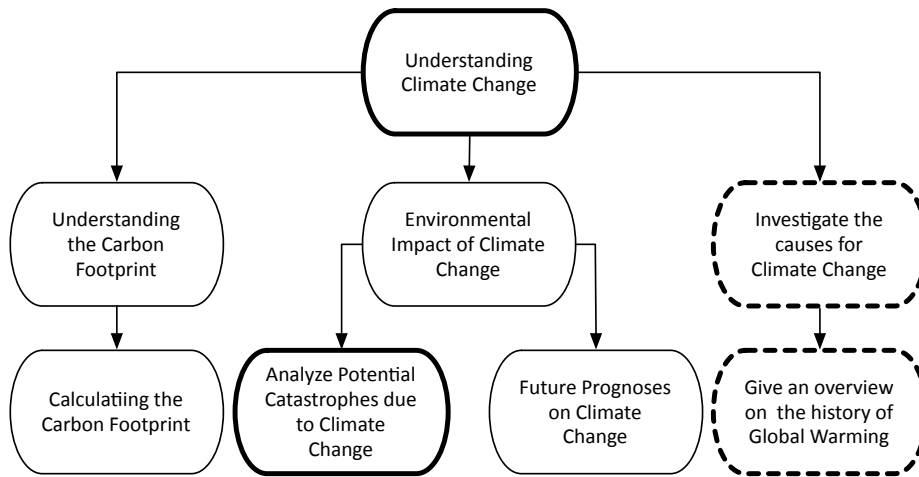
### Create Activity Hierarchy and Generate a Seed Dataset

---

An initial research on the selected topic is needed in order to generate a seed dataset to be able to create recommendations for the experiment. The selected topic first needs to be broken down into sub-activities and a hierarchical activity structure needs to be created. Then relevant web resources need to be found, tagged and attached to the relevant activities. Several possible variations of generating a seed dataset are considered [174, 175]:

- In the first variation, all participants in the experiment take part in generating the seed dataset required to generate the recommendations to be evaluated in the experiment afterwards. The advantage here is that all participants have researched on the topic and could recognize resources recommended later in the experiment.
- In the second variation, the participants are separated into two distinct groups, one group only generates the seed dataset and the other group takes part in the experiment. In this variation, a larger number of participants is needed, as a part is only used to generate the seed dataset.
- In the third variation, the participants are separated into two groups again, but the group that generates the seed dataset takes part in the experiment as well.

For the crowdsourcing experiment, the second variation is selected as this decouples the generation of the seed dataset from the actual experiment, making it most suitable for crowdsourcing. In order to generate the seed dataset for the experiment, 5 recruited participants (students) were asked to conduct an initial research on the topic using the platform CROKODIL [16]. Firstly, the topic was broken down into 7 sub-activities in a hierarchical activity structure as shown in Figure 6.2. About 70 web resources were found on the Web relating to the activities. These web resources were tagged and attached to the



**Figure 6.2:** Activity Hierarchy for Experiments

	Users	Tags	Resources	Activities	Posts	Tag Assignments
CROKODIL Dataset	44	863	502	186	637	1895
Seed Dataset	5	165	65	8		

**Table 6.3:** Seed Dataset as part of the CROKODIL extended Folksonomy

corresponding activities in the hierarchy structure. An extended folksonomy comprising the users, tags, resources and activities thus form a seed dataset to generate the recommendations on.

### Generate Recommendations for the Experiment

The two recommender algorithms: AScore and FolkRank are then run on this seed dataset as part of the CROKODIL extended folksonomy [19] comprising the users, resources, tags, activities and activity hierarchies as shown in Table 6.3. Such a limited seed dataset would be inadequate for an offline experiment but it is sufficient to prepare a crowdsourcing experiment.

The four activities highlighted in Figure 6.2 are selected as the focus for the algorithms in order to generate recommendations specifically for these activities. These activities are also later on specified in the questionnaires as the crowdworkers participating in the experiment must be aware of the activity to which the recommendations have been generated for. In Experiment Spring, the activity “*Understanding Climate Change*” was selected as super-activity and the activity “*Analyze the catastrophes which are currently happening or going to happen because of the higher worldwide temperature*” as sub-activity for the experiment. In Experiment Autumn, two new activities were chosen: “*Give an overview about the history of global warming*” was selected as sub-activity and “*Investigate the causes of climate change*” as super-activity. Hence, the recommendations generated by AScore and FolkRank were different in both experiments, thus ensuring that the results do not depend on the activities nor the recommendations selected for the experiments. Duplicate recommendations from both algorithms are filtered out. The recommendations generated for both Experiment Spring and Experiment Autumn are listed with their URLs in Appendix C.1.1.

The main challenge in the preparation phase is defining suitable evaluation goals that can be broken down into small compact tasks that are solvable online by crowdworkers, who generally want to accomplish these tasks in a short period of time. It helps to pose simple, short questions to well-defined tasks that can be accomplished in about 15 - 20 minutes.

	Sub-Activity	Super-Activity
AScore	A_Sub	A_Super
FolkRank	F_Sub	F_Super

**Table 6.4:** Treatment Conditions

Experiment Spring			
	Sub-Activity	Super-Activity	Total Participants
AScore	A_Sub: 45	A_Super: 39	84
FolkRank	F_Sub: 39	F_Super: 36	75
Total Participants	84	75	159
Experiment Autumn			
	Sub-Activity	Super-Activity	Total Participants
AScore	A_Sub: 80	A_Super: 73	153
FolkRank	F_Sub: 76	F_Super: 85	161
Total Participants	156	158	314

**Table 6.5:** Random Assignment of Participants across Treatment Conditions

### 6.2.2 The Execution Phase

The execution phase is shown on the right in Figure 6.1. The questionnaire prepared in the preparation phase is hosted on a research survey platform and offered as a task to participants on a crowdsourcing platform. After the experiment, the results are extracted and analysed.

#### Crowdsourcing Platforms and Participants

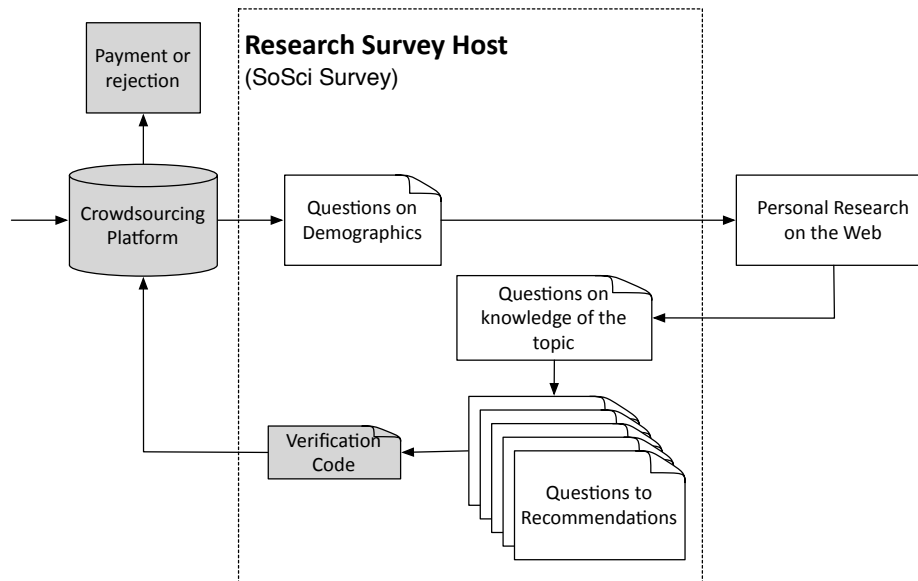
The crowdsourcing platforms microWorkers<sup>2</sup> and CrowdFlower<sup>3</sup> were chosen for the experiments. Each participant is randomly assigned to one of four treatment conditions as shown in Table 6.4.

- A\_Sub where recommendations from AScore are made to a sub-activity.
- A\_Super where recommendations from AScore are made to a super-activity.
- F\_Sub where recommendations from FolkRank are made to a sub-activity.
- F\_Super where recommendations from FolkRank are made to a super-activity.

The distribution of the participants in the experiments across treatment conditions for both Experiment Spring and Experiment Autumn is shown in Table 6.5 and more details regarding the participants in Appendix C.2. For Experiment Spring, the evaluation jobs were placed on two crowdsourcing platforms: 60 jobs on microWorkers and 40 jobs on CrowdFlower. Crowdworkers participated from all over the world, most however came from the USA and Bangladesh. After eliminating a total of 243 spammers, a total of 67 fully answered questionnaires were gotten from crowdworkers: 35 from microworkers and 32 from CrowdFlower. Additionally, 92 voluntary non-crowdworkers (mostly students) were invited to take part in the survey. 159 questionnaires were considered for the evaluation in Experiment Spring. A total of 84 participants received recommendations from AScore and 75 received recommendations from FolkRank. The sub-activity “*Analyze the potential catastrophes due to climate change*” was assigned to 84

<sup>2</sup> <http://www.microworkers.com>, retrieved 19.06.2014

<sup>3</sup> <http://crowdflower.com>, retrieved 19.06.2014



**Figure 6.3:** Workflow of Crowdsourcing Experiment from the Perspective of a Crowdworker

participants (45 for A\_Sub and 39 for F\_Sub) and the super-activity “*Understanding climate change*” to 75 participants (39 for A\_Super and 36 for F\_Super).

In Experiment Autumn, all evaluation jobs were placed on the crowdsourcing platform microWorkers. microWorkers was chosen as it allows the restriction of access to certain countries. As the majority of spammers identified came from Bangladesh, access was restricted to countries other than Bangladesh for most of the experiment. After eliminating 549 spammers, a total of 314 participants took part in the experiment, nearly twice as many as in Experiment Spring. 153 participants were given recommendations from AScore and 161 participants recommendations from FolkRank. The sub-activity “*Give an overview about the history of global warming*” were assigned to 156 participants (80 for A\_Sub and 76 for F\_Sub) and 158 to the super-activity “*Investigate the causes of climate change*” (73 for A\_Super and 85 for F\_Super).

Further details of the demographics of the participants in the experiments are shown in Appendix C.2. The ages of the participants ranged mostly between 20 to 35 years. Most participants had a Bachelor’s Degree (BSc.) and the majority were male. The countries most representative were India, Bangladesh, Nepal and USA. In Experiment Spring, Germany had a high frequency due to the volunteers. The demographics correspond to those in related work [207]

---

## Workflow of Experiment

---

The workflow of the crowdsourcing experiment from the perspective of a crowdworker is shown in Figure 6.3. The steps specific to a crowdsourcing experiment are highlighted. After selecting the microtask on the crowdsourcing platform, the crowdworker is given the link to the questionnaire hosted on a research survey host. Screenshots of the questionnaire are shown in Appendix C. At the beginning of the experiment, after collecting demographic and general information like age, gender, level of education and country, the participants are asked to perform a short research on the Web about the specified topic in order to be able to judge the relevance, novelty and diversity of the recommended resources to be shown later on in the experiment. After this, the crowdworker is asked to judge his current state of knowledge regarding the topic of Climate Change in order to identify experts in this field. When the experiment is completed, a verification code is given to the participant. Crowdworkers can submit this verification code to the crowdsourcing platform and receive payment if the experiment was correctly completed or a rejection if the answers were judged as spam by the owner of the microtask.

---

## Iteration Bursts

---

The number of participants taking part in the experiment at one go is controlled by setting iteration bursts, where a limited number, like about 50 to 100 participants are allowed to take part in the survey in one iteration. This allows for a better control of the quality of the crowdworkers as their answers need to be cross-checked, spammers identified and reported. Then the next burst of participants are released and so on. One iteration burst is usually completed by crowdworkers within a few hours of being released, however an efficient detection and filtering of spammers poses a major challenge here and defining effective control questions is crucial. After each iteration burst, the verified participants are paid. For these experiments, a payment of 0.75\$ for each questionnaire was offered, which is comparable to related work (see Table 6.2), where most jobs are even offered at lower rates, depending of course on the complexity of the task. Finally, after all iteration bursts have been completed and the spammers filtered out, the responses to the questionnaires are extracted from the crowdsourcing platform and the results are analysed.

---

## Filtering Spammers

---

Filtering criteria were applied successively according to priority. The first filter applied is the total time spent on the survey. If the participant spent less than 5 minutes on the survey, the participant is marked as a suspected spammer and the next filter is applied and so on. However, as time is not always a relevant indication of not spamming (spending a lot of time on the survey does not mean the participant was necessarily more thorough), other filters need to be applied in parallel as well in order to detect spammers [12]. Other filters are for example the values of the scores given - here clear repetitive patterns are easy to detect or completely random values. As a final filter, the control questions are applied.

---

## 6.3 Crowdsourcing Evaluation Results

---

In this section, the results from Experiment Spring and Experiment Autumn are analysed and statistical tests are performed. First results from AScore are compared to FolkRank for all three hypotheses, then results from AScore's sub-activity are compared to AScore's super-activity. Inference statistics are conducted where the aggregated mean values are compared. Independent two samples Student's t-tests [83] as explained in Appendix B are conducted, comparing relevance, novelty and diversity for algorithms AScore and FolkRank. The results of the t-tests are shown, where:

- The **t-value (t)** is calculated by the independent two samples student's t-test.
- The **degrees of freedom (df)** is  $(n_1 - 1) + (n_2 - 1)$ , where  $n_1$  is the number of scores from the first group and  $n_2$  the number of scores from the second group.
- The **p-value (p)** is the significance level and is chosen at 0.05.
- **Inference** is the decision to reject the null hypothesis or not and as a consequence to support the alternative hypothesis or not.

First, the variances of the samples are tested with an F-Test. Depending on the results of this test, a two sample t-test or a Welch t-test is performed. In this thesis, the statistics tool R<sup>4</sup> is used to compute the statistical tests.

Descriptive statistics are presented in Appendix C.3 where the mean, median, standard deviation, the minimum and maximum values, the first and third quartiles, the kurtosis, skewness and standard error of the answers are shown. The answers to the questions were given on a Likert scale from 1 - 7. Table C.7 in Appendix C.3 gives the summary statistics for all questions on AScore for Experiment Spring

---

<sup>4</sup> <http://www.r-project.org>, retrieved 20.08.2014

and Experiment Autumn. Table C.8 in Appendix C.3 shows the summary statistics of the answers given to FolkRank for Experiment Spring and Experiment Autumn. In Experiment Spring, volunteers were also recruited in addition to crowdworkers for the experiment. Table C.13 in Appendix C.3 shows the summary statistics of the answers given from crowdworkers and volunteers. Overall, the crowdworkers tend to give higher ratings (independent of algorithms) when compared to the volunteers. Table C.14 in Appendix C.3 gives the results of the Cohen's d measure of effect size.

---

### 6.3.1 Analysis of Results Comparing Algorithms AScore and FolkRank

---

The results of Experiment Spring and Experiment Autumn are analysed comparing results from AScore to those from FolkRank.

---

#### Results of Hypotheses 1A - 3A for AScore and FolkRank

---

The postulated hypotheses 1A - 3A in Section 6.2 claim that recommendations made by AScore are more relevant, novel and diverse than those made by FolkRank. The aggregated mean values across the questions for each hypothesis for Experiment Spring are plotted in Figure 6.4 (a). Figure 6.4 (b) shows the aggregated mean values across the questions for each hypothesis for Experiment Autumn. The error bars on the mean plots show the standard deviation i.e. mean + standard deviation is the upper limit and mean - standard deviation is the lower limit of the error bars. In both Experiment Spring and Experiment Autumn, the aggregated mean values for each hypothesis for AScore are higher than those for FolkRank.

#### Inference Statistics for Hypotheses 1A - 3A for AScore and FolkRank

Independent two samples Student's t-tests were conducted for Experiment Spring and Experiment Autumn, comparing relevance, novelty and diversity for algorithms AScore and FolkRank. The results of the t-tests are shown in Table 6.6 and analyzed in detail below.

**Hypothesis 1A: Relevance:** In Table 6.6, the results show a significant difference in the scores for algorithm AScore ( $M = 4.3$ ,  $SD = 1.54$ ) and FolkRank ( $M = 4$ ,  $SD = 1.59$ );  $t(2367) = 4.65$ ,  $p = 3.58e-06 < 0.05$  in Experiment Spring as well as in Experiment Autumn where AScore ( $M = 4.17$ ,  $SD = 1.49$ ) and FolkRank ( $M = 3.96$ ,  $SD = 1.42$ );  $t(4707) = 4.84$ ,  $p = 1.36e-06 < 0.05$ . These results suggest that algorithm AScore recommends more relevant resources than algorithm FolkRank in both Experiment Spring and Experiment Autumn. Thus supporting Hypothesis 1A: Relevance.

**Hypothesis 2A: Novelty:** In Experiment Spring, there is a significant difference in the scores for algorithm AScore ( $M = 4.26$ ,  $SD = 1.58$ ) and FolkRank ( $M = 3.94$ ,  $SD = 1.66$ );  $t(2367) = 4.82$ ,  $p = 1.53e-06 < 0.05$  and there is a significant difference in the scores in Experiment Autumn for algorithm AScore ( $M = 4.31$ ,  $SD = 1.53$ ) and FolkRank ( $M = 4.1$ ,  $SD = 1.41$ );  $t(4707) = 4.95$ ,  $p = 7.65e-07 < 0.05$ , thus suggesting that algorithm AScore recommends more novel resources than algorithm FolkRank in both Experiment Spring and Experiment Autumn. This means Hypothesis 2A: Novelty is supported.

**Hypothesis 3A: Diversity:** There exists a significant difference in the scores for algorithm AScore ( $M = 4.16$ ,  $SD = 1.69$ ) and FolkRank ( $M = 3.9$ ,  $SD = 1.67$ );  $t(2367) = 3.78$ ,  $p = 1.62e-04 < 0.01$  in Experiment Spring. This is also the case in Experiment Autumn where AScore ( $M = 4.31$ ,  $SD = 1.48$ ) and FolkRank ( $M = 4.04$ ,  $SD = 1.45$ );  $t(4705) = 6.42$ ,  $p = 1.50e-10 < 0.05$ . These results suggest that algorithm AScore recommends more diverse resources than algorithm FolkRank, thereby supporting Hypothesis 3A: Diversity for both Experiment Spring and Experiment Autumn.

From the results, it can be inferred that there exists a significant difference in the scores for algorithm AScore and FolkRank for all three hypotheses in both Experiment Spring and Experiment Autumn. These results suggest that algorithm AScore overall recommends more relevant, novel and diverse resources



Experiment Spring							
Hypothesis	Algorithm	M	SD	t	df	p	Inference
1A: Relevance	AScore	4.30	1.54	4.65	2367	3.58e-06	Supported $p < 0.05$
	FolkRank	4.00	1.59				
2A: Novelty	AScore	4.26	1.58	4.82	2367	1.53e-06	Supported $p < 0.05$
	FolkRank	3.94	1.66				
3A: Diversity	AScore	4.16	1.69	3.78	2367	1.62e-04	Supported $p < 0.05$
	FolkRank	3.90	1.67				
Experiment Autumn							
Hypothesis	Algorithm	M	SD	t	df	p	Inference
1A: Relevance	AScore	4.17	1.49	4.84	4707	1.36e-06	Supported $p < 0.05$
	FolkRank	3.96	1.42				
2A: Novelty	AScore	4.31	1.53	4.95	4707	7.65e-07	Supported $p < 0.05$
	FolkRank	4.10	1.41				
3A: Diversity	AScore	4.31	1.48	6.42	4705	1.50e-10	Supported $p < 0.05$
	FolkRank	4.04	1.45				

**Table 6.6:** Experiment Spring and Experiment Autumn: Results of Two Sample Student's T-Tests for AScore and FolkRank

than algorithm FolkRank for both experiments. The results from Experiment Spring are supported by Experiment Autumn thus showing that the results of the experiment are reproducible.

#### Results of Hypotheses 1A - 3A for A\_Sub and F\_Sub

A further investigation is made whether hypotheses 1A - 3A also hold for recommendations to sub-activities A\_Sub and F\_Sub.

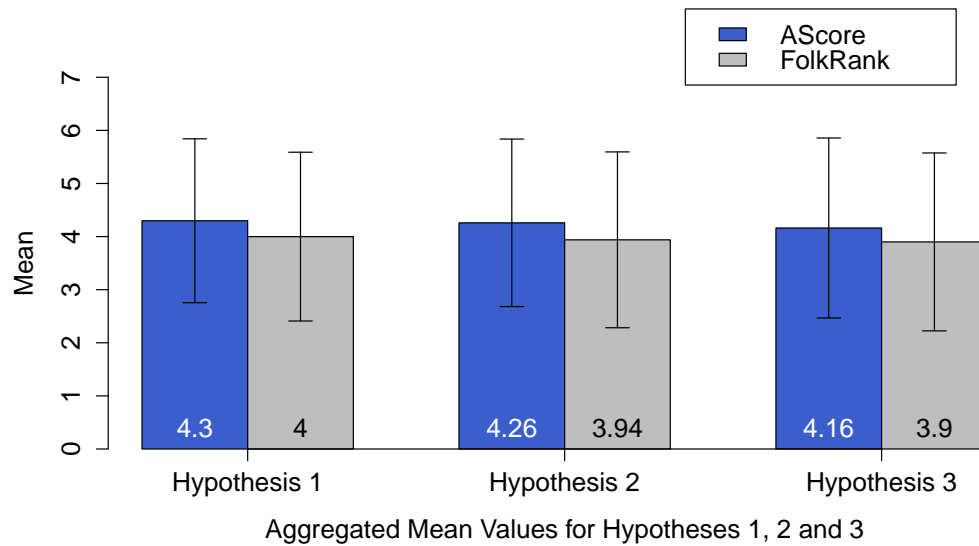
#### Inference Statistics for A\_Sub and F\_Sub

Figure 6.5 (a) shows the results of all three hypotheses for Experiment Spring for A\_Sub and F\_Sub. Figure 6.5 (b) shows the results of A\_Sub and F\_Sub for all three hypotheses for Experiment Autumn. In both Experiment Spring and Experiment Autumn, the mean values for each hypothesis are higher for A\_Sub than for F\_Sub. Table 6.7 shows the results of the significance tests made for each hypothesis and these are analyzed below.

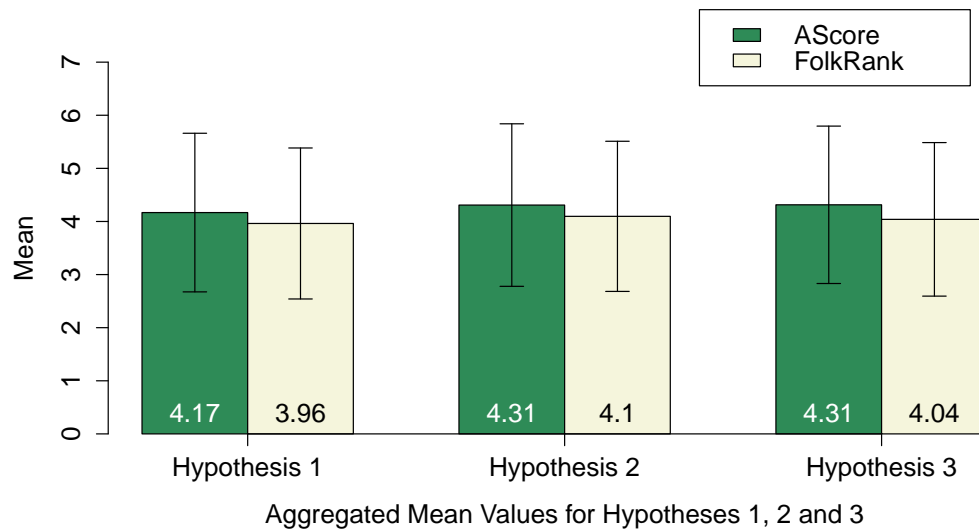
**Hypothesis 1A: Relevance:** The results in Table 6.7 show a significant difference in the scores for A\_Sub ( $M = 4.44$ ,  $SD = 1.56$ ) and F\_Sub ( $M = 3.95$ ,  $SD = 1.51$ );  $t(1242) = 5.57$ ,  $p = 3.05e-08 < 0.05$  in Experiment Spring. In Experiment Autumn, there is also a significant difference where A\_Sub ( $M = 4.27$ ,  $SD = 1.50$ ) and F\_Sub ( $M = 4.04$ ,  $SD = 1.40$ );  $t(2337) = 3.40$ ,  $p = 1.00e-04 < 0.05$ . These results suggest that recommendations by AScore are more relevant to the sub-activity than FolkRank for Experiment Spring and Experiment Autumn.

**Hypothesis 2A: Novelty:** In Experiment Spring, there is a significant difference in the scores for A\_Sub ( $M = 4.36$ ,  $SD = 1.57$ ) and F\_Sub ( $M = 3.97$ ,  $SD = 1.55$ );  $t(1242) = 4.39$ ,  $p = 1.20e-05 < 0.05$ . There is also a significant difference in the scores in Experiment Autumn for A\_Sub ( $M = 4.39$ ,  $SD = 1.56$ ) and F\_Sub ( $M = 4.11$ ,  $SD = 1.41$ );  $t(2331) = 4.56$ ,  $p = 5.38e-06 < 0.05$ . This suggests that algorithm AScore recommends more novel resources to sub-activities than FolkRank in both Experiment Spring and Experiment Autumn.





(a) Experiment Spring: Results of Hypotheses 1A - 3A for AScore and FolkRank



(b) Experiment Autumn: Results of Hypotheses 1A - 3A for AScore and FolkRank

**Figure 6.4:** Experiment Spring and Experiment Autumn: Results of AScore and FolkRank

Experiment Spring							
Hypothesis	Algorithm	M	SD	t	df	<i>p</i>	Inference
1A: Relevance	A_Sub	4.44	1.56	5.57	1242	3.05e-08	Supported <i>p</i> <0.05
	F_Sub	3.95	1.51				
2A: Novelty	A_Sub	4.36	1.57	4.39	1242	1.20e-05	Supported <i>p</i> <0.05
	F_Sub	3.97	1.55				
3A: Diversity	A_Sub	4.27	1.72	3.21	1242	0.0014	Supported <i>p</i> <0.05
	F_Sub	3.96	1.61				
Experiment Autumn							
Hypothesis	Algorithm	M	SD	t	df	<i>p</i>	Inference
1A: Relevance	A_Sub	4.27	1.50	3.40	2337	1.00e-04	Supported <i>p</i> <0.05
	F_Sub	4.04	1.40				
2A: Novelty	A_Sub	4.39	1.56	4.56	2331	5.38e-06	Supported <i>p</i> <0.05
	F_Sub	4.11	1.41				
3A: Diversity	A_Sub	4.47	1.48	6.55	2336	7.10e-11	Supported <i>p</i> <0.05
	F_Sub	4.07	1.44				

**Table 6.7:** Experiment Spring and Experiment Autumn: Results of Two Sample Student's t-Tests for A\_Sub and F\_Sub

**Hypothesis 3A: Diversity:** There exists a significant difference in the scores for A\_Sub ( $M = 4.27$ ,  $SD = 1.72$ ) and F\_Sub ( $M = 3.96$ ,  $SD = 1.61$ );  $t(1242) = 3.21$ ,  $p = 0.0014 < 0.05$  in Experiment Spring. This is also the case in Experiment Autumn where A\_Sub ( $M = 4.47$ ,  $SD = 1.48$ ) and F\_Sub ( $M = 4.07$ ,  $SD = 1.44$ );  $t(2336) = 6.55$ ,  $p = 7.10e-11 < 0.05$ . These results suggest that AScore does recommend more diverse resources to sub-activities than FolkRank for both Experiment Spring and Experiment Autumn.

From the results, it can be inferred that there does exist a significant difference in the scores for A\_Sub and F\_Sub for all three hypotheses in both Experiment Spring and Experiment Autumn. These results suggest that AScore recommends more relevant, novel and diverse resources to sub-activities than FolkRank. Therefore the results of all three hypotheses are supported for the sub-activities as well.

---

#### Results of Hypotheses 1A - 3A for A\_Super and F\_Super

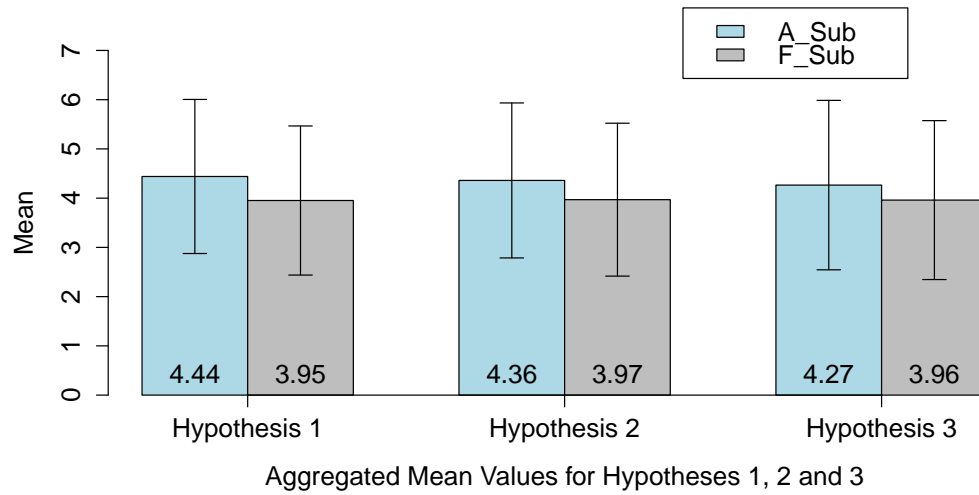
---

It is also investigated whether hypotheses 1A - 3A holds for recommendations to super-activities A\_Super and F\_Super.

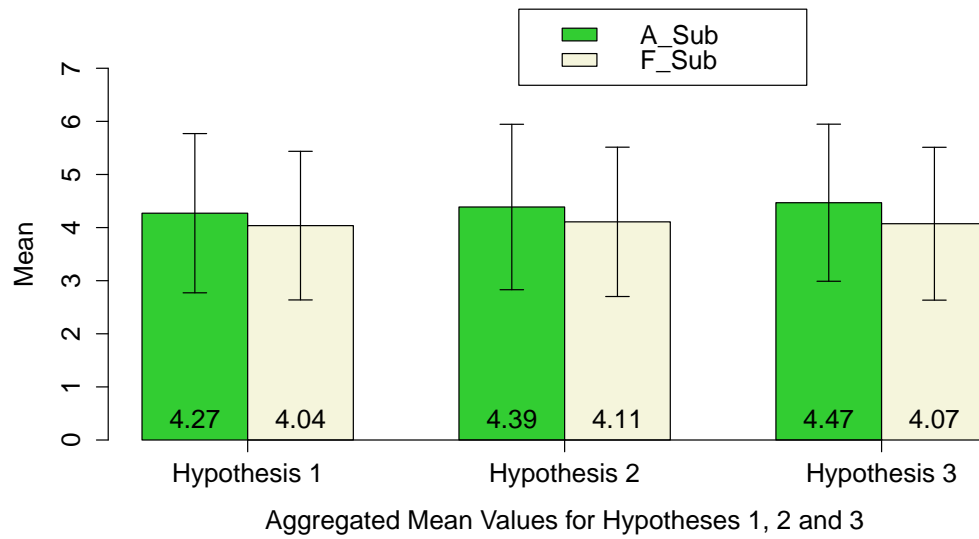
#### Inference Statistics for A\_Super and F\_Super

Figure 6.6 (a) shows the results of all three hypotheses for Experiment Spring. Figure 6.6 (b) shows the results of all three hypotheses for Experiment Autumn. In both Experiment Spring and Experiment Autumn, the mean values for each hypothesis are higher for A\_Super than for F\_Super. Table 6.8 shows the results of the significance tests made for each hypothesis and discussed in the following.

**Hypothesis 1A: Relevance:** In Table 6.8 for Experiment Spring, the results do not show a significant difference in the scores for A\_Super ( $M = 4.14$ ,  $SD = 1.50$ ) and F\_Super ( $M = 4.05$ ,  $SD = 1.67$ );  $t(1087) = 0.93$ ,  $p = 0.35 > 0.05$ . However in Experiment Autumn, there is a significant difference where A\_Super ( $M = 4.05$ ,  $SD = 1.48$ ) and F\_Super ( $M = 3.90$ ,  $SD = 1.44$ );  $t(2368) = 2.65$ ,  $p = 0.0081 < 0.05$ . These results suggest that recommendations by AScore are more relevant to the super-activity than FolkRank only for Experiment Autumn.



(a) Experiment Spring: Results of Hypotheses 1A - 3A for A\_Sub and F\_Sub



(b) Experiment Autumn: Results of Hypotheses 1A - 3A for A\_Sub and F\_Sub

**Figure 6.5:** Experiment Spring and Experiment Autumn: Results of A\_Sub and F\_Sub

Experiment Spring							
Hypothesis	Algorithm	M	SD	t	df	p	Inference
1A: Relevance	A_Super	4.14	1.50	0.93	1087	0.35	Not Supported $p > 0.05$
	F_Super	4.05	1.67				
2A: Novelty	A_Super	4.15	1.57	2.40	1083	0.017	Supported $p < 0.05$
	F_Super	3.91	1.76				
3A: Diversity	A_Super	4.04	1.66	2.08	1123	0.037	Supported $p < 0.05$
	F_Super	3.83	1.74				
Experiment Autumn							
Hypothesis	Algorithm	M	SD	t	df	p	Inference
1A: Relevance	A_Super	4.05	1.48	2.65	2368	0.0081	Supported $p < 0.05$
	F_Super	3.90	1.44				
2A: Novelty	A_Super	4.22	1.50	2.28	2368	0.023	Supported $p < 0.05$
	F_Super	4.09	1.42				
3A: Diversity	A_Super	4.14	1.47	2.23	2367	0.026	Supported $p < 0.05$
	F_Super	4.01	1.45				

**Table 6.8:** Experiment Spring and Experiment Autumn: Results of Two Sample Student's t-Tests for A\_Super and F\_Super

**Hypothesis 2A: Novelty:** In Experiment Spring, there is a significant difference in the scores for A\_Super ( $M = 4.15$ ,  $SD = 1.57$ ) and F\_Super ( $M = 3.91$ ,  $SD = 1.76$ );  $t(1083) = 2.40$ ,  $p = 0.017 < 0.05$  and there is a significant difference in the scores in Experiment Autumn for A\_Super ( $M = 4.22$ ,  $SD = 1.50$ ) and F\_Super ( $M = 4.09$ ,  $SD = 1.42$ );  $t(2368) = 2.28$ ,  $p = 0.023 < 0.05$ . Thus suggesting that algorithm AScore recommends more novel resources to super-activities than FolkRank in both Experiment Spring and Experiment Autumn.

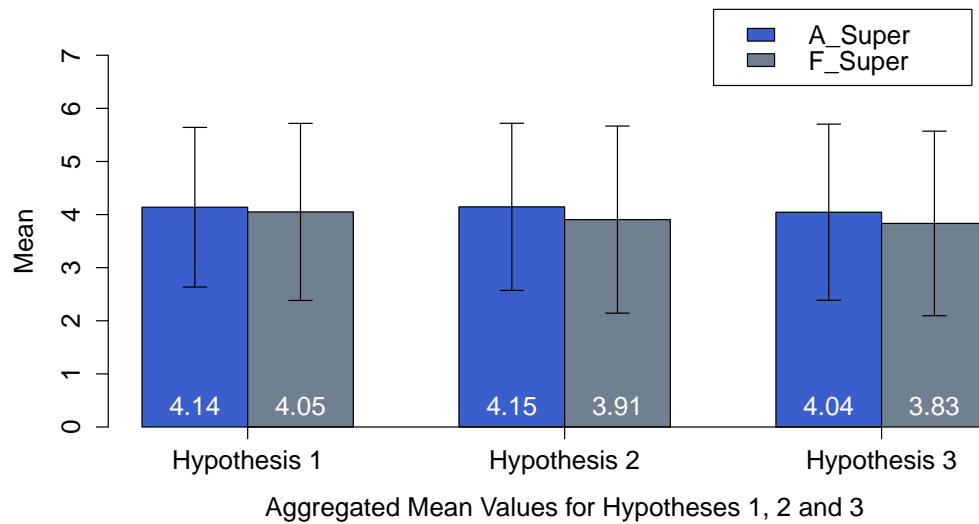
**Hypothesis 3A: Diversity:** There exists a significant difference in the scores for A\_Super ( $M = 4.04$ ,  $SD = 1.66$ ) and F\_Super ( $M = 3.83$ ,  $SD = 1.74$ );  $t(1123) = 2.08$ ,  $p = 0.037 < 0.05$  in Experiment Spring. This is also the case in Experiment Autumn where A\_Super ( $M = 4.14$ ,  $SD = 1.47$ ) and F\_Super ( $M = 4.01$ ,  $SD = 1.45$ );  $t(2367) = 2.23$ ,  $p = 0.026 < 0.05$ . These results suggest that AScore does recommend more diverse resources to super-activities than FolkRank in both Experiment Spring and Experiment Autumn.

From the results, it can be inferred that there does exist a significant difference in the scores for A\_Super and F\_Super for all three hypotheses in Experiment Autumn and for two hypotheses in Experiment Spring. These results suggest that AScore recommends more relevant, novel and diverse resources to super-activities than FolkRank in nearly all experiments. Therefore the results of all three hypotheses are supported for the super-activities as well, except for Hypothesis 1: Relevance for Experiment Spring.

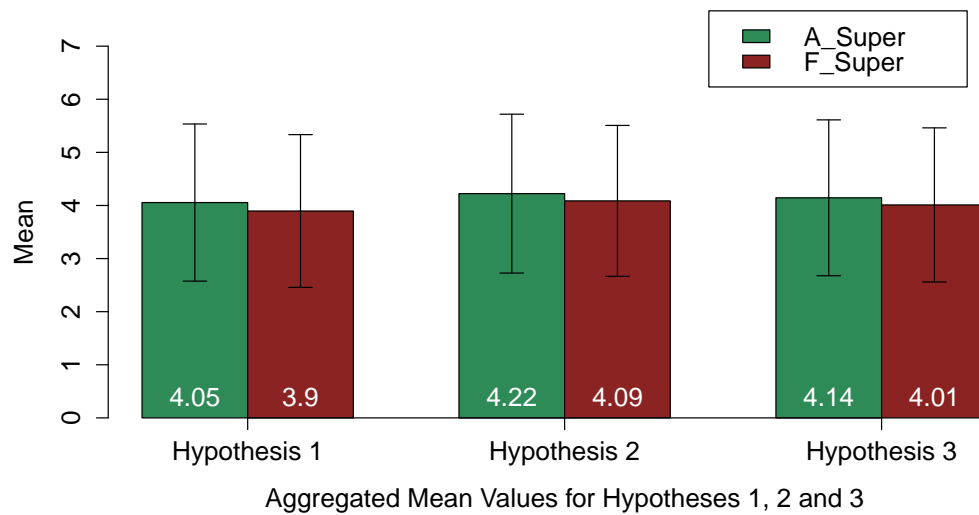
The overall results show that recommendations made by AScore to sub-activities A\_Sub are more relevant, novel and diverse than those made by FolkRank to sub-activities F\_Sub. Recommendations made by AScore to super-activities A\_Super are in nearly all the cases more relevant, novel and diverse than those made by FolkRank to super-activities F\_Super.

### 6.3.2 Analysis of Results Comparing Recommendations to Sub-Activities

The results from Experiment Spring and Experiment Autumn are analysed comparing results from AScore Sub- and Super-Activities and FolkRank Sub- and Super-Activities. The results show that exploiting



(a) Experiment Spring: Results of Hypotheses 1A - 3A for A\_Super and F\_Super



(b) Experiment Autumn: Results of Hypotheses 1A - 3A for A\_Super and F\_Super

**Figure 6.6:** Experiment Spring and Experiment Autumn: Results of A\_Super and F\_Super

---

activities and activity hierarchies as described in the algorithm AScore in Section 4.1 are beneficial as AScore recommends more relevant, novel and diverse resources the more detailed a learner gets in his research. Descriptive statistics are shown in Appendix C.3.

---

#### Results of Hypotheses 1B - 3B for A\_Sub and A\_Super

---

The postulated hypotheses Hypotheses 1B - 3B in Section 6.2 claim that recommendations made by AScore to a more specific topic, which would be the sub-activity in an activity hierarchy, should be more relevant, novel and diverse than recommendations made to a more general activity higher up in the activity hierarchy, which would be a super-activity.

#### Inference Statistics for A\_Sub and A\_Super for Hypotheses 1B - 3B

The aggregated mean values for all three hypotheses for Experiment Spring are shown in Figure 6.7 (a) and in Figure 6.7 (b) are the aggregated mean values for all three hypotheses for Experiment Autumn. In both Experiment Spring and Experiment Autumn, the mean values for each hypothesis are higher for the sub-activity A\_Sub than for the super-activity A\_Super. Table 6.9 shows the results of the significance tests made for each hypothesis and the analysis explained in detail below.

**Hypothesis 1B: Relevance:** In Table 6.9, the results show a significant difference in the scores for A\_Sub ( $M = 4.44$ ,  $SD = 1.56$ ) and A\_Super ( $M = 4.14$ ,  $SD = 1.50$ );  $t(1242) = 3.46$ ,  $p = 5.65e-04 < 0.05$  in Experiment Spring as well as in Experiment Autumn where A\_Sub ( $M = 4.27$ ,  $SD = 1.50$ ) and A\_Super ( $M = 4.05$ ,  $SD = 1.48$ );  $t(2293) = 3.47$ ,  $p = 5.31e-04 < 0.05$ . These results suggest that recommendations by the algorithm AScore are more relevant to the sub-activity than to the super-activity in both Experiment Spring and Experiment Autumn. Thus supporting Hypothesis 1B: Relevance.

**Hypothesis 2B: Novelty:** In Experiment Spring, there is a significant difference in the scores for A\_Sub ( $M = 4.36$ ,  $SD = 1.57$ ) and A\_Super ( $M = 4.15$ ,  $SD = 1.57$ );  $t(1242) = 2.4$ ,  $p = 0.017 < 0.05$  and there is a significant difference in the scores in Experiment Autumn for A\_Sub ( $M = 4.39$ ,  $SD = 1.56$ ) and A\_Super ( $M = 4.22$ ,  $SD = 1.50$ );  $t(2293) = 2.58$ ,  $p = 0.01 < 0.05$ . Thus suggesting that algorithm AScore recommends more novel resources to the sub-activity in both Experiment Spring and Experiment Autumn. This means Hypothesis 2B: Novelty is supported.

**Hypothesis 3B: Diversity:** There exists a significant difference in the scores for A\_Sub ( $M = 4.27$ ,  $SD = 1.72$ ) and A\_Super ( $M = 4.04$ ,  $SD = 1.66$ );  $t(1243) = 2.30$ ,  $p = 0.022 < 0.05$  in Experiment Spring. This is also the case in Experiment Autumn where A\_Sub ( $M = 4.47$ ,  $SD = 1.48$ ) and A\_Super ( $M = 4.14$ ,  $SD = 1.47$ );  $t(2290) = 5.26$ ,  $p = 1.61e-07 < 0.05$ . These results suggest that AScore recommends more diverse resources to the sub-activity than to the super-activity, thereby supporting Hypothesis 3B: Diversity for both Experiment Spring and Experiment Autumn.

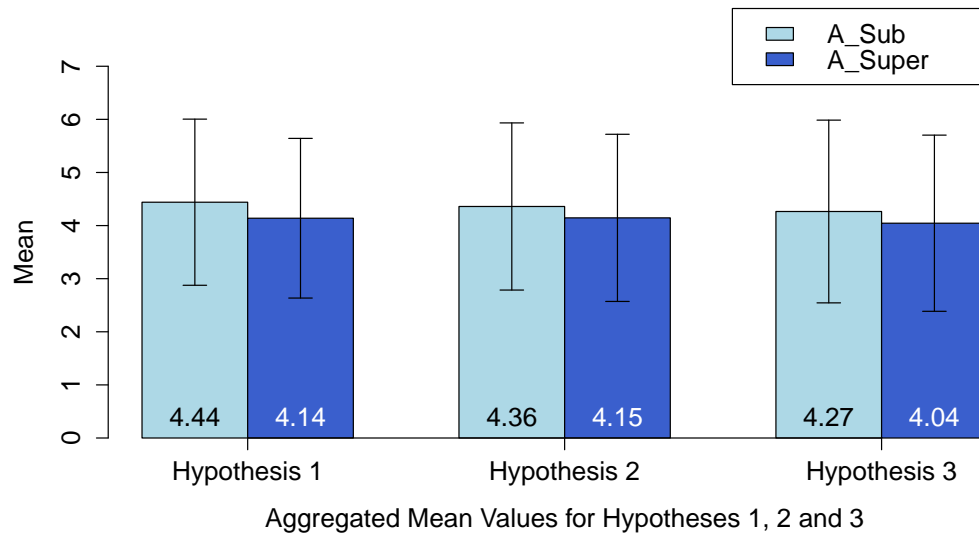
From the results, it can be inferred that there exists a significant difference in the scores for A\_Sub and A\_Super for all three hypotheses in both Experiment Spring and Experiment Autumn. These results suggest that AScore recommends overall more relevant, novel and diverse resources to sub-activities than to super-activities for both Experiment Spring and Experiment Autumn.

---

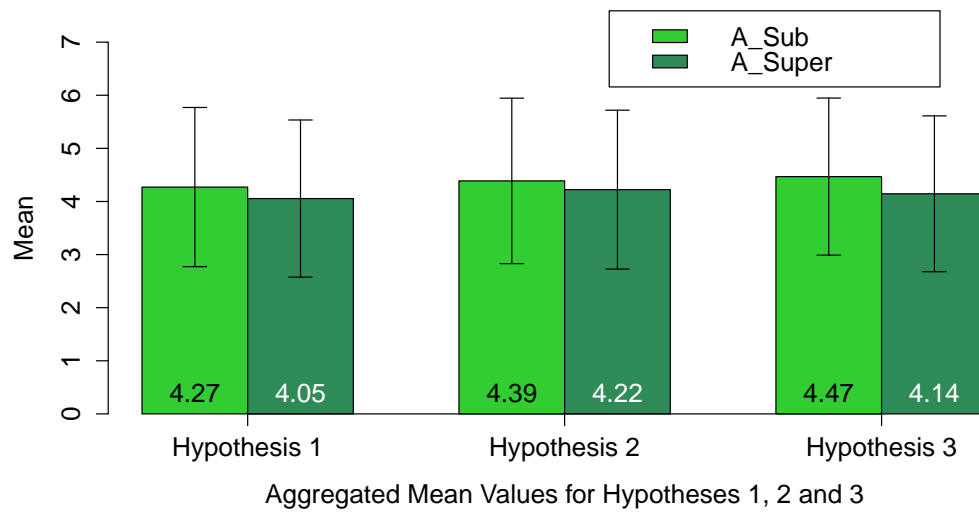
#### Results of Hypotheses 1B - 3B for F\_Sub and F\_Super

---

The postulated hypotheses Hypotheses 1B - 3B in Section 6.2 claim that recommendations made by AScore to a sub-activity in an activity hierarchy, should be more relevant, novel and diverse than recommendations made to a super-activity. As FolkRank does not consider activity hierarchies, it is assumed that the differences in means from F\_Sub and F\_Super should not be significantly different. Figure 6.8 (a) shows the results of all three hypotheses for Experiment Spring for F\_Sub and F\_Super. Figure 6.8 (b)



(a) Experiment Spring: Results of Hypotheses 1B - 3B for A\_Sub and A\_Super



(b) Experiment Autumn: Results of Hypotheses 1B - 3B for A\_Sub and A\_Super

**Figure 6.7:** Experiment Spring and Experiment Autumn: Results of A\_Sub and A\_Super



Experiment Spring							
Hypothesis	Algorithm	M	SD	t	df	p	Inference
1B: Relevance	A_Sub	4.44	1.56	3.46	1242	5.65e-04	Supported
	A_Super	4.14	1.50				p < 0.05
2B: Novelty	A_Sub	4.36	1.57	2.40	1242	0.017	Supported
	A_Super	4.15	1.57				p < 0.05
3B: Diversity	A_Sub	4.27	1.72	2.30	1243	0.022	Supported
	A_Super	4.04	1.66				p < 0.05
Experiment Autumn							
Hypothesis	Algorithm	M	SD	t	df	p	Inference
1B: Relevance	A_Sub	4.27	1.50	3.47	2293	5.31e-04	Supported
	A_Super	4.05	1.48				p < 0.05
2B: Novelty	A_Sub	4.39	1.56	2.58	2293	0.01	Supported
	A_Super	4.22	1.50				p < 0.05
3B: Diversity	A_Sub	4.47	1.48	5.26	2290	1.61e-07	Supported
	A_Super	4.14	1.47				p < 0.05

**Table 6.9:** Experiment Spring and Experiment Autumn: Results of Two Sample Student's t-Tests for A\_Sub and A\_Super

shows the results of F\_Sub and F\_Super for all three hypotheses for Experiment Autumn. In both Experiment Spring and Experiment Autumn, the mean values for each hypothesis are not higher for the sub-activity F\_Sub than for the super-activity F\_Super.

#### Inference Statistics for F\_Sub and F\_Super

Table 6.10 shows the results of the significance tests made for each hypothesis for F\_Sub and F\_Super which are discussed in detail below.

**Hypothesis 1B: Relevance:** In Table 6.10, the results do not show a significant difference in the scores for F\_Sub (M = 3.95, SD = 1.51) and F\_Super (M = 4.05, SD = 1.67);  $t(1123) = -1.03$ ,  $p = 0.30 > 0.05$ . However in Experiment Autumn, there is a significant difference where F\_Sub (M = 4.04, SD = 1.40) and F\_Super (M = 3.90, SD = 1.44);  $t(2412) = 2.44$ ,  $p = 0.015 < 0.05$ . These results suggest that recommendations by the algorithm FolkRank are more relevant to the sub-activity than to the super-activity only for Experiment Autumn.

**Hypothesis 2B: Novelty:** In Experiment Spring, there is no significant difference in the scores for F\_Sub (M = 3.97, SD = 1.55) and F\_Super (M = 3.91, SD = 1.76);  $t(1077) = 0.64$ ,  $p = 0.52 > 0.05$  and there is no significant difference in the scores in Experiment Autumn for F\_Sub (M = 4.11, SD = 1.41) and F\_Super (M = 4.09, SD = 1.42);  $t(2412) = 0.38$ ,  $p = 0.71 > 0.05$ . Thus suggesting that algorithm FolkRank does not recommend more novel resources to the sub-activity than to the activities higher up on the hierarchy in both Experiment Spring and Experiment Autumn.

**Hypothesis 3B: Diversity:** There exists no significant difference in the scores for F\_Sub (M = 3.96, SD = 1.61) and F\_Super (M = 3.83, SD = 1.74);  $t(1122) = 1.27$ ,  $p = 0.20 > 0.05$  in Experiment Spring. This is also the case in Experiment Autumn where F\_Sub (M = 4.07, SD = 1.44) and F\_Super (M = 4.01, SD = 1.45);  $t(2413) = 1.06$ ,  $p = 0.29 > 0.05$ . These results suggest that FolkRank does not recommend more diverse resources to the sub-activity than to the super-activity in both Experiment Spring and Experiment Autumn.

From the results, it can be inferred that there does not exist a significant difference in the scores for F\_Sub and F\_Super for all three hypotheses in Experiment Spring. In Experiment Autumn, only Hypothesis 1B:

Experiment Spring							
Hypothesis	Algorithm	M	SD	t	df	<i>p</i>	Inference
1B: Relevance	F_Sub	3.95	1.51	-1.03	1123	0.30	Not Supported <i>p</i> >0.05
	F_Super	4.05	1.67				
2B: Novelty	F_Sub	3.97	1.55	0.64	1077	0.52	Not Supported <i>p</i> >0.05
	F_Super	3.91	1.76				
3B: Diversity	F_Sub	3.96	1.61	1.27	1122	0.20	Not Supported <i>p</i> >0.05
	F_Super	3.83	1.74				
Experiment Autumn							
Hypothesis	Algorithm	M	SD	t	df	<i>p</i>	Inference
1B: Relevance	F_Sub	4.04	1.40	2.44	2412	0.015	Supported <i>p</i> <0.05
	F_Super	3.90	1.44				
2B: Novelty	F_Sub	4.11	1.41	0.38	2412	0.71	Not Supported <i>p</i> >0.05
	F_Super	4.09	1.42				
3B: Diversity	F_Sub	4.07	1.44	1.06	2413	0.29	Not Supported <i>p</i> >0.05
	F_Super	4.01	1.45				

**Table 6.10:** Experiment Spring and Experiment Autumn: Results of Two Sample Student's t-Tests for F\_Sub and F\_Super

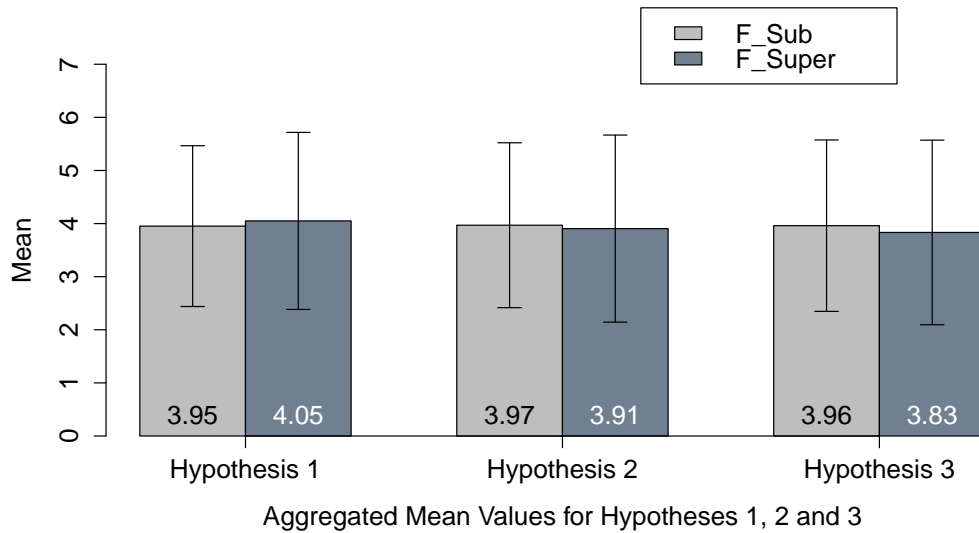
Relevance shows a significant difference in scores, however this is not supported in Experiment Spring. Thus, overall, the results suggest that FolkRank does not recommend more relevant, novel and diverse resources to sub-activities in contrast to the results of A\_Sub and A\_Super presented above.

## Limitations

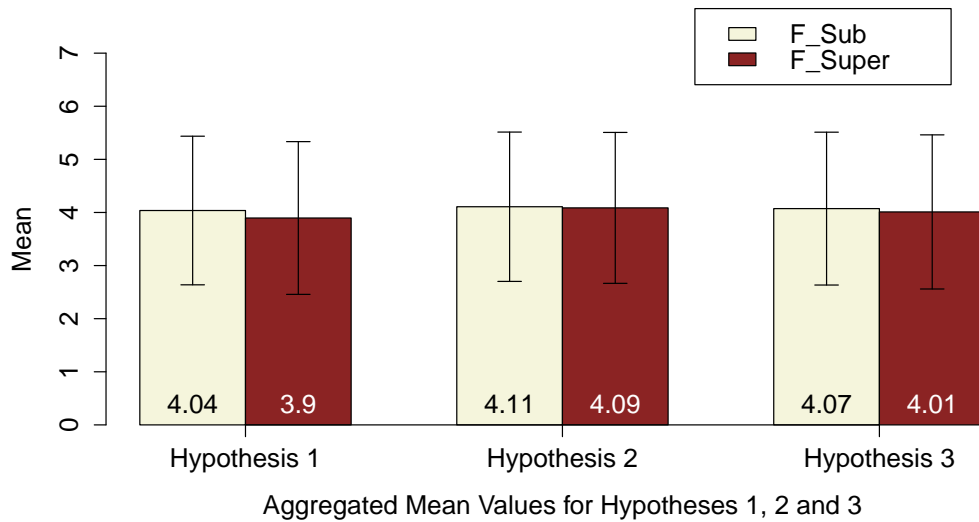
The proposed crowdsourcing concept faces several challenges. An efficient detection and filtering of spammers poses a major challenge and defining effective quality control measures such as control questions is crucial. Furthermore, suitable evaluation goals need to be defined that can be broken down into small compact tasks that are solvable online by crowdworkers, who generally want to accomplish these tasks quickly. It helps to pose simple, short questions to well-defined tasks that can be accomplished in a short period of time.

A further challenge is that crowdworkers, being publicly and randomly invited users, may have different motivations as the typical learners using the system. However, self-directed learning with resources found on the Web, as is the case in a resource-based learning scenario, can be argued as being an acceptable learning scenario to be evaluated using crowdsourcing, as the crowdworkers are potential learners on the Web.

The results from the t-tests for both Experiment Spring and Experiment Autumn show overall a small effect size ( $d < 0.3$ ) in Table C.14 in Appendix C.3, this indicates that the evaluation results from the baseline recommender algorithm FolkRank and the proposed hybrid recommender algorithm AScore, although significantly different are still very similar. This is due to the fact that AScore is fundamentally based on FolkRank and thus comparing results with FolkRank as a baseline poses a very challenging evaluation scenario.



(a) Experiment Spring: Results of Hypotheses 1B - 3B for F\_Sub and F\_Super



(b) Experiment Autumn: Results of Hypotheses 1B - 3B for F\_Sub and F\_Super

**Figure 6.8:** Experiment Spring and Experiment Autumn: Results of F\_Sub and F\_Super

	Hypothesis	Experiment Spring	Experiment Autumn
AScore vs. FolkRank	1A: Relevance	☒	☒
	2A: Novelty	☒	☒
	3A: Diversity	☒	☒
A_Sub vs. F_Sub	1A: Relevance	☒	☒
	2A: Novelty	☒	☒
	3A: Diversity	☒	☒
A_Super vs. F_Super	1A: Relevance	☐	☒
	2A: Novelty	☒	☒
	3A: Diversity	☒	☒
A_Sub vs. A_Super	1B: Relevance	☒	☒
	2B: Novelty	☒	☒
	3B: Diversity	☒	☒
F_Sub vs. F_Super	1B: Relevance	☐	☒
	2B: Novelty	☐	☐
	3B: Diversity	☐	☐

**Table 6.11:** Summary of Crowdsourcing Results

## 6.4 Summary

In this chapter, an evaluation concept based on crowdsourcing has been proposed as an alternative evaluation method for evaluating recommender systems for TEL. A proof-of-concept evaluation was conducted testing AScore for relevance, novelty and diversity. Results from a repeated proof-of-concept evaluation experiment shows that the algorithm AScore provides more relevant, novel and diverse recommendations than the state-of-the-art algorithm FolkRank. Additionally, AScore provides more relevant, novel and diverse recommendations to sub-activities than to activities higher up in the hierarchy thereby providing learners with more support the more precise their research becomes.

These results confirm that the crowdsourcing concept can be successfully applied as an alternative evaluation approach for TEL recommender algorithms, thereby combining the advantages of offline experiments and user studies. The results from the crowdsourcing experiments are summarized in Table 6.11. In conclusion, the results of both Experiment Spring and Experiment Autumn support all three hypotheses: Hypothesis 1: Relevance, Hypothesis 2: Novelty and Hypothesis 3: Diversity. The results from Experiment Autumn validate the evaluation concept and affirm that neither the choice of activities nor the selected recommendations for the experiments directly influence the results obtained. The repeated proof-of-concept evaluation experiments show that the results obtained are reproducible and the proposed crowdsourcing evaluation concept is indeed a promising evaluation method to evaluate TEL recommender algorithms.

---

## 7 Conclusion and Outlook

---

When learning with resources found on the Web, learners are faced with a lot of challenges. Learners often have no teacher nor any didactically prepared courses or learning resources to work with. On their own, learners have to find and select resources relevant to their needs amongst the vast amount of resources found on the Web. The aim of this thesis is thus to support resource-based learning by providing personalized recommendations of learning resources. This concluding chapter summarizes the main contributions and results of this thesis and gives an outlook on future work.

---

### 7.1 Main Contributions

---

The first research goal is to develop and evaluate personalized recommender algorithms that support resource-based learning. In particular, to provide personalized recommendations of learning resources that consider additional semantic information gained from a resource-based learning scenario, such as the current learning activity the learner is working on, the learner's and other learner's hierarchical activity structures in the community, as well as related learning resources, their tags and semantic tag types. To this aim, graph-based recommender approaches have been investigated as a basis for creating feature-enhanced hybrid graph-based recommender algorithms [27] considering additional semantic information gained from extended folksonomies. A detailed analysis of the application scenario is conducted where additional semantic information that could be exploited to improve graph-based recommendation approaches for TEL are identified. A definition of an extended folksonomy model for a RBL application scenario, taking CROKODIL as a concrete implemented example, is presented. A new concept to provide personalized recommendations of learning resources for a resource-based learning scenario is presented and integrated into CROKODIL as a proof-of-concept.

Several hybrid graph-based concepts to provide personalized recommendations of learning resources in a resource-based learning scenario are proposed, implemented and evaluated. AScore and AInheritscore focus on exploiting the activities and activity hierarchies in order to provide personalized recommendations of learning resources relevant to the activity the learner is currently working on. The activity hierarchies are exploited as an additional structure to improve the spread of weights through the folksonomy graph, thereby providing the learner with relevant recommendations of learning resources from sub-activities or activities higher up in the activity hierarchy. AspectScore exploits semantic tagging to recommend learning resources by giving preference to certain paths through the folksonomy determined by the semantic tag types. The semantic tag types of the learner are taken to represent the concepts or aspects of learning resources the learner is interested in, thereby defining the learning context of the learner. Hence by focusing on a certain tag type, AspectScore can provide personalized recommendations of learning resources relevant to the learning context of the learner. InteliScore utilizes the additional semantic information gained from the semantic relatedness between the concepts represented by the tags in the folksonomy to recommend learning resources belonging to similar concepts even when the learner does not know the exact terminology to describe these concepts and search for related resources. VSScore exploits the context-specific information gained from a folksonomy to provide recommendations of learning resources relevant to the learning context of the learner in the folksonomy. Thereby, the learning context of a user is represented as a vector space with the dimensions representing the entities in the folksonomy.

The proposed recommender approaches are evaluated using offline experiments on three historical datasets representing different evaluation scenarios including CROKODIL as a concrete resource-based learning scenario. The overall evaluation results from the offline experiments show that the proposed hybrid concepts are effective in recommending relevant learning resources when compared to other comparable baseline recommender algorithms. Thus, the exploitation of additional semantic information

---

is beneficial in providing more relevant recommendations in a resource-based learning scenario. The evaluation results are however dependent on the evaluation dataset, evaluation method and metrics used for the evaluation.

The second goal of this thesis is to investigate alternative evaluation approaches for recommender algorithms for resource-based learning. As more and more recommender systems are developed for TEL, it becomes increasingly important to evaluate their effectiveness in supporting learners. As a contribution to this thesis, a detailed survey of TEL evaluation methodologies is conducted in order to identify the evaluation methodologies used for TEL recommender systems over the years. An evaluation concept for hybrid graph-based recommender algorithms for extended folksonomies is presented along with an evaluation method LeaveRTOOut and an evaluation metric Mean Normalized Precision to complement the state-of-the-art evaluation method and metrics, thereby providing an alternative perspective on the evaluation performance of the algorithms in an offline experiment on a historical dataset.

Furthermore, in an effort to overcome the incompleteness problem faced by offline experiments and to be able to measure user-centric metrics such as novelty and diversity, a crowdsourcing evaluation concept is proposed as an alternative evaluation approach. Crowdsourcing retains the advantages of offline experiments, in giving access to sufficient participants, as well as being easy to conduct and repeat, and in addition enables the evaluation of user-centric metrics. A repeated crowdsourcing evaluation of the AScore algorithm is conducted as a proof-of-concept. Results from both runs of the crowdsourcing experiment support the postulated hypotheses that AScore is more effective than the state-of-the-art FolkRank in recommending more relevant, novel and diverse learning resources to a particular activity the learner is working on. In addition, AScore provides more relevant, novel and diverse recommendations to more specific topics in sub-activities lower in the hierarchy, thereby providing learners with more support the more detailed and precise their research becomes. The results from the repeat of the initial experiment further validate the evaluation concept and affirm that neither the choice of activities nor the selected recommendations for the experiments directly influence the results obtained. Hence, these results show that crowdsourcing can be successfully applied as an alternative approach for evaluating TEL recommender algorithms.

---

## 7.2 Outlook

---

The contributions of this thesis lay the foundation for further hybrid graph-based recommender approaches for TEL. It would be interesting to consider further sources of semantic information found in folksonomies that could be beneficial for other TEL scenarios. For example, as more and more information about social interactions and connections between learners are made available, this could be a future source of additional information that could be used to enhance personalized recommender systems [49, 136], particularly by enriching the folksonomy graph with social ties between users [30]. In social networks, there exist two types of social ties according to their social influence and strength in the network [89]: weak ties that indicate relationships to acquaintances and strong ties that indicate close relationships to friends and family. Sociological theories show that not only strong ties but also weak ties could have a great influence on the propagation of information in a social network [89]. Furthermore, Social Network Analysis (SNA) offers a key source of additional information about the importance or influence of a node or social tie in a social network [176]. This information could be used to introduce new links between users or to adjust the weights of links or nodes in a folksonomy graph, thereby influencing the weights spread between the nodes in the folksonomy graph [29]. Folksonomy datasets having information about the social network between users are however presently hard to gain access to due to privacy issues. As a result, the evaluation of such approaches is still very limited. However this area of research has the potential to become very interesting in future as more and more social network data becomes available [176].

Due to the resource-based learning application scenario considered in this thesis, the focus was made on the recommendation of learning resources. Learning resources are however only one possible entity of

---

a folksonomy that could be recommended to a learner [161]. Particularly in a TEL scenario, it would be beneficial to the learner to have other learners recommended that could become potential collaborators or to find learners with common pre-knowledge or trying to solve the same or similar problems [159]. Relevant and interesting learning activities could also be recommended to the learner. A challenge here would be the adaptation of the existing evaluation methods to the type of entity being recommended. The proposed evaluation method LeaveRTOut, which is a contribution to this thesis could give helpful insights into how a similar adaptation could be made.

The recommender algorithms in this thesis were conceived, implemented and evaluated for a resource-based learning scenario which presumes a small active online community of learners working together on common activities. Therefore the runtime speed of the algorithms, which is dependent on the size and complexity of the folksonomy graph, is neglected as this does not apply to the application scenario. In order to extend the methods presented in this work to other application scenarios as well as to broaden their applicability to large and diverse datasets, several improvements and extensions are possible. As the size and complexity of a folksonomy increases rather rapidly, especially in scenarios other than a TEL scenario [143], the evaluation of hybrid graph-based recommender algorithms would need to be optimized. A time-based evaluation approach [143] that focuses on the recent parts of a folksonomy, thereby reducing the size of the relevant part of the folksonomy to be considered during evaluation could be a potential solution to this challenge. A time-based folksonomy model would need to be defined, as well as the evaluation methods adapted to include the creation of a test and training partitions of the dataset based on the timestamps in the folksonomy. It would then be possible to create a new evaluation metric such as a novelty time-based metric that considers resources to be novel that are recommended only after a specified point in time.

Furthermore, based on the results and insights gained from the survey on evaluation methodologies of TEL recommender algorithms, research should not only focus on implementing improved recommender algorithms but this should go hand in hand with user-centric evaluations [45]. The evaluation of TEL recommender systems should be a joint effort between computer scientists and experts from other domains like pedagogics or psychology. The evaluation of a TEL recommender algorithm should be integrated into the design and development of the recommender algorithm. The evaluation of recommender algorithms for TEL could now be executed more often using the crowdsourcing evaluation concept proposed in this thesis, which offers an easier, faster evaluation approach and access to sufficient participants. In addition, different variations, parameters and settings could be tested as well as different combinations of hybrid recommender systems. Experiments can also be easily repeated and specific parts of the recommender system already tested even as early as in the requirements and design phases, thus enabling a comprehensive evaluation of the recommender system. As the recommender algorithm only makes up a part of a complete recommender system, it remains a challenge to investigate how crowdsourcing could be used to evaluate other aspects of a TEL recommender system, for example, the effects of the presentation of the recommendations or the usefulness of explanations of the recommendations made to the learner.

Learning on the Web has become a fundamental part of our everyday lives, and the challenges of learning anytime and everywhere on the Web will become even more demanding in future. Recommender systems will continue to play an increasingly important role in supporting learners to learn with and from others on the Web. Learners will no longer have to battle with the huge amount of information on the Web, but will rather have resources offered to them as personalized recommendations relevant to their current task and learning needs. This thesis contributes towards reaching this goal by exploiting particular semantic information specific to the recommendation scenario thereby increasing the relevance and effectiveness of the recommendations made. This thesis also highlights the importance of a fitting evaluation approach integrated into the implementation phases of a recommender algorithm, with the aim to develop recommender algorithms that provide recommendations better tailored to the needs of the end users.





---

## Bibliography

---

- [1] Fabian Abel. *Contextualization, User Modeling and Personalization in the Social Web*. PhD Thesis, Gottfried Wilhelm Leibniz Universität Hannover, 2011.
- [2] Fabian Abel, Mischa Frank, Nicola Henze, Daniel Krause, Daniel Plappert, and Patrick Siehndel. GroupMe! - Where Semantic Web Meets Web 2.0. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference on the Semantic Web*, pages 871–878. Springer, 2007.
- [3] Fabian Abel, Matteo Baldoni, Cristina Baroglio, Nicola Henze, Daniel Krause, and Viviana Patti. Context-Based Ranking in Folksonomies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, pages 209–218. ACM, 2009.
- [4] Fabian Abel, Ig Ibert Bittencourt, Evandro de Barros Costa, Nicola Henze, Daniel Krause, and Julita Vassileva. Recommendations in Online Discussion Forums for E-Learning Systems. *IEEE Transactions on Learning Technologies (TLT)*, 3(2):165–176, 2010.
- [5] Fabian Abel, Nicola Henze, Ricardo Kawase, and Daniel Krause. The Impact of Multifaceted Tagging on Learning Tag Relations and Search. In *Proceedings of the 7th Extended Semantic Web Conference on the Semantic Web: Research and Applications*, pages 90–105. Springer, 2010.
- [6] Fabian Abel, Nicola Henze, and Daniel Krause. Optimizing Search and Ranking in Folksonomy Systems by Exploiting Context Information. In *Web Information Systems and Technologies*, volume 45 of *Lecture Notes in Business Information Processing*, pages 113–127. Springer, 2010.
- [7] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
- [8] Gediminas Adomavicius and Alexander Tuzhilin. Context-Aware Recommender Systems. In *Recommender Systems Handbook*, chapter 7, pages 217–253. Springer, 2011.
- [9] Mario Aehnel, Mirko Ebert, Günter Beham, Stefanie N. Lindstaedt, and Alexander Paschen. A Socio-technical Approach towards Supporting Intra-organizational Collaboration. In *Times of Convergence. Technologies Across Learning Contexts*, volume 5192 of *Lecture Notes in Computer Science*, pages 33–38. Springer, 2008.
- [10] Hend S. Al-Khalifa and Hugh C. Davis. FAsTA: A Folksonomy-Based Automatic Metadata Generator. In *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 414–419. Springer, 2007.
- [11] Omar Alonso and Ricardo Baeza-Yates. Design and Implementation of Relevance Assessments Using Crowdsourcing. In *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2011.
- [12] Omar Alonso and Stefano Mizzaro. Using crowdsourcing for TREC relevance assessment. *Information Processing & Management*, 48(6):1053–1066, 2012.
- [13] Omar Alonso, Daniel E. Rose, and Benjamin Stewart. Crowdsourcing for Relevance Evaluation. *ACM SIGIR Forum*, 42(2):9–15, 2008.

- 
- [14] Mojisola Anjorin, Doreen Böhnstedt, and Christoph Rensing. Towards Graph-Based Recommendations for Resource-Based Learning Using Semantic Tag Types. In *DeLFI 2011: Die 9. e-Learning Fachtagung Informatik - Poster Workshops Kurzbeiträge*. TUDpress, 2011.
- [15] Mojisola Anjorin, Renato Domínguez García, and Christoph Rensing. CROKODIL: a platform supporting the collaborative management of web resources for learning purposes. In *Proceedings of the 16th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE '11*, pages 361–361. ACM, 2011.
- [16] Mojisola Anjorin, Christoph Rensing, Kerstin Bischoff, Christian Bogner, Lasse Lehmann, Anna Lenka Reger, Nils Faltin, Achim Steinacker, Andy Lüdemann, and Renato Domínguez García. CROKODIL - A Platform for Collaborative Resource-Based Learning. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2011.
- [17] Mojisola Anjorin, Christoph Rensing, and Ralf Steinmetz. Towards Ranking in Folksonomies for Personalized Recommender Systems in E-learning. In *Proceedings of the 2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation*, volume 781, pages 22–25. CEUR Workshop Proceedings, 2011.
- [18] Mojisola Anjorin, Ivan Dackiewicz, Alejandro Fernández, and Christoph Rensing. A Framework for Cross-Platform Graph-based Recommendations for TEL. In *Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2012)*, volume 896, pages 83–88. CEUR Workshop Proceedings, 2012.
- [19] Mojisola Anjorin, Thomas Rodenhausen, Renato Domínguez García, and Christoph Rensing. Exploiting Semantic Information for Graph-Based Recommendations of Learning Resources. In *21st Century Learning for 21st Century Skills*, volume 7563 of *Lecture Notes in Computer Science*, pages 9–22. Springer, 2012.
- [20] Sebastian Bab and Luise Kranich. On Self-adapting Recommendations of Curricula for an Individual Learning Experience. In *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 589–590. Springer, 2013.
- [21] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Publishing Company, 2nd edition, 2011. ISBN 978-0-321-41691-9.
- [22] Bakhtiyor Bahritidinov, Eduardo Sánchez, and Manuel Lama. Recommending Teachers for Collaborative Authoring Tools. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 438–442. IEEE Computer Society, 2011.
- [23] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing Web Search Using Social Annotations. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, pages 501–510. ACM, 2007.
- [24] Vincent Barré, Christophe Choquet, and Sébastien Iksal. Observation Scenario Development Using Recommendations. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 605–607. IEEE Computer Society, 2007.
- [25] Scott Bateman, Christopher Brooks, Gordon McCalla, and Peter Brusilovsky. Applying collaborative tagging to e-learning. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*. ACM, 2007.
- [26] Günter Beham, Barbara Kump, Tobias Ley, and Stefanie N. Lindstaedt. Recommending Knowledgeable People in a Work-Integrated Learning System. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2783–2792. Procedia Computer Science, 2010.

- 
- [27] Alejandro Bellogín, Iván Cantador, Fernando Díez, Pablo Castells, and Enrique Chavarriaga. An Empirical Comparison of Social, Collaborative Filtering, and Hybrid Recommenders. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):14:1–14:29, 2013.
- [28] Francesco Bellotti, Jaroslava Mikulecká, Linda Napoletano, and Hana Rohrova. Designing a Constructionistic Framework for T-Learning. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 549–554. Springer, 2006.
- [29] Clément Benaych. Exploiting Social Networks to Recommend Resources in Social Bookmarking Applications. Master Thesis, Technische Universität Darmstadt, 2013.
- [30] Kerstin Bischoff. We love rock’n’roll: analyzing and predicting friendship links in last. fm. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 47–56. ACM, 2012.
- [31] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can All Tags be Used for Search? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM ’08*, pages 193–202. ACM, 2008.
- [32] Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, Henry S. Thompson, and Thanh Tran Duc. Repeatable and Reliable Search System Evaluation Using Crowdsourcing. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’11*, pages 923–932. ACM, 2011.
- [33] Toine Bogers. *Recommender Systems for Social Bookmarking*. PhD Thesis, Tilburg University, 2009.
- [34] Christian Bogner. Lernen ohne Aufsicht. *Zeitschrift für E-Learning*, 1(Bd. 2009):8–22, 2009.
- [35] Doreen Böhnstedt, Philipp Scholl, Christoph Rensing, and Ralf Steinmetz. Collaborative Semantic Tagging of Web Resources on the Basis of Individual Knowledge Networks. In *User Modeling, Adaptation, and Personalization*, volume 5535 of *Lecture Notes in Computer Science*, pages 379–384. Springer, 2009.
- [36] Doreen Böhnstedt, Lasse Lehmann, Christoph Rensing, and Ralf Steinmetz. Automatic Identification of Tag Types in a Resource-based Learning Scenario. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 57–70. Springer, 2011.
- [37] Aldo Bongio, Jan van Bruggen, Stefano Ceri, Valentin Cristea, Peter Dolog, Andreas Hoffmann, Maristella Matera, Marzia Mura, Antonio Vincenzo Taddeo, Xuan Zhou, and Larissa Zoni. COOPER: Towards a Collaborative Open Environment of Project-Centred Learning. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 561–566. Springer, 2006.
- [38] Jorge Bozo, Rosa Alarcón, and Sebastian Iribarra. Recommending Learning Objects According to a Teachers’ Context Model. In *Sustaining TEL: From Innovation to Learning and Practice*, volume 6383 of *Lecture Notes in Computer Science*, pages 470–475. Springer, 2010.
- [39] Paul Bra, David Smits, and Natalia Stash. Creating and Delivering Adaptive Courses with AHA! In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 21–33. Springer, 2006.
- [40] Simone Braun, Claudiu Schora, and Valentin Zacharias. Semantics to the bookmarks: A review of social semantic bookmarking systems. In *Proceedings of the 5th I-SEMANTICS*, pages 445–454. Verlag der Technischen Universität Graz, 2009.
- [41] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

- 
- [42] Julien Broisin, Mihaela Brut, Valentin Butoianu, Florence Sedes, and Philippe Vidal. A Personalized Recommendation Framework based on CAM and Document Annotations. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2839–2848. Procedia Computer Science, 2010.
- [43] Peter Brusilovsky. Adaptive hypermedia. *User modeling and user-adapted interaction*, 11(1-2): 87–110, 2001.
- [44] Mihaela Brut and Florence Sedes. Ontology-Based Solution for Personalized Recommendations in E-Learning Systems. Methodological Aspects and Evaluation Criterias. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 469–471. IEEE Computer Society, 2010.
- [45] Jürgen Buder and Christina Schwind. Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior*, 28(1):207–216, 2012.
- [46] Robin Burke. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
- [47] Robin Burke. Hybrid Web Recommender Systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 377–408. Springer, 2007.
- [48] Iván Cantador, Ioannis Konstas, and Joemon M. Jose. Categorising Social Tags to Improve Folksonomy-Based Recommendations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1):1–15, 2011.
- [49] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har’El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized Social Search Based on the User’s Social Network. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM ’09*, pages 1227–1236. ACM, 2009.
- [50] Praveen Chandar and Ben Carterette. Using Preference Judgments for Novel Document Retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’12*, pages 861–870. ACM, 2012.
- [51] Ting-Wen Chang, Moushir M. El-Bishouty, Sabine Graf, and Kinshuk. Recommendation Mechanism Based on Students’ Working Memory Capacity in Learning Systems. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 333–335. IEEE Computer Society, 2013.
- [52] Mohamed Amine Chatti, Matthias Jarke, Theresia Devi Indriasari, and Marcus Specht. NetLearn: Social Network Analysis and Visualizations for Learning. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 310–324. Springer, 2009.
- [53] Mohamed Amine Chatti, Simona Dakova, Hendrik Thüs, and Ulrik Schroeder. Tag-Based Collaborative Filtering Recommendation in Personal Learning Environments. *IEEE Transactions on Learning Technologies (TLT)*, 6(4):337–349, 2013.
- [54] Jun-Ming Chen, Yeali S. Sun, and Meng Chang Chen. A Hybrid Tag-Based Recommendation Mechanism to Support Prior Knowledge Construction. In *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*, pages 23–25. IEEE Computer Society, 2012.
- [55] Weiqin Chen and Richard Persen. Reusing Collaborative Knowledge as Learning Objects -The Implementation and Evaluation of AnnForum. In *Times of Convergence. Technologies Across Learning Contexts*, volume 5192 of *Lecture Notes in Computer Science*, pages 92–103. Springer, 2008.



- 
- [56] Chen-Chung Chi and Chin-Hwa Kuo. The Design of English Article Recommender Mechanism for Senior High School Students. In *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*, pages 541–545. IEEE Computer Society, 2012.
- [57] Chen-Chung Chi, Chin-Hwa Kuo, and Chia-Chun Peng. The Designing of a Web Page Recommendation System for ESL. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 730–734. IEEE Computer Society, 2007.
- [58] Chen-Chung Chi, Chin-Hwa Kuo, Ming-Yuan Lu, and Nai-Lung Tsao. Concept-Based Pages Recommendation by Using Cluster Algorithm. In *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on*, pages 298–300. IEEE Computer Society, 2008.
- [59] Maarten Clements. Personalization of Social Media. In *Proceedings of the BCS IRSG Symposium: Future Directions in Information Access 2007*, 2007.
- [60] Paul De Bra, Ad Aerts, Bart Berden, Barend De Lange, Brendan Rousseau, Tomi Santic, David Smits, and Natalia Stash. AHA! The adaptive hypermedia architecture. In *Proceedings of the 14th ACM conference on Hypertext and hypermedia*, pages 81–84. ACM, 2003.
- [61] Christian Desrosiers and George Karypis. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook*, chapter 4, pages 107–144. Springer, 2011.
- [62] Irina Diaconita, Christoph Rensing, Stephan Tittel, et al. Context-aware question and answering for community-based learning. *DeLFI 2013–Die 11. e-Learning Fachtagung Informatik*, 2013.
- [63] Ernesto Diaz-Aviles, Marco Fisichella, Wolfgang Nejdl, Ricardo Kawase, and Avaré Stewart. Unsupervised Auto-tagging for Learning Object Enrichment. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 83–96. Springer, 2011.
- [64] Pierre Dillenbourg. What do you mean by collaborative learning? *Collaborative-learning: Cognitive and computational approaches*, pages 1–19, 1999.
- [65] Stephan Doerfel, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Leveraging Publication Metadata and Social Data into FolkRank for Scientific Publication Recommendation. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web*, RSWeb '12, pages 9–16. ACM, 2012.
- [66] Renato Domínguez García. *Unterstützung des Ressourcen-basierten Lernens in Online Communities - Automatische Erstellung von Grosstaxonomien in verschiedenen Sprachen*. PhD thesis, Technische Universität Darmstadt, 2013.
- [67] Renato Domínguez García, Philipp Scholl, and Christoph Rensing. Supporting Resource-based Learning on the Web Using Automatically Extracted Large-scale Taxonomies from Multiple Wikipedia Versions. In *Advances in Web-Based Learning*, volume 7048 of *Lecture Notes in Computer Science*, pages 314–319. Springer, 2011.
- [68] Renato Domínguez García, Matthias Bender, Mojisola Anjorin, Christoph Rensing, and Ralf Steinmetz. FReSET: an evaluation framework for folksonomy-based recommender systems. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web*, RSWeb '12, pages 25–28. ACM, 2012.
- [69] Hendrik Drachsler, Hans G. K. Hummel, and Rob Koper. Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. In *Proceedings of the 1st Workshop on Social Information Retrieval for Technology Enhanced Learning (SIRTEL07)*, volume 307, pages 18–26. CEUR Workshop Proceedings, 2007.

- 
- [70] Hendrik Drachsler, Hans Hummel, and Rob Koper. Using Simulations to Evaluate the Effects of Recommender Systems for Learners in Informal Learning Networks. In *Proceedings of the 2nd Workshop on Social Information Retrieval for Technology Enhanced Learning (SIRTEL'08)*, volume 382. CEUR Workshop Proceedings, 2008.
- [71] Hendrik Drachsler, Hans G. K. Hummel, and Rob Koper. Navigation Support for Learners in Informal Learning Environments. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 303–306. ACM, 2008.
- [72] Hendrik Drachsler, Hans Hummel, and Rob Koper. Identifying the Goal, User model and Conditions of Recommender Systems for Formal and Informal Learning. *Journal of Digital Information*, 10(2), 2009.
- [73] Hendrik Drachsler, Dries Pecceu, Tanja Arts, Edwin Hutten, Lloyd Rutledge, Peter van Rosmalen, Hans G. K. Hummel, and Rob Koper. ReMashed - Recommendations for Mash-Up Personal Learning Environments. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 788–793. Springer, 2009.
- [74] Hendrik Drachsler, Toine Bogers, Riina Vuorikari, Katrien Verbert, Erik Duval, Nikos Manouselis, Guenter Beham, Stephanie Lindstaedt, Hermann Stern, Martin Friedrich, et al. Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning. *Procedia Computer Science*, 1(2):2849–2858, 2010.
- [75] Erik Duval and Wayne Hodgins. A LOM Research Agenda. In *WWW (Alternate Paper Tracks)*, 2003.
- [76] Erik Duval, Eddy Forte, Kris Cardinaels, Bart Verhoeven, Rafael Van Durm, Koen Hendrikx, Maria Wentland Forte, Norbert Ebel, Maciej Macowicz, Ken Warkentyne, et al. The ariadne knowledge pool system. *Communications of the ACM*, 44(5):72–78, 2001.
- [77] Martin Ebner and Anja Lorenz. Web 2.0 als Basistechnologien für CSCL-Umgebungen. *CSCL-Kompodium*, 2:97–111, 2012.
- [78] Mojisola Erdt and Christoph Rensing. Evaluating Recommender Algorithms for Learning using Crowdsourcing. In *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on*, pages 513–517. IEEE Computer Society, 2014.
- [79] Mojisola Erdt, Florian Jomrich, Katja Schöler, and Christoph Rensing. Investigating Crowdsourcing as an Evaluation Method for TEL Recommender Systems. In *ECTEL meets ECSCW 2013: Workshop on Collaborative Technologies for Working and Learning*, volume 1047, pages 25–29. CEUR Workshop Proceedings, 2013.
- [80] Soude Fazeli, Hendrik Drachsler, Francis Brouns, and Peter Sloep. A Trust-based Social Recommender for Teachers. In *Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2012)*, volume 896, pages 49–60. CEUR Workshop Proceedings, 2012.
- [81] Alejandro Fernández, Mojisola Erdt, Ivan Dackiewicz, and Christoph Rensing. Recommendations from Heterogeneous Sources in a Technology Enhanced Learning Ecosystem. In *Recommender Systems for Technology Enhanced Learning*, pages 251–265. Springer, 2014.
- [82] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 25–32. ACM, 2011.

- 
- [83] Andy Field, Jeremy Miles, and Zoë Field. *Discovering Statistics Using R*. SAGE Publications, 2012. ISBN 978-14462-0046-9.
- [84] Beatriz Florian, Christian Glahn, Hendrik Drachsler, Marcus Specht, and Ramón Fabregat Gesa. Activity-Based Learner-Models for Learner Monitoring and Recommendations in Moodle. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 111–124. Springer, 2011.
- [85] Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness Using Wikipedia-Based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7 of *IJCAI '07*, pages 1606–1611. Morgan Kaufmann Publishers Inc., 2007.
- [86] Khairil Imran Bin Ghauth and Nor Aniza Abdullah. Building an E-learning Recommender System Using Vector Space Model and Good Learners Average Rating. In *Advanced Learning Technologies, 2009. ICAIT 2009. Ninth IEEE International Conference on*, pages 194–196. IEEE Computer Society, 2009.
- [87] Mario Gollwitzer and Reinhold S. Jäger. *Evaluation Kompakt*. Beltz Kompakt. Beltz PVU, 2009. ISBN 978-3-621-27758-7.
- [88] Monique Grandbastien, Suzana Loskovska, Samuel Nowakowski, and Jelena Jovanovic. Using online presence data for recommending human resources in the OP4L project. In *Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2012)*, volume 896, pages 89–94. CEUR Workshop Proceedings, 2012.
- [89] Mark Granovetter. The strength of weak ties: A network theory revisited. *Sociological theory*, 1 (1):201–233, 1983.
- [90] Volker Gries, Ulrike Lucke, and Djamshid Tavangarian. Werkzeuge zur Spezialisierung von XML-Sprachen für vereinfachte, didaktisch unterstützte Erstellung von E-Learning-Inhalten. In *Lernen im digitalen Zeitalter - DeLFI 2009, die 7. E-Learning-Fachtagung Informatik*, volume 153 of *Lecture Notes in Informatics*, pages 211–222. Gesellschaft für Informatik e.V., Köllen Verlag, 2009.
- [91] Zinan Guo and Jim E. Greer. Electronic Portfolios as a Means for Initializing Learner Models for Adaptive Tutorials. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 482–487. Springer, 2006.
- [92] Jörg Haake, Gerhard Schwabe, and Martin Wessner. *CSCL-Kompodium: Lehr-und Handbuch zum computerunterstützten, kooperativen Lernen*. Oldenbourg Verlag, 2nd edition, 2012.
- [93] Maryam Habibi and Andrei Popescu-Belis. Using crowdsourcing to compare document recommendation strategies for conversations. In *RecSys Workshop on Recommendation Utility Evaluation: Beyond RMSE, RUE 2012*, page 15. Idiap, 2012.
- [94] Michael J. Hannafin and Janette Hill. Resource-Based Learning. *Handbook of Research on Educational Communications and Technology*, pages 525–536, 2007.
- [95] Marek Hatala, Karen Tanenbaum, Ron Wakkary, Kevin Muise, Bardia Mohabbati, Greg Corness, Jim Budd, and Tom Loughin. Experience Structuring Factors Affecting Learning in Family Visits to Museums. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 37–51. Springer, 2009.
- [96] Taher H. Haveliwala. Topic-Sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 517–526. ACM, 2002.

- 
- [97] Sandy El Helou, Christophe Salzmann, Stéphane Sire, and Denis Gillet. The 3A contextual ranking system: simultaneously recommending actors, assets, and group activities. In *Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09*, pages 373–376. ACM, 2009.
- [98] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.
- [99] Jerry L. Hintze and Ray D. Nelson. Violin Plots: A Box Plot-Density Trace Synergism. *The American Statistician*, 52(2):181–184, 1998.
- [100] Stefan Hoermann, Tomas Hildebrandt, Christoph Rensing, and Ralf Steinmetz. ResourceCenter-A Digital Learning Object Repository with an Integrated Authoring Tool Set. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA*, pages 3453–3460, 2005.
- [101] Stefan Hoermann, Christoph Rensing, and Ralf Steinmetz. Wiederverwendung von Lernressourcen mittels Authoring by Aggregation im ResourceCenter. In *DeLFI 2005: 3. Deutsche e-Learning Fachtagung Informatik*, volume 66 of *Lecture Notes in Informatics*, pages 153–164. Gesellschaft für Informatik e.V., Köllen Verlag, 2005.
- [102] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A Social Bookmark and Publication Sharing System. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102, 2006.
- [103] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information Retrieval in Folksonomies: Search and Ranking. In *Proceedings of the 3rd European Semantic Web Conference on the Semantic Web: Research and Applications*, pages 411–426. Springer, 2006.
- [104] Ching-Kun Hsu, Chih-Kai Chang, and Gwo-Jen Hwang. Development of a Reading Material Recommendation System Based on a Multi-expert Knowledge Acquisition Approach. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pages 273–277. IEEE Computer Society, 2009.
- [105] Ching-Kun Hsu, Gwo-Jen Hwang, and Chih-Kai Chang. Development of a reading material recommendation system based on a knowledge engineering approach. *Computers & Education*, 55(1):76–83, 2010.
- [106] Ching-Kun Hsu, Gwo-Jen Hwang, and Chih-Kai Chang. A personalized recommendation-based mobile learning approach to improving the reading performance of EFL students. *Computers & Education*, 63(0):327–336, 2013.
- [107] Lantao Hu, Zhao Du, Qiuli Tong, and Yongqi Liu. Context-Aware Recommendation of Learning Resources Using Rules Engine. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 181–183. IEEE Computer Society, 2013.
- [108] Oswald Huber. *Das psychologische Experiment: eine Einführung*. Huber, 5 edition, 2009. ISBN 978-3-456-84707-8.
- [109] Teresa Hurley and Stephan Weibelzahl. Using MotSaRT to Support On-Line Teachers in Student Motivation. In *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 101–111. Springer, 2007.
- [110] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 2010. ISBN 978-0-5214-9336-9.



- 
- [111] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag Recommendations in Folksonomies. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 506–514. Springer, 2007.
- [112] Glen Jeh and Jennifer Widom. SimRank: A Measure of Structural-Context Similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 538–543. ACM, 2002.
- [113] Zoran Jeremic, Jelena Jovanovic, and Dragan Gasevic. Towards a Semantic-Rich Collaborative Environment for Learning Software Patterns. In *Times of Convergence. Technologies Across Learning Contexts*, volume 5192 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2008.
- [114] Zoran Jeremic, Jelena Jovanovic, Dragan Gasevic, and Marek Hatala. Project-Based Collaborative Learning Environment with Context-Aware Educational Services. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 441–446. Springer, 2009.
- [115] Ajita John and Dorée Seligmann. Collaborative Tagging and Expertise in the Enterprise. In *Collaborative Web Tagging Workshop in conjunction with WWW2006*, 2006.
- [116] Roger T. Johnson and David W. Johnson. Cooperative learning: two heads learn better than one. *Context [periódico na internet]*, 1988.
- [117] Florian Jomrich. Crowdsourcing as an Online Evaluation Method for Recommender Systems for E-learning. Bachelor Thesis, Technische Universität Darmstadt, 2013.
- [118] Paul-Thomas Kandzia, Serge Linckels, Thomas Ottmann, and Stephan Trahasch. Lecture Recording—a Success Story. *it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, 55(3):115–122, 2013.
- [119] Ricardo Kawase, Bernardo Pereira Nunes, and Patrick Siehndel. Content-based Movie Recommendation within Learning Contexts. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 171–173. IEEE Computer Society, 2013.
- [120] Gabriella Kazai. In Search of Quality in Crowdsourcing for Search Engine Evaluation. In *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 165–176. Springer, 2011.
- [121] Mohamed Koutheaïr Khribi, Mohamed Jemni, and Olfa Nasraoui. Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. In *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on*, pages 241–245. IEEE Computer Society, 2008.
- [122] Sung-Hee Kim, Hyokun Yun, and Ji Soo Yi. How to Filter out Random Clickers in a Crowdsourcing-based Study? In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, pages 15:1–15:7. ACM, 2012.
- [123] Uwe Kirschenmann, Maren Scheffel, Martin Friedrich, Katja Niemann, and Martin Wolpers. Demands of Modern PLEs and the ROLE Approach. In *Sustaining TEL: From Innovation to Learning and Practice*, volume 6383 of *Lecture Notes in Computer Science*, pages 167–182. Springer, 2010.
- [124] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456. ACM, 2008.

- 
- [125] Ralf Klamma, Marc Spaniol, and Yiwei Cao. Community Aware Content Adaptation for Mobile Technology Enhanced Learning. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 227–241. Springer, 2006.
- [126] Aleksandra Klašnja-Milićević, Alexandros Nanopoulos, and Mirjana Ivanović. Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 33(3):187–209, 2010.
- [127] Aleksandra Klašnja-Milićević, Boban Vesin, Mirjana Ivanović, and Zoran Budimac. E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3):885–899, 2011.
- [128] Aleksandra Klašnja-Milićević, Boban Vesin, Mirjana Ivanović, and Zoran Budimac. Personalisation of Programming Tutoring System Using Tag-Based Recommender Systems. In *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*, pages 666–667. IEEE Computer Society, 2012.
- [129] Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.
- [130] Bart P. Knijnenburg. Conducting User Experiments in Recommender Systems. In *Proceedings of the 6th ACM Conference on Recommender Systems, RecSys '12*, pages 3–4. ACM, 2012.
- [131] Bart P. Knijnenburg, Martijn C. Willemsen, and Alfred Kobsa. A Pragmatic Procedure to Support the User-centric Evaluation of Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender Systems, RecSys '11*, pages 321–324. ACM, 2011.
- [132] Knowledge and Data Engineering Group. University of Kassel: Benchmark Folksonomy Data from BibSonomy, 2011. URL <http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/2011-07-01.tgz>. [Accessed Dec. 2, 2013].
- [133] Johannes Konert, Dmitrij Burlak, Stefan Göbel, and Ralf Steinmetz. GroupAL: ein Algorithmus zur Formation und Qualitätsbewertung von Lerngruppen in E-Learning-Szenarien mittels n-dimensionaler Gütekriterien. *DeLFI 2013–Die 11. e-Learning Fachtagung Informatik*, pages 71–82, 2013.
- [134] Johannes Konert, Nico Gerwien, Stefan Göbel, and Ralf Steinmetz. Bringing game achievements and community achievements together. In *Proceedings of the 7th European Conference on Games Based Learning (ECGBL)*, pages 319–328. Academic Conferences International, Academic Bookshop, 2013.
- [135] Xi Kong, Susanne Boll, and Wilko Heuten. A Hybrid Multi-recommender System for a Teaching and Learning Community for the Dual System of Vocational Education and Training. In *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 613–614. Springer, 2013.
- [136] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202. ACM, 2009.
- [137] Antonis Koukourikos, John Stoitsis, Pythagoras Karampiperis, and Pythagoras Karampiperis. Sentiment Analysis: A tool for Rating Attribution to Content in Recommender Systems. In *Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2012)*, volume 896, pages 61–70. CEUR Workshop Proceedings, 2012.

- 
- [138] Artus Krohn-Grimberghe, Andre Busche, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Active Learning for Technology Enhanced Learning. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 512–518. Springer, 2011.
- [139] Marius Kubatz, Fatih Gedikli, and Dietmar Jannach. LocalRank - Neighborhood-Based, Fast Computation of Tag Recommendations. In *Proceedings of the 12th International Conference on E-Commerce and Web Technologies*, pages 258–269. Springer, 2011.
- [140] Pablo Lachmann and Andreas Kiefel. Recommending Learning Activities as Strategy for Enabling Self-Regulated Learning. In *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*, pages 704–705. IEEE Computer Society, 2012.
- [141] Nikolas Landia, Sarabjot Singh Anand, Andreas Hotho, Robert Jäschke, Stephan Doerfel, and Folke Mitzlaff. Extending FolkRank with content data. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web*, RSWeb '12, pages 1–8. ACM, 2012.
- [142] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006. ISBN 978-0-691-12202-1.
- [143] Neal Kiritkumar Lathia. *Evaluating collaborative filtering over time*. PhD thesis, University College London, 2010.
- [144] Jean Lave and Etienne Wenger. *Situated Learning: Legitimate Peripheral Participation*. Learning in Doing: Social, Cognitive and Computational Perspectives. Cambridge University Press, 1991. ISBN 9780521423748.
- [145] Tobias Ley and Barbara Kump. Which User Interactions Predict Levels of Expertise in Work-Integrated Learning? In *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 178–190. Springer, 2013.
- [146] Stefanie N. Lindstaedt, Günter Beham, Barbara Kump, and Tobias Ley. Getting to Know Your User - Unobtrusive User Model Maintenance within Work-Integrated Learning Environments. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2009.
- [147] Robert Lokaiczky, Eicke Godehardt, Andreas Faatz, Manuel Görtz, Andrea Kienle, Martin Wessner, and Armin Ulbrich. Exploiting Context Information for Identification of Relevant Experts in Collaborative Workplace-Embedded E-Learning Environments. In *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 217–231. Springer, 2007.
- [148] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*, chapter 3, pages 73–105. Springer, 2011.
- [149] Ulrike Lucke. Design eines pervasiven Lernspiels für Studienanfänger. In *DeLFI 2011 - Die 9. e-Learning Fachtagung Informatik*, volume 188 of *Lecture Notes in Informatics*, pages 103–114. Gesellschaft für Informatik e.V., Köllen Verlag, 2011.
- [150] Ulrike Lucke and Christoph Rensing. A survey on pervasive education. *Pervasive and Mobile Computing*, 2013.
- [151] Valentina Maccatrozzo, Lora Aroyo, and Willem Robert van Hage. Crowdsourced Evaluation of Semantic Patterns for Recommendations. In *Late-Breaking Results, Project Papers and Workshop Proceedings of the 21st Conference on User Modeling, Adaptation, and Personalization*, volume 997 of *UMAP 2013 Extended Proceedings*. CEUR Workshop Proceedings, 2013.

- 
- [152] Eleni E. Mangina and John Kilbride. Evaluation of keyphrase extraction algorithm and tiling process for a document/resource recommender within e-learning environments. *Computers & Education*, 50(3):807–820, 2008.
- [153] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN 978-0-521-86571-5.
- [154] Nikos Manouselis and Constantina Costopoulou. Designing a Web-based Testing Tool for Multi-Criteria Recommender Systems. *Engineering Letters, Special Issue on Web Engineering*, 13(3), 2006.
- [155] Nikos Manouselis, Riina Vuorikari, and Frans Van Assche. Simulated Analysis of MAUT Collaborative Filtering for Learning Object Recommendation. In *Proceedings of the 1st Workshop on Social Information Retrieval for Technology Enhanced Learning (SIRTEL07)*, volume 307, pages 17–20. CEUR Workshop Proceedings, 2007.
- [156] Nikos Manouselis, Riina Vuorikari, and Frans Van Assche. Collaborative recommendation of e-learning resources: an experimental investigation. *Journal of Computer Assisted Learning*, 26(4): 227–242, 2010.
- [157] Nikos Manouselis, Hendrik Drachsler, Riina Vuorikari, Hans Hummel, and Rob Koper. Recommender Systems in Technology Enhanced Learning. In *Recommender Systems Handbook*, chapter 12, pages 387–415. Springer, 2011.
- [158] Nikos Manouselis, Giorgos Kyrgiazos, John Stoitsis, and John Stoitsis. Revisiting the Multi-Criteria Recommender System of a Learning Portal. In *Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2012)*, volume 896, pages 35–48. CEUR Workshop Proceedings, 2012.
- [159] Nikos Manouselis, Hendrik Drachsler, Katrien Verbert, and Erik Duval. *Recommender Systems for Learning*. Springer, 2013. ISBN 978-1-4614-4361-2.
- [160] Leandro Balby Marinho, Alexandros Nanopoulos, Lars Schmidt-Thieme, Robert Jäschke, Andreas Hotho, and Panagiotis Symeonidis Stumme, Gerd. Social Tagging Recommender Systems. In *Recommender Systems Handbook*, chapter 19, pages 615–644. Springer, 2011.
- [161] Leandro Balby Marinho, Andreas Hotho, Robert Jäschke, Alexandros Nanopoulos, Steffen Rendle, Lars Schmidt-Thieme, and Panagiotis Symeonidis Stumme, Gerd. *Recommender Systems for Social Tagging Systems*. Springer, 2012. ISBN 978-1-4614-1893-1.
- [162] Olga Marino and Gilbert Paquette. A competency - driven advisor system for multi-actor learning environments. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2871–2876. Procedia Computer Science, 2010.
- [163] Estefanía Martín, Rosa M. Carro, and Pilar Rodríguez. A Mechanism to Support Context-Based Adaptation in M-Learning. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 302–315. Springer, 2006.
- [164] Winter Mason and Duncan J. Watts. Financial Incentives and the "Performance of Crowds". *ACM SIGKDD Explorations Newsletter*, 11(2):100–108, 2010.
- [165] Gord McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education*, 2004(1), 2004.

- 
- [166] Sean M. McNee, John Riedl, and Joseph A. Konstan. Making Recommendations Better: An Analytic Model for Human-recommender Interaction. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1103–1108. ACM, 2006.
- [167] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101. ACM, 2006.
- [168] Florian Mehm. Authoring of Adaptive Single-Player Educational Games. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 37(2):157–160, 2014.
- [169] Florian Mehm, Stefan Göbel, and Ralf Steinmetz. An Authoring Tool for Educational Adventure Games: Concept, Game Models and Authoring Processes. *International Journal of Game-Based Learning (IJGBL)*, 3(1):63–79, 2013.
- [170] Friedrich Meincke, Ulrike Lucke, and Djamshid Tavangarian. Empfehlungen zur Nutzung eines Textverarbeitungswerkzeugs zur Erstellung von XML-basierten eLearning-Inhalten. In *DeLFI 2011 - Die 9. e-Learning Fachtagung Informatik*, volume 188 of *Lecture Notes in Informatics*, pages 9–20. Gesellschaft für Informatik e.V., Köllen Verlag, 2011.
- [171] Christos Mettouris, Achilleas Achilleos, and George Angelos Papadopoulos. A Context Modelling System and Learning Tool for Context-Aware Recommender Systems. In *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 619–620. Springer, 2013.
- [172] Marek Meyer, Christoph Rensing, and Ralf Steinmetz. Multigranularity Reuse of Learning Resources. *ACM Transactions on Multimedia Computing, Communications and Applications*, 7(1):1:1–1:23, 2011.
- [173] Pavel Michlík and Mária Bieliková. Exercises recommending for limited time learning. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2821–2828. Procedia Computer Science, 2010.
- [174] Markus Migenda. Supporting Resource-based Learning using Semantic Graph-based Recommendations. Master Thesis, Technische Universität Darmstadt, 2013.
- [175] Markus Migenda, Mojisola Erdt, Michael Gutjahr, and Christoph Rensing. Semantische graph-basierte empfehlungen zur unterstützung des ressourcen-basierten lernens. In *Proceedings der Pre-Conference Workshops der 11. e-Learning Fachtagung Informatik-DeLFI 2013*. Logos Verlag, 2013.
- [176] Peter Mika. Ontologies are us: A Unified Model of Social Networks and Semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5:5–15, 2007.
- [177] Felix Mödritscher. Towards a Recommender Strategy for Personal Learning Environments. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2775–2782. Procedia Computer Science, 2010.
- [178] Sasa Nesic, Dragan Gasevic, and Mehdi Jazayeri. An Ontology-Based Framework for Authoring Assisted by Recommendation. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 227–231. IEEE Computer Society, 2007.
- [179] Paul Nicholson. A history of e-learning. In *Computers and education*, pages 1–11. Springer, 2007.
- [180] Katja Niemann and Martin Wolpers. Usage Context-Boosted Filtering for Recommender Systems in TEL. In *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 246–259. Springer, 2013.



- 
- [181] Athitaya Nitchot, Lester Gilbert, and Gary B. Wills. Towards a Competence Based System for Recommending Study Materials (CBSR). In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 629–631. IEEE Computer Society, 2010.
- [182] Xavier Ochoa and Erik Duval. Relevance Ranking Metrics for Learning Objects. *IEEE Transactions on Learning Technologies (TLT)*, 1(1):34–48, 2008.
- [183] Xavier Ochoa, Stefaan Ternier, Gonzalo Parra, and Erik Duval. A Context-Aware Service Oriented Framework for Finding, Recommending and Inserting Learning Objects. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 697–702. Springer, 2006.
- [184] Marc-Andre Orthmann, Martin Friedrich, Uwe Kirschenmann, Katja Niemann, Maren Scheffel, Hans-Christian Schmitz, and Martin Wolpers. Usage-based Clustering of Learning Objects for Recommendation. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 553–557. IEEE Computer Society, 2011.
- [185] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 66, Stanford InfoLab, 1999.
- [186] Isabella Peters. *Folksonomies: Indexing and Retrieval in Web 2.0*. Knowledge & Information. De Gruyter Saur, 2009. ISBN 978-3-598-44185-1.
- [187] Alex Pongpech, Maria E. Orlowska, and Shazia W. Sadiq. Personalized Courses Recommendation Functionality for Flex-eL. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 631–633. IEEE Computer Society, 2007.
- [188] Vlad Posea and Stefan Trausan-Matu. Bringing the Social Semantic Web to the Personal Learning Environment. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 148–150. IEEE Computer Society, 2010.
- [189] Tiago Thompsen Primo and Rosa Maria Vicari. A Recommender System that Allows Reasoning and Interoperability over Educational Content Metadata. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 598–599. IEEE Computer Society, 2011.
- [190] Pearl Pu, Li Chen, and Rong Hu. A User-centric Evaluation Framework for Recommender Systems. In *Proceedings of the 5th ACM Conference on Recommender Systems, RecSys '11*, pages 157–164. ACM, 2011.
- [191] Glenda Rakes. Using the Internet as a Tool in a Resource-Based Learning Environment. *Educational Technology*, 36(5):52–56, 1996.
- [192] Maryam Ramezani. *Using Data Mining for Facilitating User Contributions in the Social Semantic Web*. PhD Thesis, Karlsruher Institut für Technologie, 2011.
- [193] Vinicius Faria Culmant Ramos, Paul De Bra, and Geraldo Xexéo. Qualitative and Quantitative Evaluation of an Adaptive Course in GALE. In *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 301–313. Springer, 2013.
- [194] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-Validation. In *Encyclopedia of Database Systems*, pages 532–538. Springer, 2009.
- [195] Christoph Rensing. MOOCs – Bedeutung von Massive Open Online Courses für die Hochschullehre. *PIK-Praxis der Informationsverarbeitung und Kommunikation*, 36(2):141–145, 2013.

- 
- [196] Christoph Rensing and Doreen Böhnstedt. Informelles, Ressourcen-basiertes Lernen. *i-com Zeitschrift für interaktive und kooperative Medien*, 11(1):15–18, 2012.
- [197] Christoph Rensing, Philipp Scholl, Doreen Böhnstedt, and Ralf Steinmetz. Recommending and Finding Multimedia Resources in Knowledge Acquisition Based on Web Resources. In *Proceedings of the 19th International Conference on Computer Communications and Networks*, pages 1–6, 2010.
- [198] Christoph Rensing, Christian Bogner, Thomas Prescher, Renato Domínguez García, and Mojisola Anjorin. Aufgabenprototypen zur Unterstützung der Selbststeuerung im Ressourcen-basierten Lernen. In *DeLFI 2011 - Die 9. e-Learning Fachtagung Informatik*, volume 188 of *Lecture Notes in Informatics*, pages 151–162. Gesellschaft für Informatik e.V., Köllen Verlag, 2011.
- [199] Christoph Rensing, Stephan Tittel, and Mojisola Anjorin. Location based Learning Content Authoring and Content Access in the docendo platform. In *PerCom-WORKSHOPS 2011: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 165–170. IEEE eXpress Conference Publishing, 2011.
- [200] Christoph Rensing, Andy Lüdemann, Birgit Stübing, and Frederick Schulz. Erfahrungen in der Gestaltung und Umsetzung von selbstgesteuerten Ressourcen-basierten Lernszenarien in der betrieblichen Aus-und Weiterbildung. In *Workshop zu Web2.0 in der beruflichen Weiterbildung im Rahmen der DeLFI 2012*, pages 1–6. FernUniversität, Hagen, 2012.
- [201] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to Recommender Systems Handbook*. Springer, 2011.
- [202] Thomas Rodenhausen. Ranking Resources in Folksonomies by Exploiting Semantic and Context-specific Information. Master Thesis, Technische Universität Darmstadt, 2012.
- [203] Thomas Rodenhausen, Mojisola Anjorin, Renato Domínguez, Christoph Rensing, and Ralf Steinmetz. Ranking Resources in Folksonomies by Exploiting Semantic Information. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies, i-KNOW '12*, pages 11:1–11:8. ACM, 2012.
- [204] Thomas Rodenhausen, Mojisola Anjorin, Renato Domínguez García, and Christoph Rensing. Context Determines Content: An Approach to Resource Recommendation in Folksonomies. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web, RSWeb '12*, pages 17–24. ACM, 2012.
- [205] Daniel Rodriguez-Cerezo, Mercedes Gómez-Albarrán, and José Luis Sierra. Supporting Self-Regulated Learning in Technical Domains with Repositories of Learning Objects and Recommender Systems. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 613–614. IEEE Computer Society, 2011.
- [206] Cristóbal Romero, Sebastián Ventura, Jose Antonio Delgado, and Paul De Bra. Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems. In *Creating New Learning Experiences on a Global Scale*, volume 4753 of *Lecture Notes in Computer Science*, pages 292–306. Springer, 2007.
- [207] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. Who Are the Crowdworkers?: Shifting Demographics in Mechanical Turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10*, pages 2863–2872. ACM, 2010.
- [208] Almudena Ruiz-Iniesta, Guillermo Jiménez-Díaz, and Mercedes Gómez-Albarrán. User-Adaptive Recommendation Techniques in Repositories of Learning Objects: Combining Long-Term and Short-Term Learning Goals. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 645–650. Springer, 2009.



- 
- [209] Almudena Ruiz-Iniesta, Guillermo Jiménez-Díaz, and Mercedes Gómez-Albarrán. Recommendation in Repositories of Learning Objects: A Proactive Approach that Exploits Diversity and Navigation-by-Proposing. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pages 543–545. IEEE Computer Society, 2009.
- [210] Mojtaba Salehi, Isa Nakhai Kamalabadi, and Mohammad Bagher Ghaznavi Ghouschi. An Effective Recommendation Framework for Personal Learning Environments Using a Learner Preference Tree and a GA. *IEEE Transactions on Learning Technologies (TLT)*, 6(4):350–363, 2013.
- [211] Olga C. Santos. A Recommender System to Provide Adaptive and Inclusive Standard-based Support Along the Elearning Life Cycle. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 319–322. ACM, 2008.
- [212] Olga C. Santos and Jesus Boticario. Meaningful Pedagogy Via Covering the Entire Life Cycle of Adaptive eLearning in Terms of a Pervasive Use of Educational Standards: The aLFanet Experience. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 691–696. Springer, 2006.
- [213] Olga C. Santos and Jesus Boticario. Guiding Learners in Learning Management Systems through Recommendations. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 596–601. Springer, 2009.
- [214] Olga C. Santos and Jesus Boticario. Modeling recommendations for the educational domain. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2793–2800. Procedia Computer Science, 2010.
- [215] Olga C. Santos, Jorge Couchet, and Jesus Boticario. Personalized E-learning and E-mentoring through User Modelling and Dynamic Recommendations for the Inclusion of Disabled at Work and Education. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pages 514–518. IEEE Computer Society, 2009.
- [216] Javier Sanz-Rodriguez, Juan Manuel Doderó, and Salvador Sánchez Alonso. Ranking Learning Objects through Integration of Different Quality Indicators. *IEEE Transactions on Learning Technologies (TLT)*, 3(4):358–363, 2010.
- [217] Andreas P. Schmidt and Simone Braun. Context-Aware Workplace Learning Support: Concept, Experiences, and Remaining Challenges. In *Innovative Approaches for Learning and Knowledge Sharing*, volume 4227 of *Lecture Notes in Computer Science*, pages 518–524. Springer, 2006.
- [218] Sebastian Schmidt, Philipp Scholl, Christoph Rensing, and Ralf Steinmetz. Cross-Lingual Recommendations in a Resource-Based Learning Scenario. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 356–369. Springer, 2011.
- [219] Bernhard Schmitz and Bettina S Wiese. New perspectives for the evaluation of training sessions in self-regulated learning: Time-series analyses of diary data. *Contemporary Educational Psychology*, 31(1):64–96, 2006.
- [220] Karin Schoefegger, Paul Seitlinger, and Tobias Ley. Towards a User Model for Personalized Recommendations in Work-Integrated Learning: A Report on an Experimental Study with a Collaborative Tagging System. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2829–2838. Procedia Computer Science, 2010.
- [221] Philipp Scholl, Doreen Mann, Christoph Rensing, and Ralf Steinmetz. Support of Acquisition and Organization of Knowledge Artifacts in Informal Learning Contexts. In *European Distance and E-Learning Network*, editors, *EDEN - Book of Abstracts*, page 16, 2007.

- 
- [222] Philipp Scholl, Doreen Böhnstedt, Renato Domínguez García, Christoph Rensing, and Ralf Steinmetz. Extended Explicit Semantic Analysis for Calculating Semantic Relatedness of Web Resources. In *Sustaining TEL: From Innovation to Learning and Practice*, volume 6383 of *Lecture Notes in Computer Science*, pages 324–339. Springer, 2010.
- [223] Svenja Schröder, Sabrina Ziebarth, Nils Malzahn, and Heinz Ulrich Hoppe. Self-profiling of Competences for the Digital Media Industry: An Exploratory Study. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 365–378. Springer, 2009.
- [224] Christina Schwind, Jürgen Buder, Ulrike Cress, and Friedrich W. Hesse. Preference-inconsistent recommendations: An effective approach for reducing confirmation bias and stimulating divergent thinking? *Computers & Education*, 58(2):787–796, 2012.
- [225] Cornelia Seeberg. *Life long learning: modulare Wissensbasen für elektronische Lernumgebungen*. Springer DE, 2002.
- [226] Cornelia Seeberg, Achim Steinacker, Klaus Reichenberger, Abdulmotaleb El Saddik, Stephan Fischer, and Ralf Steinmetz. From the user’s needs to adaptive documents. In *Proceedings of the Conference on Integrated Design and Process Technology*, 2000.
- [227] Guy Shani and Asela Gunawardana. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, chapter 8, pages 257–297. Springer, 2011.
- [228] Mark Sheehan and Young Park. pGPA: A Personalized Grade Prediction Tool to Aid Student Success. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys ’12, pages 309–310. ACM, 2012.
- [229] Brett E. Shelton, Joel Duffin, Yuxuan Wang, and Justin Bal. Linking OpenCourseWares and open education resources: creating an effective search and recommendation system. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2865 – 2870. Procedia Computer Science, 2010.
- [230] Miguel-Ángel Sicilia, Elena García Barriocanal, Salvador Sánchez Alonso, and Cristian Cechinel. Exploring user-based recommender results in large learning object repositories: the case of MERLOT. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2859–2864. Procedia Computer Science, 2010.
- [231] Rory L. L. Sie, Marlies Bitter-Rijpkema, and Peter B. Sloep. A Simulation for Content-based and Utility-based Recommendation of Candidate Coalitions in Virtual Creativity Teams. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2883–2888. Procedia Computer Science, 2010.
- [232] Sven Strickroth and Niels Pinkwart. High quality recommendations for small communities: the case of a regional parent network. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys ’12, pages 107–114. ACM, 2012.
- [233] Nguyen Thai-Nghe, Lucas Drumond, Artus Krohn-Grimberghe, and Lars Schmidt-Thieme. Recommender System for Predicting Student Performance. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2811–2819. Procedia Computer Science, 2010.
- [234] Mike Thelwall. *Link Analysis: An Information Science Approach*. Elsevier Academic Press, 2004. ISBN 0-120-88553-0.

- 
- [235] Chi-Cheng Tsai, Ching-I Chung, Yi-Ting Huang, Chia-Hsing Shen, Yu-Chieh Wu, and Jie-Chi Yang. VCSR: Video Content Summarization for Recommendation. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 862–864. IEEE Computer Society, 2007.
- [236] Kun Hua Tsai, Tung-Cheng Hsieh, Ti Kai Chiu, Ming-Che Lee, and Tzone I. Wang. Automated Course Composition and Recommendation based on a Learner Intention. In *Advanced Learning Technologies, 2007. ICALT 2007. Seventh IEEE International Conference on*, pages 274–278. IEEE Computer Society, 2007. ISBN 978-0-7695-2916-5.
- [237] Ekaterina Vasilyeva, Paul De Bra, and Mykola Pechenizkiy. Immediate Elaborated Feedback Personalization in Online Assessment. In *Times of Convergence. Technologies Across Learning Contexts*, volume 5192 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2008.
- [238] Katrien Verbert, Hendrik Drachsler, Nikos Manouselis, Martin Wolpers, Riina Vuorikari, and Erik Duval. Dataset-driven research for improving recommender systems for learning. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge, LAK '11*, pages 44–53. ACM, 2011.
- [239] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. *IEEE Transactions on Learning Technologies (TLT)*, 5(4):318–335, 2012.
- [240] Katrien Verbert, Xavier Ochoa, Michael Derntl, Martin Wolpers, Abelardo Pardo, and Erik Duval. Semi-automatic assembly of learning resources. *Computers & Education*, 59(4):1257–1272, 2012.
- [241] Lev S. Vygotsky. *Mind in Society*. Cambridge, MA: Harvard University Press, 1978.
- [242] Xin Wan, Toshie Ninomiya, and Toshio Okamoto. LRMDCR: A Learner’s Role-Based Multi Dimensional Collaborative Recommendation for Group Learning Support. In *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on*, pages 603–605. IEEE Computer Society, 2008.
- [243] Pei-Yu Wang and Hui-Chun Yang. Using collaborative filtering to support college students’ use of online forum for english learning. *Computers & Education*, 59(2):628–637, 2012.
- [244] Xin Wang, Fang Yuan, and Li Qi. Recommendation in Education Portal by Relation Based Importance Ranking. In *Advances in Web Based Learning - ICWL 2008*, pages 39–48. Springer, 2008.
- [245] Yuanyuan Wang and Kazutoshi Sumiya. Semantic Ranking of Lecture Slides based on Conceptual Relationship and Presentational Structure. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2801–2810. Procedia Computer Science, 2010.
- [246] Nicolas Weber, Karin Schoefegger, Jenny Bimrose, Tobias Ley, Stefanie N. Lindstaedt, Alan Brown, and Sally-Anne Barnes. Knowledge Maturing in the Semantic MediaWiki: A Design Study in Career Guidance. In *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 700–705. Springer, 2009.
- [247] Armin Weinberger, Bernhard Ertl, Frank Fischer, and Heinz Mandl. Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science*, 33(1):1–30, 2005.
- [248] Etienne Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1999.
- [249] Martin Wessner. *Kontextuelle Kooperation in virtuellen Lernumgebungen*. Eul Verlag, 2005.

- 
- [250] Jane Yau, Jeanne Lam, and KS Cheung. A Review of e-Learning Platforms in the Age of e-Learning 2.0. In *Hybrid Learning and Education*, pages 208–217. Springer, 2009.
  - [251] Vicente Arturo Romero Zaldivar and Daniel Burgos. Meta-Mender: A meta-rule based recommendation system for educational applications. In *Proceedings of the 1st Workshop on Recommender Systems for Technology Enhanced Learning (RecSysTEL 2010)*, volume 1, pages 2877–2882. Procedia Computer Science, 2010.
  - [252] Vicente Arturo Romero Zaldivar, Raquel M. Crespo García, Daniel Burgos, Carlos Delgado Kloos, and Abelardo Pardo. Automatic Discovery of Complementary Learning Resources. In *Towards Ubiquitous Learning*, volume 6964 of *Lecture Notes in Computer Science*, pages 327–340. Springer, 2011.
  - [253] Raphael Zender, Julian Dehne, Hendrik Geßner, and Ulrike Lucke. RouteMe - Routing in Ad-hoc-Netzen als pervasives Lernspiel. *i-com*, 12(1):45–52, 2013.
  - [254] Raphael Zender, Richard Metzler, and Ulrike Lucke. FreshUP - A pervasive educational game for freshmen. *Pervasive and Mobile Computing*, (0), 2013.
  - [255] Birgit Zimmermann, Christoph Rensing, and Ralf Steinmetz. Experiences in using patterns to support process experts in process description and wizard creation. In *Transactions on pattern languages of programming II*, pages 34–61. Springer, 2011.



---

## List of Acronyms

---

CROKODIL Communities, Web-Ressourcen und Kompetenzentwicklungsdienste integrierende Lernumgebung .....	13
RBL Resource Based Learning.....	1
TEL Technology Enhanced Learning.....	1





---

# Appendix

---



---

## A Details of the Survey on TEL Recommender Algorithms

---

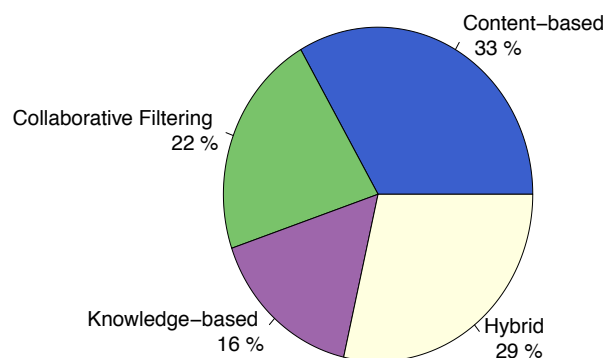
The details of the survey on TEL recommender algorithms presented in Chapter 3 are presented below. The publications selected for the survey are listed in Table A.1.

**Type of Recommender Algorithm:** The different types of recommender algorithms identified in the survey are shown in Figure A.1. Unlike in recommender systems for e-commerce, where collaborative filtering is the most common type [110], the survey shows that content-based and hybrid recommender systems seem to dominate in TEL. The type of recommender system is however not always easy to determine especially in the early years when a common classification of recommender systems was not yet widely accepted. Table A.2 gives an overview of the type of recommender algorithm over the years. Content-based and hybrid algorithms were dominant especially in the early years. Collaborative filtering and knowledge-based algorithms could only be identified as from 2007. Two publications could not be classified, one was a survey paper [239] and the other a framework [171].

- **Content-based:** Content-based recommender algorithms identified were mostly based on fundamental approaches from information retrieval, machine learning, natural language processing, data and web mining techniques. Some methods noted were for example Vector Space Model [86], Support Vector Machines [56], clustering [57, 128] and semantic relatedness [222, 218, 229].
- **Collaborative filtering:** These approaches included user-based, item-based, multi-attribute collaborative filtering [156, 158] as well as graph-based techniques for example based on a random walk or Markov Chain Model [18, 19, 97, 107, 242]. Tag-based [54] and trust-based recommender systems [80, 137] were also identified.
- **Knowledge-based:** These approaches used domain knowledge and ontology-based modeling of user profiles and document models [44, 209].
- **Hybrids:** These included combinations of content-based and collaborative filtering [58, 121] as well as content-based and knowledge-based [114, 231], and collaborative filtering and knowledge-based [71, 223].

**Type of Recommended Items:** The recommender systems identified in the survey made very diverse recommendations ranging from the common learning resources and fellow peers recommendations to more TEL specific recommendations such as learning sequences, advice to teachers or grade predictions. The findings in Figure A.2 and Table A.3 are not surprising and confirm those reported in related work [45]. The most common recommendations are resources. Some recommender systems recommended several types of recommendations, for example [97, 138, 146, 187, 251].

- **Resources:** This category was very wide comprising for example of learning resources [138, 146], learning material [107], learning content [128], learning objects [38, 180], learning items [53],



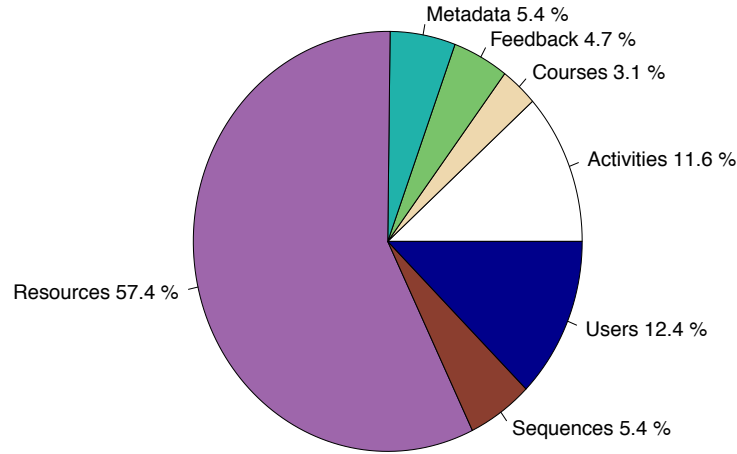
**Figure A.1:** Type of Recommender Algorithms

Publication Venue	Years Reviewed	Selected Publications (107)
Computers & Education	2008 - 2013	[105, 106, 127, 152, 224, 240, 243]
Journal of Computer Assisted Learning (JCAL)	2010	[156]
IEEE Transactions on Learning Technologies (TLT)	2008 - 2013	[4, 53, 182, 210, 216, 239]
IEEE International Conference on Advanced Learning Technologies (ICALT)	2007 - 2013	[22, 24, 44, 51, 54, 56, 58, 57, 86, 104, 107, 119, 121, 140, 128, 178, 181, 184, 187, 189, 205, 209, 215, 235, 236, 242]
European Conference on Technology Enhanced Learning (EC-TEL)	2006 - 2013	[9, 10, 19, 20, 28, 37, 38, 39, 52, 55, 63, 73, 84, 91, 95, 109, 113, 114, 123, 125, 135, 138, 145, 146, 147, 163, 171, 180, 183, 193, 206, 208, 212, 213, 217, 218, 222, 223, 237, 246, 252]
ACM Conference on Recommender Systems (RecSys)	2008 - 2012	[71, 97, 211, 228, 232]
Workshop on Recommender Systems for Technology Enhanced Learning (RecSys-TEL)	2010 - 2012	[26, 42, 162, 173, 177, 214, 220, 229, 230, 231, 233, 245, 251, 18, 80, 88, 137, 158]
Workshop on Social Information Retrieval for Technology-Enhanced Learning Exchange (SIRTEL)	2007 - 2008	[69, 70, 155]

**Table A.1:** Overview of Selected Publications for Survey

Year	Content-based	Collaborative Filtering	Knowledge-based	Hybrids
2013	[106, 119, 20, 180]	[107, 53]	[51, 145]	[135, 210, 193]
2012	[128, 140, 56]	[18, 80, 19] [228, 224, 158, 137, 243]	[88]	[54, 240, 232]
2011	[138, 252, 217, 22, 205]	[184]	[189, 63]	[84, 127]
2010	[42, 123, 105, 173, 222, 229]	[156, 230, 233, 220, 4, 216]	[162, 251, 214, 245, 181]	[231, 44, 26, 38, 177]
2009	[86, 146, 246, 208, 209]	[73, 97]	[104, 95, 213, 52]	[215, 114, 223]
2008	[182]	[242, 70]	[9]	[152, 211, 121, 58, 113, 237, 70, 55]
2007	[10, 24, 187, 206, 57, 236]	[109, 155]	[178, 236]	[147, 69]
2006	[183, 163, 217, 37, 212]			[125, 39, 28, 91]

**Table A.2:** Overview of the Type of Recommender Algorithm over Time



**Figure A.2:** Type of Recommended Items

Type of Recommended Items	Publications
Resources	[135, 106, 210, 119, 107, 51, 53, 180, 145, 18, 54, 19, 232, 128, 224, 158, 137, 56, 243, 88, 189, 138, 84, 252, 218, 205, 184, 42, 156, 123, 105, 230, 222, 229, 44, 216, 4, 233, 38, 245, 181, 86, 104, 146, 246, 114, 208, 73, 223, 97, 208, 152, 9, 113, 55, 121, 58, 242, 211, 70, 182, 206, 178, 235, 57, 236, 155, 183, 125, 217, 37, 39, 28, 91]
Users	[53, 80, 22, 146, 107, 97, 9, 26, 18, 231, 52, 114, 113, 147, 217, 37]
Metadata	[63, 177, 10, 193, 220, 246, 24]
Activities	[140, 127, 71, 138, 146, 187, 173, 97, 18, 214, 95, 213, 215, 69, 163]
Sequences	[240, 251, 236, 20, 135, 193, 206]
Courses	[187, 212, 251, 236]
Feedback	[228, 109, 123, 214, 237, 162]

**Table A.3:** Overview of the Type of Recommended Items

reading material [106], educational resources [121], assets [97], documents and pictures [232], collaboration media [9], movies [119], TV Programmes [28], web pages [224], posts or threads [4], messages [55] and job profiles [223].

- Users: This included fellow learning peers or learners [53], teachers [80], teams of teachers [22], experts [146], tutors [107], researchers [97], collaboration partners [9] and generally other people [26].
- Metadata: Examples were tags [10, 63, 177], concepts [193, 220] and categories [246].
- Activities: Examples of activities found in the survey were learning activities [71, 140, 127], learning tasks [138], learning goals [146, 187], assignments, exercises and test questions [173], and group activities [97].
- Sequences: This included learning sequences [240], learning paths [251] and the order of learning through a course [236].
- Courses: The most common courses were language courses [187, 212, 251].
- Feedback: This was the most diverse category ranging from user feedback, and user actions to more guidance-like feedback recommendations such as prompts, advice to teachers or learners, grade predictions [228] or possible intervention strategies to increase motivation [109].

An overview of the distribution of offline experiments across historical datasets and synthetic datasets over the years is given in Table A.4. An overview of the number of test weeks and number of participants in real life testing is shown in Table A.5. An overview of the different methods used in the user studies are shown in Table A.6. An overview of the focus of the evaluations are shown in Table A.7. An overview of the effects measured by evaluations in the survey are shown in Table A.8.

Years	Historical Datasets	Synthetic Datasets
2013	[180, 210, 53, 193]	
2012	[80, 19, 158, 240, 137, 56, 232]	
2011	[138, 218, 184, 63]	
2010	[230, 222, 216, 4, 233, 245, 177, 156]	[231]
2009	[146]	
2008	[58, 152, 182]	[70]
2007	[109, 206]	[155]

**Table A.4:** Offline Experiments

Years	Publications	Number of test weeks	Number of participants
2013	[145]	8	6
2012	[243]	8	144
	[88]	20	51
	[232]	4	199
2011	[252]	8	220
	[127]	16	440
2010	[26]	12	Not stated
	[229]	32	1763
	[156]	8	770
	[220]	10	25
2009	[246]	Not stated	Not stated
2006	[125]	Not stated	Not stated

**Table A.5:** Real Life Testing

Method	Publications
Questionnaire	[106, 119, 193, 53, 54, 232, 224, 252, 22, 63, 105, 214, 104, 95, 73, 223, 97, 10, 57]
Expert's Opinion	[232, 252, 22, 105, 10]
Interviews	[95, 114, 55]
Observations	[214, 177, 146, 213, 55]
Pre/ Post Test	[106, 193, 173, 104, 95, 237, 212]
Lab Experiment	[224, 214, 177, 146, 213, 55, 212]
Online Experiment	[119, 224, 73]
Crowdsourcing	[119]

**Table A.6:** Overview of Methods used in User Studies

Years	Recommender Algorithm	Recommender System	Both
2013	[210, 119, 53, 180]	[106, 193]	[145]
2012	[54, 240, 19, 158, 137, 56]	[224, 243, 88]	[232, 80]
2011	[218, 63, 22, 184, 138]		[252, 127]
2010	[231, 156, 230, 222, 216, 4, 233, 245]	[173, 229, 214, 177, 220]	[105, 26]
2009	[146, 97]	[104, 95, 213, 246, 114, 73, 223]	
2008	[152, 58, 182]	[237, 55]	[70]
2007	[10, 206, 57, 155]		[109]
2006		[125, 212]	

**Table A.7:** Focus of Evaluation

Effects Measured	Publications
Accuracy	[106, 210, 119, 53, 180, 54, 19, 232, 158, 137, 56, 138, 252, 218, 127, 22, 63, 156, 105, 173, 230, 222, 216, 4, 233, 26, 245, 104, 146, 246, 114, 97, 182, 55, 10, 206, 155, 57, 109]
Correlations	[193, 180, 184, 231, 229, 216, 177, 220, 146, 223]
Knowledge Levels	[145, 54, 127, 26, 220, 146]
Learning Motivation	[243, 105, 104]
Learning Performance	[193, 106, 243, 224, 233, 173, 233, 114, 237, 212, 127, 146]
Task Support	[53, 240, 88, 95, 146, 70, 127, 156, 152, 58, 155]
User Feedback	[106, 53, 145, 80, 232, 240, 137, 88, 127, 63, 105, 214, 4, 104, 95, 213, 114, 73, 223, 237, 55, 125, 212]

**Table A.8:** Effects Measured by Evaluation





---

## B Details of the Evaluation of the Hybrid Algorithms

---

In the following sections of this appendix, details of the evaluation of the hybrid graph-based algorithms presented in Chapter 4 and evaluated in Chapter 5 are given.

---

### B.1 Fundamentals of Evaluation Metrics and Statistical Significance Testing

---

---

#### B.1.1 Accuracy Prediction Metrics

---

Basic evaluation metrics needed to understand the proposed evaluation metrics in Chapter 5 are presented below. Table B.1 shows the classification of relevant items and non-relevant items recommended and not recommended [153].  $q$  is a query entity or a so called information need such as a search query or a tag or a user for whom recommendations should be generated. There can be many query entities  $q \in Q$ . True positives (tp) are items that are relevant to the information need or query entity ( $q$ ) and were recommended or can be found in the ranked list of recommendations. False positives (fp) are items that were recommended but which are not relevant to the information need. False negatives (fn) are items that have been rated as not relevant and were as such not recommended, these items were however relevant to the information need and should have been recommended. True negatives (tn) are items that were not recommended and are not relevant to the information need. This classification will be used to explain the following accuracy prediction metrics: precision, recall and f-measure.

**Precision (P)** is the number of relevant items retrieved (true positives) over all retrieved items (true and false positives) as shown in Equation B.1 [153].

$$P = \frac{|\text{relevant items retrieved}|}{|\text{retrieved items}|} = \frac{tp}{tp + fp} \quad (\text{B.1})$$

**Recall (R)** is the number of relevant items retrieved (true positives) over all relevant items (true positives and false negatives) as shown in Equation B.2 [153].

$$R = \frac{|\text{relevant items retrieved}|}{|\text{relevant items}|} = \frac{tp}{tp + fn} \quad (\text{B.2})$$

**F-Measure (F)** is also known as the  $F_1$  measure and is the harmonic mean which combines precision and recall into a single metric as shown in Equation B.3 [153].

$$F = \frac{2PR}{P + R} \quad (\text{B.3})$$

---

#### B.1.2 Statistical Significance Tests

---

Statistical significance tests required in Chapter 4 and in Chapter 5 are presented below.

	relevant items	non-relevant items
recommended items	true positives (tp)	false positives (fp)
not recommended items	false negatives (fn)	true negatives (tn)

**Table B.1:** Classification of True/False Positives and True/False Negatives, adapted from [153]

	$H_0$ is true	$H_0$ is false
<b>Null Hypothesis <math>H_0</math> is not rejected</b>	Alternative hypothesis $H_1$ is correctly rejected	Alternative hypothesis $H_1$ is wrongly rejected (Type II error)
<b>Reject Null Hypothesis <math>H_0</math></b>	Alternative hypothesis $H_1$ is wrongly supported (Type I error)	Alternative hypothesis $H_1$ is correctly supported

**Table B.2:** Type I and II Errors

## Null Hypothesis Significance Testing

The alternative hypothesis  $H_1$  states that the predicted effect occurred and was measured i.e. the manipulated independent variable had a direct effect on the measured dependent variable. The null hypothesis  $H_0$  states the contrary, that the predicted effect did not occur [83]. For example, if the alternative hypothesis  $H_1$  states that algorithm A performs better than algorithm B and the mean ratings measured for algorithm A are higher than the mean ratings for algorithm B. Then the null hypothesis  $H_0$  would state that there is no difference in the mean ratings for algorithm A and algorithm B, as such, algorithm A did not perform better than algorithm B. In other words, the data collected showing a difference in their mean ratings was collected by chance. The main aim in null hypothesis significance testing is to reject the null hypothesis. When the null hypothesis can be rejected with a certain probability, then the alternative hypothesis is supported with a certain confidence.

The  $p$ -value represents a chosen threshold value or significance level for significance testing. The null hypothesis  $H_0$  is rejected after testing for significance, if the  $p$ -value is below this threshold. The  $p$ -value is usually chosen at 0.05, which means that the probability of having collected data showing the effects stated in the null hypothesis is less than 5%. This threshold value is arbitrary and can be chosen at 0.01 or any other value [83]. In the example, the null hypothesis will be rejected if after testing for significance,  $p < 0.05$ , which means the probability or chance of having collected a particular set of data that shows that algorithm A does not perform any better than algorithm B is less than 5%. However, if  $p > 0.05$  then the null hypothesis cannot be rejected and the results of the experiment are considered not significant. This means the measured effect is obtained due to luck with a probability greater than 95%. Table B.2 shows in which cases a null hypothesis is rejected or not and when it is correctly or wrongly rejected. A Type I error occurs when a null hypothesis is wrongly rejected. This means the alternative hypothesis is wrongly supported and an effect is reported that does not exist. In other words, a false positive. Type II errors occur when there is a lot of sampling error. In this case, a null hypothesis that is false is not rejected. The alternative hypothesis is not supported and it is wrongly reported, that no effect exists. There are several ways to test for significance, depending on whether the data comes from two related samples or from two unrelated samples.

### Dependent Student's t-Test

Dependent student's t-tests evaluate the difference between two related samples. It is also known as the *paired samples student's t-test*. Commonly, the participants in one group are the same participants in the other group [83], like in a Within-test or in offline experiments. In the example above, this would mean that each user is given recommendations from algorithm A and from algorithm B. The results from algorithm A and algorithm B are then compared for each user. These results are also called paired results. However, to test recommendations in a user study, it is better to compare algorithms per user (each user is assigned to only one algorithm to test) as the same user receiving recommendations from

different algorithms might be biased, for example due to the order in which the recommendations are made [227]. Equation B.4 shows how the **t-value** is calculated:

$$t = \frac{(\text{Observed Mean} - \text{Expected Mean})}{\text{Standard Error}} = \frac{(M - 0)}{SE} = \frac{M}{SE} \quad (\text{B.4})$$

Where the observed mean is the mean of difference scores from the same user. The expected mean is the population mean of difference scores, which for null hypothesis testing is 0 as no difference is expected.  $M$  is the mean of the difference scores and  $SE$  is the standard error (see Appendix C.3 for details on standard error, standard deviation and the mean). **Degrees of freedom** will be  $df = n - 1$ , where  $n$  is the number of scores [83].

### Independent Student's t-Test

Independent student's t-tests evaluate the difference between two independent samples. It compares the means of two different groups of data to determine if the means are significantly different from one another. This is mostly used for a between-test or A/B test. Participants are randomly assigned to either algorithm A or algorithm B, therefore, unlike the paired results, the two algorithms are not tested on the same group of participants and the results are therefore unpaired or independent [83]. The two groups of participants are not related, but the scores are comparable. Equation B.5 shows how the **t-value** is calculated:

$$t = \frac{(\text{Observed Mean Difference} - \text{Expected Mean Difference})}{\text{Average Standard Error}} = \frac{(M_1 - M_2) - 0}{SE_{1,2}} = \frac{(M_1 - M_2)}{(SE_1 + SE_2)/2} \quad (\text{B.5})$$

Where the observed mean difference is the difference between the means from two different users. The expected mean is the difference between two population means, which for null hypothesis testing is 0 as no difference is expected.  $M$  is the mean and  $SE_1$  is the standard error for the first sample and  $SE_2$  is the standard error for the second sample. The average of the standard errors for both samples is taken as  $SE_{1,2} = (SE_1 + SE_2)/2$ . As the the standard errors are averaged together, it is necessary that the variances of both samples are similar. Therefore the variances of both samples need to be tested with an *F-Test*. Depending on the results of this test, a *two sample t-test* or a *Welch t-test* is performed. A two sample t-test is performed if the variances are the same and a Welch t-test if they are not. **Degrees of freedom** will be calculated as  $df = (n_1 - 1) + (n_2 - 1)$ , where  $n_1$  is the number of scores from the first group and  $n_2$  the number of scores from the second group [83].

---

### Wilcoxon Signed-Rank Tests

---

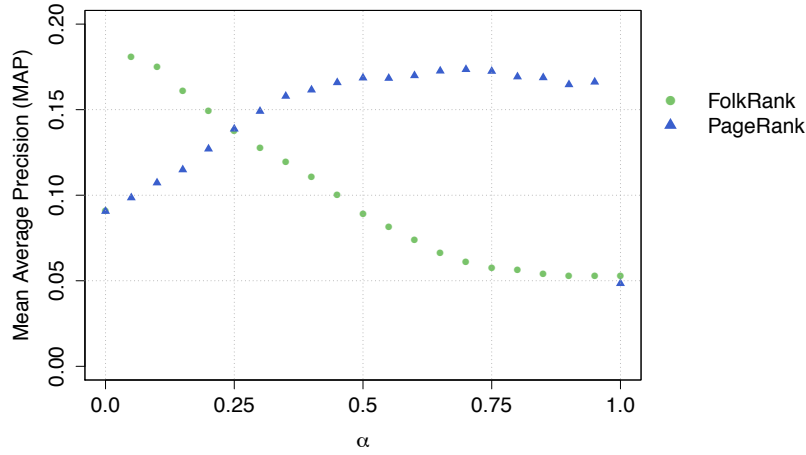
The Wilcoxon signed-rank test compares results pairwise. It compares two sets of scores from the same participants. The absolute differences between the paired scores are ranked. Only non-zero values are considered. The ranks of the positive scores (greater than zero) are summed up giving a  $V$  value. The Wilcoxon signed-rank test does not assume the data is normally distributed. It however requires that the results are comparable. The Wilcoxon signed-rank test can be said to be the nonparametric test equivalent to the dependent t-test [83].

---

### Wilcoxon Rank-Sum Tests

---

The Wilcoxon rank-sum test compares scores from two groups of different participants. This is also known as a Mann-Whitney's U test. It does not assume the data is normally distributed. It however requires that the results from different participants are comparable and still assumes the equality of variances. For the calculation, the median of the two groups, not the means are used, giving a  $W$  value. The Wilcoxon rank-sum test can be said to be the nonparametric equivalent to the independent t-test [83].



**Figure B.1:** Parameter Analysis of FolkRank's biased jump probability parameter

---

### Effect Size - Cohen's D

---

**Cohen's d** is the measure of effect size when comparing two means [83]. It is the difference of the two means divided by the pooled standard deviation as shown in Equation B.6.

$$d = \frac{(M_1 - M_2)}{SD_{1,2}} = \frac{(M_1 - M_2)}{(SD_1 + SD_2)/2} \quad (\text{B.6})$$

Where  $M_1$  is the mean of the first group,  $M_2$  the mean of the second group and  $SD_{1,2}$  is the pooled standard deviation of both groups.

Practically, Cohen's d gives an indication of the strength or impact of the significance test results obtained. Commonly  $d = 0.2$  is termed a small effect, around  $d = 0.5$  a moderate effect and  $d > 0.8$  a large effect.

---

## B.2 Evaluation Details

---



---

### B.2.1 Parametrization

---

The parameters used for the evaluation of the hybrid graph-based algorithms in Chapter 5 are determined in the following sections.

---

#### Parametrization of FolkRank

---

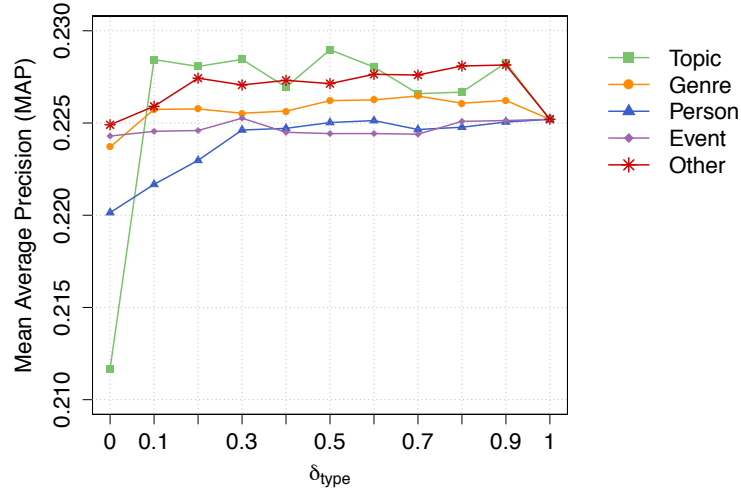
FolkRank is parametrized with the biased jump probability parameter  $\alpha$  of the biased surfer model. A sensitivity analysis is conducted with steps of 0.05.  $\alpha$  is varied in the interval  $\alpha \in [0.0, 1.0]$ . The results are shown in Figure B.1. The maximal MAP for FolkRank is achieved with  $\alpha = 0.05$ . The maximal MAP for PageRank with the biased surfer model is achieved with  $\alpha = 0.70$ . FolkRank achieves a higher MAP than PageRank, hence the subtraction of the scores obtained from PageRank is justified. On the basis of these results, the parameter  $\alpha = 0.05$  is set for FolkRank for the evaluations in this thesis.

---

#### Parametrization of AlnheritScore

---

AlnheritScore takes the values of GRank's parameters which according to a sensitivity analysis in [1] shall be set to:  $d_a = 10$  and  $d_b = d_c = 2$ .



**Figure B.2:** Parameter Analysis of  $\delta_{type}$

### Semantic Tag Type Parameter Analysis for AspectScore

LeavePostOut results from FolkRank with the user of a post as query entity are used to determine the parameter values  $\delta_{type}$  and  $\gamma_{type_q, type}$  for AspectScore. Each of the tag types are investigated in order to determine which tag type is the most effective as query entity [203]. The tag type *Location* is not considered as it is not represented in the dataset. The influence of the tag type in AspectScore is determined by  $\delta_{type}$  and  $\gamma_{type_q, type}$ . These parameters are analysed with regard to the tag types. For each tag type in the dataset, the parameter  $\delta_{type_A}$  is varied between 0.0 and 1.0. Parameter  $\gamma_{type_q, type}$  is set to 1 if the tag type of a tag is the current tag type being analysed and set to 0 if not. Setting  $\gamma_{type_q, type} = 1$  for certain tag types restricts the now *biased* surfer to paths having only these tag types. Tags of the same type as the query tag are given preference. Thus AspectScore can be said to be FolkRank with a tag type biased surfer on a disambiguated folksonomy graph based on tag types. The parameter analysis of  $\delta_{type}$  is shown in Figure B.2. For each of the tag types, there is an initial increase in MAP, meaning all tag types have a positive influence on the effectiveness of the recommendations, however the tag type *Topic* results in the highest MAPs. Therefore, tag type *Topic* is chosen to be the preferred tag type in AspectScore. For the evaluation, the parameters are set to  $\delta_{type} = 1$  and  $\gamma_{type_q, type} = 1$ .

LeavePostOut					
	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
Minimum	1	1	1	1	1
1st Quartile	6	5	10	5	18
Median	27	24	41	20	50
Mean	38	36	47	30	55
3rd Quartile	64	58	73	45	85
Maximum	138	135	138	133	138
LeaveRTOut					
	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
Minimum	1	1	1	1	1
1st Quartile	5	5	5	5	17
Median	11	11	11	12	38
Mean	23	22	25	23	51
3rd Quartile	28	28	28	33	81
Maximum	143	143	143	142	143

**Table B.3:** BibSonomy - Distribution of Recommendations for LeavePostOut and LeaveRTOut

## B.2.2 Detailed Evaluation Results on the BibSonomy Dataset

Table B.3 shows the descriptive statistics for the distribution of recommendation positions in a ranked list of recommendations for LeavePostOut and LeaveRTOut. The Mean Normalized Precision (MNP) evaluation results of the top ten positions in the ranked list of recommendations are given in Table B.4. Table B.5 shows the Mean Average Precision (MAP) results of LeavePostOut and LeaveRTOut. A summary of descriptive statistics for the average precision values is shown in Table B.6. Detailed results of the significance tests stating p-values are shown in Table B.7. The Cohen's d effect sizes are given in Table B.8.

## B.2.3 Detailed Evaluation Results on the GroupMe! Dataset

Table B.9 shows the descriptive statistics for the distribution of recommendation positions in a ranked list of recommendations for LeavePostOut and LeaveRTOut. The results of the Mean Normalized Precision (MNP) for the top ten positions in the ranked list of recommendations for LeavePostOut and LeaveRTOut are shown in Table B.10. Table B.11 shows the Mean Average Precision (MAP) results of LeavePostOut and LeaveRTOut. A summary of descriptive statistics for the average precision values is shown in Table B.12. The detailed results of the significance tests stating p-values are shown in Table B.13. The Cohen's d effect size are shown in Table B.14.

## B.2.4 Detailed Evaluation Results on the CROKODIL Dataset

The positions of the ranked recommendations obtained from LeavePostOut and LeaveRTOut are shown in Table B.15. The MNP evaluation results of the top ten positions in the ranked list of recommendations are shown in Table B.16 and in Table B.17 the MAP results of LeavePostOut and LeaveRTOut. A summary of descriptive statistics for the average precision values is shown in Table B.18. Detailed results of the significance tests stating p-values are shown in Table B.19 and the Cohen's d values in Table B.20.



LeavePostOut					
Position	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
1	0.096	0.111	0.083	0.106	0.042
2	0.159	0.166	0.130	0.170	0.062
3	0.193	0.199	0.153	0.203	0.084
4	0.222	0.229	0.172	0.231	0.105
5	0.248	0.265	0.186	0.261	0.117
6	0.271	0.281	0.208	0.290	0.129
7	0.294	0.291	0.225	0.308	0.144
8	0.303	0.305	0.236	0.332	0.154
9	0.316	0.320	0.245	0.349	0.171
10	0.325	0.341	0.256	0.365	0.180
LeaveRTOut					
Position	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
1	0.098	0.093	0.082	0.066	0.039
2	0.141	0.132	0.147	0.135	0.065
3	0.182	0.174	0.187	0.177	0.091
4	0.233	0.227	0.229	0.227	0.114
5	0.278	0.277	0.301	0.278	0.114
6	0.316	0.318	0.348	0.323	0.114
7	0.378	0.375	0.387	0.354	0.175
8	0.420	0.431	0.417	0.398	0.178
9	0.453	0.465	0.454	0.426	0.188
10	0.486	0.489	0.493	0.457	0.188

**Table B.4:** BibSonomy - Mean Normalized Precision (MNP) Results for LeavePostOut and LeaveRTOut

LeavePostOut				
FolkRank	AspectScore	InteliScore	VSScore	Most Popular
0.181	0.194	0.150	0.197	0.094
LeaveRTOut				
FolkRank	AspectScore	InteliScore	VSScore	Most Popular
0.203	0.199	0.198	0.182	0.099

**Table B.5:** BibSonomy - Mean Average Precision (MAP) Results for LeavePostOut and LeaveRTOut

LeavePostOut					
	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
Minimum	0.007	0.007	0.007	0.008	0.007
1st Quartile	0.016	0.017	0.014	0.022	0.012
Median	0.037	0.042	0.025	0.050	0.020
Mean	0.181	0.194	0.150	0.197	0.094
3rd Quartile	0.167	0.200	0.100	0.200	0.055
Maximum	1.000	1.000	1.000	1.000	1.000
Standard Deviation	0.297	0.311	0.281	0.305	0.210
LeaveRTOut					
	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
Minimum	0.007	0.007	0.007	0.007	0.007
1st Quartile	0.036	0.036	0.036	0.030	0.012
Median	0.091	0.091	0.091	0.083	0.026
Mean	0.203	0.199	0.198	0.182	0.099
3rd Quartile	0.200	0.200	0.200	0.200	0.059
Maximum	1.000	1.000	1.000	1.000	1.000
Standard Deviation	0.286	0.279	0.271	0.254	0.207

**Table B.6:** BibSonomy - Summary of Average Precision Results for LeavePostOut and LeaveRTOut

LeavePostOut					
	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
FolkRank		p=0.9377 W=489937	<b>p&lt;2.2e-16</b> V=259517	p=1 V=144924	<b>p&lt;2.2e-16</b> V=329467
AspectScore	p= 0.062 W=530164		<b>p=6.8e-11</b> W=594141	p=0.997 W=473668	<b>p=2.2e-16</b> W=657443
InteliScore	p=1 V=111575	p=1 W=425959		p=1 V=110747	<b>p=1.9e-06</b> V=291934
VSScore	<b>p=1.2e-10</b> V=240079	<b>p=0.003</b> W=546433	<b>p=2.2e-16</b> V=298313		<b>p=2.2e-16</b> V=356204
Most Popular	p=1 V=177054	p=1 W=362658	p=1 V=207566	p=1 V=148307	
LeaveRTOut					
	FolkRank	AspectScore	InteliScore	VSScore	Most Popular
FolkRank		p=0.641 W=5099037	0.334 V=942246	<b>p=5.7e-06</b> V=2118560	<b>p=2.2e-16</b> V=919450
AspectScore	p= 0.359 W=5152547		p=0.426 W=5139529	<b>p=0.004</b> W=5320745	<b>p=2.2e-16</b> W=7376339
InteliScore	p=0.667 V= 921169	p=0.574 W=5112055		<b>p=0.002</b> V=2049952	<b>p=2.2e-16</b> V=3893987
VSScore	p=1 V=1747030	p=0.996 W=4930839	p=0.998 V=1804524		<b>p=2.2e-16</b> V=3641433
Most Popular	p=1 V=1043375	p=1 W=2875245	p=1 V=1053098	p=1 V=1305652	

**Table B.7:** BibSonomy - Results of pairwise Wilcoxon Significance Tests for LeavePostOut and LeaveRTOut

LeavePostOut				
	FolkRank	AspectScore	InteliScore	VSScore
AspectScore	d=0.043			
InteliScore	d=0.108	d=0.149		
VSScore	d=0.054	d=0.010	d=0.162	
Most Popular	d=0.337	d=0.376	d=0.224	d=0.393
LeaveRTOut				
	FolkRank	AspectScore	InteliScore	VSScore
AspectScore	d=0.014			
InteliScore	d=0.017	d=0.003		
VSScore	d=0.076	d=0.061	d=0.060	
Most Popular	d=0.415	d=0.406	d=0.409	d=0.359

**Table B.8:** BibSonomy - Results of Cohen's d for LeavePostOut and LeaveRTOut

LeavePostOut						
	FolkRank	GFolkRank	AScore	GRank	AIInheritscore	Most Popular
Minimum	1	1	1	1	1	24
1st Quartile	6	1	1	1	1	1657
Median	81	1	1	14	2	1706
Mean	110	12	14	233	197	1610
3rd Quartile	113	2	3	194	162	1781
Maximum	1689	1726	1729	1785	1773	1789
LeaveRTOut						
	FolkRank	GFolkRank	AScore	GRank	AIInheritscore	Most Popular
Minimum	1	1	1	1	1	1
1st Quartile	6	5	5	8	13	223
Median	39	28	28	35	62	739
Mean	85	66	66	126	166	836
3rd Quartile	106	81	80	103	199	1515
Maximum	1615	577	632	1777	1789	1789

**Table B.9:** GroupMe! - Distribution of Recommendations for LeavePostOut and LeaveRTOut

LeavePostOut						
Position	FolkRank	GFolkRank	AScore	GRank	AIInheritscore	Most Popular
1	0.140	0.538	0.600	0.320	0.385	0.000
2	0.194	0.753	0.736	0.383	0.505	0.000
3	0.197	0.873	0.759	0.401	0.555	0.000
4	0.198	0.902	0.821	0.407	0.561	0.000
5	0.200	0.911	0.832	0.412	0.563	0.000
6	0.256	0.913	0.851	0.425	0.568	0.000
7	0.256	0.916	0.870	0.440	0.573	0.000
8	0.260	0.916	0.877	0.456	0.575	0.000
9	0.262	0.916	0.889	0.470	0.581	0.000
10	0.264	0.918	0.894	0.474	0.588	0.000
LeaveRTOut						
Position	FolkRank	GFolkRank	AScore	GRank	AIInheritscore	Most Popular
1	0.111	0.115	0.116	0.079	0.053	0.013
2	0.150	0.164	0.164	0.108	0.077	0.018
3	0.184	0.194	0.194	0.131	0.094	0.018
4	0.227	0.245	0.245	0.157	0.115	0.026
5	0.248	0.280	0.281	0.196	0.143	0.026
6	0.263	0.301	0.299	0.221	0.166	0.030
7	0.274	0.319	0.316	0.236	0.184	0.030
8	0.288	0.333	0.329	0.261	0.204	0.033
9	0.293	0.344	0.348	0.271	0.214	0.033
10	0.297	0.354	0.359	0.289	0.223	0.046

**Table B.10:** GroupMe! - Mean Normalized Precision (MNP) Results for LeavePostOut and LeaveRTOut

LeavePostOut					
FolkRank	GFolkRank	AScore	GRank	AIInheritscore	Most Popular
0.188	0.696	0.705	0.378	0.473	0.001
LeaveRTOut					
FolkRank	GFolkRank	AScore	GRank	AIInheritscore	Most Popular
0.180	0.196	0.195	0.144	0.107	0.024

**Table B.11:** GroupMe! - Mean Average Precision (MAP) Results for LeavePostOut and LeaveRTOut

LeavePostOut						
	FolkRank	GFolkRank	AScore	GRank	AINheritscore	Most Popular
Minimum	0.00059	0.00058	0.00058	0.00056	0.00056	0.00056
1st Quartile	0.00885	0.50000	0.33333	0.00515	0.00617	0.00056
Median	0.01235	1.00000	1.00000	0.07143	0.50000	0.00059
Mean	0.18776	0.69623	0.70460	0.37759	0.47282	0.00081
3rd Quartile	0.16667	1.00000	1.00000	1.00000	1.00000	0.00060
Maximum	1.00000	1.00000	1.00000	1.00000	1.00000	0.04167
Standard Deviation	0.34730	0.34959	0.38195	0.44393	0.44492	0.00154
LeaveRTOut						
	FolkRank	GFolkRank	AScore	GRank	AINheritscore	Most Popular
Minimum	0.00062	0.00170	0.00160	0.00056	0.00056	0.00056
1st Quartile	0.00941	0.01230	0.01250	0.00969	0.00503	0.00066
Median	0.02564	0.03570	0.03570	0.02857	0.01626	0.00014
Mean	0.17998	0.19520	0.19570	0.14351	0.10745	0.00245
3rd Quartile	0.16667	0.20000	0.20000	0.12500	0.07692	0.00448
Maximum	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
Standard Deviation	0.31244	0.31520	0.31588	0.27127	0.23181	0.12092

**Table B.12:** GroupMe! - Summary of Average Precision Results for LeavePostOut and LeaveRTOut

LeavePostOut						
	FolkRank	GFolkRank	AScore	GRank	AINheritscore	Most Popular
FolkRank		p=1 V=15135	p=1 V=12971	p=1 V=389864	p=1 V=318824	p=2.2e-16 V=1731033
GFolkRank	p=2.2e-16 V=1278502		p=0.044 V=139274	p=2.2e-16 V=789388	p=2.2e-16 V=543258	p=2.2e-16 V=1739912
AScore	p=2.2e-16 V=1277450	p=0.9564 V=120287		p=2.2e-16 V=801551	p=2.2e-16 V=636697	p=2.2e-16 V=1739912
GRank	p=2.2e-16 V=1142262	p=1 V=99724	p=1 V=83565		p=1 V=267045	p=2.2e-16 V=1732969
AINheritScore	p=2.2e-16 V=1173304	p=1 V=142949	p=1 V=167849	p=2.2e-16 V=572116		p=2.2e-16 V=1734625
Most Popular	p=1 V=9012	p=1 V=133	p=1 V=133	p=1 V=7076	p=1 V=5420	
LeaveRTOut						
	FolkRank	GFolkRank	AScore	GRank	AINheritscore	Most Popular
FolkRank		p=1 V=1722996	p=1 V=1831062	p=4.9e-09 V=4133364	p=2.2e-16 V=4876434	p=2.2e-16 V=9106561
GFolkRank	p=2.2e-16 V=2975650		p=1 V=756348	p=2.2e-16 V=4641338	p=2.2e-16 V=5245258	p=2.2e-16 V=9130881
AScore	p=2.2e-16 V=3116022	p=1.4e-15 V=1149780		p=2.2e-16 V=4605533	p=2.2e-16 V=636697	p=2.2e-16 V=5232572
GRank	p=1 V=3337680	p=1 V=2718028	p=1 V=2707867		p=2.2e-16 V=3456741	p=2.2e-16 V=8919433
AINheritScore	p=1 V=3133568	p=1 V=2585645	p=1 V=2566703	p=1 V=2396590		p=2.2e-16 V=8522768
Most Popular	p=1 V=33389	p=1 V=9069	p=1 V=8452	p=1 V=220517	p=1 V=612907	

**Table B.13:** GroupMe! - Results of pairwise Wilcoxon Signed-Rank Significance Tests for LeavePostOut and LeaveRTOut

LeavePostOut					
	FolkRank	GFolkRank	AScore	GRank	AINheritScore
GFolkRank	d=1.459				
AScore	d=1.416	d=0.023			
GRank	d=0.476	d=0.797	d=0.790		
AINheritScore	d=0.714	d=0.558	d=0.559	d=0.214	
Most Popular	d=0.761	d=2.813	d=2.606	d=1.200	d=1.500
LeaveRTOut					
	FolkRank	GFolkRank	AScore	GRank	AINheritScore
GFolkRank	d=0.040				
AScore	d=0.040	d=0.001			
GRank	d=0.125	d=0.176	d=0.177		
AINheritScore	d=0.264	d=0.317	d=0.318	d=0.143	
Most Popular	d=0.656	d=0.715	d=0.716	d=0.567	d=0.449

**Table B.14:** GroupMe! - Results of Cohen's d for LeavePostOut and LeaveRTOut

LeavePostOut							
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
Minimum	1	1	1	1	1	1	7
1st Quartile	60	4	3	3	9	9	407
Median	91	22	6	4	73	34	431
Mean	94	32	22	19	134	120	411
3rd Quartile	116	41	30	23	216	184	440
Maximum	490	387	447	267	542	548	452
LeaveRTOut							
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
Minimum	1	1	1	1	1	1	1
1st Quartile	5	5	5	5	6	8	86
Median	13	13	12	12	23	30	237
Mean	26	21	22	21	45	72	229
3rd Quartile	35	29	27	28	48	92	351
Maximum	496	393	487	275	553	555	554

**Table B.15:** CROKODIL - Distribution of Recommendations for LeavePostOut and LeaveRTOut

LeavePostOut							
Position	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
1	0.002	0.080	0.160	0.145	0.108	0.100	0.000
2	0.004	0.093	0.214	0.214	0.165	0.184	0.000
3	0.004	0.104	0.327	0.400	0.167	0.188	0.000
4	0.006	0.253	0.437	0.524	0.186	0.190	0.000
5	0.006	0.262	0.455	0.550	0.197	0.212	0.000
6	0.009	0.279	0.517	0.600	0.210	0.221	0.000
7	0.013	0.355	0.548	0.617	0.216	0.238	0.004
8	0.013	0.355	0.548	0.617	0.232	0.249	0.004
9	0.015	0.355	0.552	0.617	0.262	0.268	0.004
10	0.015	0.357	0.563	0.619	0.268	0.273	0.004
LeaveRTOut							
Position	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
1	0.033	0.037	0.034	0.027	0.048	0.037	0.015
2	0.090	0.096	0.082	0.074	0.093	0.095	0.030
3	0.147	0.151	0.118	0.105	0.137	0.141	0.030
4	0.213	0.217	0.175	0.169	0.186	0.172	0.055
5	0.256	0.263	0.250	0.253	0.224	0.211	0.055
6	0.284	0.285	0.307	0.318	0.258	0.229	0.066
7	0.350	0.353	0.360	0.365	0.281	0.248	0.087
8	0.389	0.390	0.395	0.402	0.315	0.263	0.095
9	0.421	0.419	0.430	0.434	0.336	0.280	0.095
10	0.453	0.454	0.467	0.467	0.376	0.313	0.095

**Table B.16:** CROKODIL - Mean Normalized Precision (MNP) Results for LeavePostOut and LeaveRTOut

LeavePostOut						
FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
0.020	0.166	0.291	0.304	0.165	0.173	0.003
LeaveRTOut						
FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
0.153	0.158	0.151	0.145	0.143	0.127	0.042

**Table B.17:** CROKODIL - Mean Average Precision (MAP) Results for LeavePostOut and LeaveRTOut

LeavePostOut							
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
Minimum	0.0020	0.0026	0.0022	0.0037	0.0018	0.0018	0.0022
1st Quartile	0.0086	0.0244	0.0333	0.0445	0.0046	0.0054	0.0023
Median	0.0110	0.0455	0.1667	0.2500	0.0137	0.0294	0.0023
Mean	0.0198	0.1659	0.2909	0.3044	0.1645	0.1733	0.0033
3rd Quartile	0.0167	0.2500	0.3333	0.3333	0.1111	0.1111	0.0025
Maximum	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.1429
Standard Deviation	0.0548	0.2680	0.3386	0.3217	0.3141	0.3066	0.0096
LeaveRTOut							
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritscore	Most Popular
Minimum	0.0020	0.0025	0.0021	0.0036	0.0018	0.0018	0.0018
1st Quartile	0.0029	0.0345	0.0370	0.0357	0.0208	0.0109	0.0028
Median	0.0770	0.0769	0.0833	0.0833	0.0435	0.0333	0.0042
Mean	0.1530	0.1576	0.1507	0.1451	0.1434	0.1270	0.0420
3rd Quartile	0.2000	0.2000	0.1833	0.2000	0.1667	0.1250	0.0117
Maximum	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Standard Deviation	0.2019	0.2083	0.1988	0.1850	0.2278	0.2154	0.1405

**Table B.18:** CROKODIL - Summary of Average Precision Results for LeavePostOut and LeaveRTOut

LeavePostOut							
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AInheritscore	Most Popular
FolkRank		p=1 W=25708	p=1 V=3149	p=1 V=2170	p=1 V=30789	p=1 V=20624	<b>p=2.2e-16</b> V=105450
AspectScore	<b>p=2.2e-16</b> W=188662		p=1 W=76143	p=1 W=66690	<b>p=2.2e-16</b> W=145065	<b>p=4.5e-09</b> W=130558	<b>p=2.2e-16</b> W=212904
GFolkRank	<b>p=2.2e-16</b> V=104267	<b>p=1.0e-14</b> W=138226		p=1 V=6009	<b>p=2.2e-16</b> V=76230	<b>p=2.2e-16</b> V=74713	<b>p=2.2e-16</b> V=106947
AScore	<b>p=2.2e-16</b> V=103860	<b>p=2.2e-4</b> W=147680	<b>p=3.4e-15</b> V=22671		<b>p=2.2e-16</b> V=76577	<b>p=2.2e-16</b> V=75050	<b>p=2.2e-16</b> V=106951
GRank	<b>p=2.1e-16</b> V=75702	p=1 W=69305	p=1 V=18600	p=1 V=16519		p=1 V=28065	<b>p=2.2e-16</b> V=104105
AInheritScore	<b>p=2.2e-16</b> V=85867	p=1 W=83812	p=1 V=17952	p=1 V=17186	<b>p=1.6e-06</b> V=48963		<b>p=2.2e-16</b> V=103890
Most Popular	p=1 V=1966	p=1 W=1466	p=1 V=6	p=1 V=2	p=1 V=3312	p=1 V=3526	
LeaveRTOut							
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AInheritscore	Most Popular
FolkRank		p=0.854 W=838962	p=0.365 V=277685	p=0.488 V=264742	<b>p=6.6e-09</b> V=446549	<b>p=1.4e-16</b> V=487357	<b>p=2.2e-16</b> V=754624
AspectScore	p=0.146 W=879760		p=0.543 W=857294	p=0.565 W=856176	<b>p=5.8e-16</b> W=1014497	<b>p=2.2e-16</b> W=1115516	<b>p=2.2e-16</b> W=1541178
GFolkRank	p=0.6347 V=270944	p=0.458 W=861428		p=0.9993 V=32174	<b>p=5.6e-09</b> V=452513	<b>p=2.2e-16</b> V=512812	<b>p=2.2e-16</b> V=766144
AScore	p=0.5121 V=264165	p=0.435 W=862546	<b>p=6.805e-4</b> V=46829		<b>p=4.9e-08</b> V=445660	<b>p=2.2e-16</b> V=500102	<b>p=2.2e-16</b> V=763778
GRank	p=1 V=305603	p=1 W=704224	p=1 V=309482	p=1 V=312636		<b>p=2.2e-16</b> V=386662	<b>p=2.2e-16</b> V=739273
AInheritScore	p=1 V=293269	p=1 W=603205	p=1 V=282879	p=1 V=284277	p=1 V=202494		<b>p=2.2e-16</b> V=721574
Most Popular	p=1 V=31007	p=1 W=177544	p=1 V=74312	p=1 V=72793	p=1 V=76730	p=1 V=121478	

**Table B.19:** CROKODIL - Results of pairwise Wilcoxon Signed-Rank Significance Tests for LeavePostOut and LeaveRTOut



LeavePostOut						
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritScore
AspectScore	d=0.755					
GFolkRank	d=1.118	d=0.409				
AScore	d=1.233	d=0.468	d=0.041			
GRank	d=0.642	d=0.005	d=0.387	d=0.440		
AINheritScore	d=0.697	d=0.026	d=0.364	d=0.417	d=0.028	
Most Popular	d=0.419	d=0.857	d=1.200	d=1.323	d=0.725	d=0.784
LeaveRTOut						
	FolkRank	AspectScore	GFolkRank	AScore	GRank	AINheritScore
AspectScore	d=0.024					
GFolkRank	d=0.010	d=0.034				
AScore	d=0.039	d=0.064	d=0.029			
GRank	d=0.043	d=0.065	d=0.034	d=0.008		
AINheritScore	d=0.123	d=0.144	d=0.114	d=0.090	d=0.074	
Most Popular	d=0.636	d=0.651	d=0.632	d=0.628	d=0.536	d=0.467

**Table B.20:** CROKODIL - Results of Cohen's d for LeavePostOut and LeaveRTOut

---

## C Details of Crowdsourcing Experiment and Statistical Analysis of Results

---

In the following sections of this appendix, details of the crowdsourcing experiment described in Section 6.2 are presented. In addition, further descriptive and inference statistics of the results are presented as well.

---

### C.1 Crowdsourcing Experiment - Questionnaire

---

The following figures show the parts of the questionnaire used for the crowdsourcing experiments described in Section 6.2: Figure C.1 shows the general questions asked regarding demographics: age, gender, country and education and some other technical questions. Figure C.2 aims to motivate the participant to take part in a personal research on the topic of *Climate Change*. Five URLs of web pages found by the participant are expected. This question is also used to detect spammers. Figure C.3 has a question to determine the current state of the participant on the given topic before the recommendations are made. Figure C.4 shows an example of a recommendation made and a preview of the web page provided for the participant to be able to judge the content. Figure C.5 shows the questions posed to the participant regarding the recommended web resource shown in Figure C.4.

Recommendations of resources were generated for each of the activities presented in the experiments. A test user was selected for each experiment as well as two activities. The recommendations to these two activities for the selected user were then placed in the questionnaire.

---

#### C.1.1 Experiment Spring: Recommendations made to the Activities

---


For Experiment Spring, the test user called *FTestU1* was selected and the recommendations, with URLs (retrieved 10.11.2013), generated for this user by algorithms AScore and FolkRank to the selected activities are listed below. The two activities selected for Experiment Spring are boldly highlighted in Figure 6.2: *Understanding Climate Change* as super-activity and *Analyze the potential Catastrophes due to Climate Change* as sub-activity.

#### AScore Recommendations for the Activity: Understanding Climate Change

1. Global Warming Timeline  
<http://www.aip.org/history/climate/timeline.htm>
2. Outlook for the Future  
[http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12\\_1.shtml](http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12_1.shtml)
3. The Discovery of Global Warming - A History  
<http://www.aip.org/history/climate/timeline.htm>
4. MCII | Munich Climate-Insurance Initiative  
[http://www.climate-insurance.org/front\\_content.php?idcat=885](http://www.climate-insurance.org/front_content.php?idcat=885)
5. fig2.jpg (JPEG-Grafik, 841x648 Pixel)  
<http://www.jri.org.uk/resource/images/fig2.jpg>

#### FolkRank Recommendations for the Activity: Understanding Climate Change

1. Outlook for the Future  
[http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12\\_1.shtml](http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12_1.shtml)
2. The Discovery of Global Warming - A History  
<http://www.aip.org/history/climate/index.htm>
3. Climate Change: Evidence  
<http://climate.nasa.gov/evidence>



17% completed

---

**(A) General Questions:**

Gender:

Your Age:

Country:

1. What is your highest degree of education?  
(if not listed choose comparable one or "Other").

2. Which online resources, computer programs and platforms do you use (or have you used in the past) for Internet research ?

What you use:


3. From which online working platform did you reach our survey?

Platform:  ☐ none

Next

Multimedia Communications Lab (KOM) , TU Darmstadt, Germany

Figure C.1: Survey: Page 1



33% completed

---

**(B) Personal Research**

As the next step please make a quick Internet research (not more than **five minutes**) about the topic "Climate Change".

Carefully study at least **5** different web pages to get some knowledge about the topic.  
Prove your research by providing the links to these five sites below.  
Also keep these pages in mind/ open afterwards, as they will be of importance for the final step in the survey.

[Click to start your research!](#)


4. Please provide your links here:

- 
- 
- 
- 
- 

Next

Multimedia Communications Lab (KOM) , TU Darmstadt, Germany

Figure C.2: Survey: Page 2



50% completed

---

**(C) Your current state of knowledge**

Now please classify your own knowledge about the topic "Climate Change". For example if you would want to explain the topic to someone else.

Judge your knowledge from 1 (no knowledge) to 7 (expert knowledge).

1  
(no  
knowledge)
2
3
4
5
6
7  
(expert  
knowledge)

Your knowledge about the topic "Climate Change":

☐ ☐ ☐ ☐ ☐ ☐ ☐


---

Multimedia Communications Lab (KOM) , TU Darmstadt, Germany

**Figure C.3:** Survey: Page 3

**1.) Recommended resource: [Outlook for the Future](#)**

Below is a preview of this recommended web page:  
(use the scrollbar on the right of the preview to view the whole page)



**Calspace Courses**

[Climate Change - Part One](#)

Climate Change 1 Syllabus

- 1.0 - Introduction
- 2.0 - The Earth's Natural Greenhouse Effect
- 3.0 - The Greenhouse Gases
- 4.0 - CO<sub>2</sub> Emissions
- 5.0 - The Earth's Carbon Reservoirs
- 6.0 - Carbon Cycling: Some Examples
- 7.0 - Climate and Weather
- 8.0 - Global Wind Systems
- 9.0 - Clouds, Storms and Climates
- 10.0 - Global Ocean Circulation
- 11.0 - El Niño and the Southern Oscillation
- 12.0 Outlook for the Future
  - 12.1 - Introduction to Climate Change
  - 12.2 - Advances in Computer Modeling
  - 12.3 - Physics versus Fudge Factors

[Climate Change - Part Two](#)

[Introduction to Astronomy](#)

[Life in the Universe](#)

Glossary: Climate Change

Glossary: Astronomy

Glossary: Life in Universe

**Introduction to Climate Change**

**Introduction and Overview**

Now that we have learned about present climate change and mechanisms of the climate system, what we like to know is how the climate will change in the future and how this will affect living conditions where we have our homes (or our fields). We also want to know whether anything can or should be done about it, and if so, how we are going to go about doing it. Let's take these questions one at a time:

- How will climate change in the future?** To answer this we need (a) experience about how climate changes in response to outside forces, and (b) computer programs which simulate the climate system, so we can run experiments by changing the forces to see what happens to the artificial system.
- How will the overall change affect the conditions where I live?** This question is much more difficult to answer than the first. Again, there are two approaches. One is to chart past changes and see how the region of interest responded to the general change. The other is to make more complicated computer models, so that they can give more detailed information on regional change.
- Can anything be done about the climate change that is proceeding?** Well, inasmuch as the change is produced by human impact, reducing our impact presumably would reduce the rate of change. Also, we might think of ways to mitigate the impact by inventing ways to help neutralize it. However, it is highly unlikely we can reverse the change. The carbon cycle runs slowly, so it will take some hundreds to thousand of years to get it back into its original state. Also, attempts at mitigation might bring their own unforeseen problems.
- Assuming steps can be taken to minimize the impact of climate change, should we do anything about it?** It depends on how we judge, as a global community, the risks of doing nothing versus continuing on the present path of carbon fuel burning, and how we judge the economic costs compared to the risks.
- If something needs to be done, how is the world going to agree to do it?** First, of course, we would have to all agree that there is a problem and that the risks are unacceptable. Then we would have to agree on the costs associated with remedial action (such as cutting down on carbon emissions). Then we would have to agree on who is to bear what portions of those costs of action.

The main obstacle toward the task of doing anything regarding climate change is that the risks of doing nothing (that is, continuing on the present path), the benefits of doing something (that is, reducing carbon emissions), and the costs of the action (more expensive energy), are poorly defined and are by no means balanced in the same fashion between the various participants of the global community. The next most important obstacle is explosive population growth combined with the desire for a better standard of living. Yet

**5. Questions to the recommended resource above:**

**Figure C.4:** Survey: Page 4

**5. Questions to the recommended resource above:**

	0	1	2	3	4	5	more than 5
1. How many pictures and tables that are relevant to the given research topic does the given resource contain?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	1 (totally disagree)	2	3	4	5	6	7 (fully agree)
2. The given internet resource supports me very well in my research about the topic.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. If I could only use this resource, my research would still be very successful.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. Without this resource just by using my own resources, my research about the given topic would still be very good.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5. This internet resource gives me new insights and/ or information for my task.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6. I would have found this resource on my own/ anyway / during my research.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7. There are lots of important aspects about the topic described in this resource that lack in other resources.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8. This internet resource differs strongly from my other resources.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9. This resource informs me comprehensively about my topic.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10. This resource covers the whole spectrum of research about the given topic.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Figure C.5:** Survey: Page 5

- 
4. Climate Change Impacts & Threats | The Nature Conservancy  
<http://www.nature.org/ourinitiatives/urgentissues/global-warming-climate-change/threats-impacts/index.htm>
  5. Climate change brings natural disasters and disease - SciDev.Net  
<http://www.scidev.net/en/opinions/climate-change-brings-natural-disasters-and-diseas.html>

**AScore Recommendations for the Activity: Analyze the Potential Catastrophes due to Climate Change**

1. Future Climate Change | Climate Change | US EPA  
<http://www.epa.gov/climatechange/science/future.html>
2. Global Warming Timeline  
<http://www.aip.org/history/climate/timeline.htm>
3. NOAA Ocean Explorer: Education - Multimedia Discovery Missions | Lesson 7 - The Water Cycle | Activities: Global Warming and the Water Cycle  
[http://oceanexplorer.noaa.gov/edu/learning/7\\_water\\_cycle/activities/global\\_warming.html](http://oceanexplorer.noaa.gov/edu/learning/7_water_cycle/activities/global_warming.html)
4. Outlook for the Future  
[http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12\\_1.shtml](http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12_1.shtml)
5. What Problems Does Global Warming Cause | Care Badges  
<http://www.carebadges.com/?p=15>

**FolkRank Recommendations for the Activity: Analyze the Potential Catastrophes due to Climate Change**

1. Outlook for the Future  
[http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12\\_1.shtml](http://earthguide.ucsd.edu/virtualmuseum/climatechange1/12_1.shtml)
2. Climate Change: Evidence  
<http://climate.nasa.gov/evidence>
3. The Discovery of Global Warming - A History  
<http://www.aip.org/history/climate/index.htm>
4. Climate Change Impacts & Threats | The Nature Conservancy  
<http://www.nature.org/ourinitiatives/urgentissues/global-warming-climate-change/threats-impacts/index.htm>
5. Climate change brings natural disasters and disease - SciDev.Net  
<http://www.scidev.net/en/opinions/climate-change-brings-natural-disasters-and-diseas.html>

---

### C.1.2 Experiment Autumn: Recommendations made to the Activities

---

For Experiment Autumn, the test user *FTestU3* was chosen and the recommendations, with URLs (retrieved 10.11.2013), from algorithms AScore and FolkRank for this users to the individual activities selected are listed below. The two activities selected for Experiment Autumn are boldly highlighted as dotted lines in Figure 6.2: *Understanding Climate Change* as super-activity and *Analyze the potential Catastrophes due to Climate Change* as sub-activity.

**AScore Recommendations for the Activity: Investigate the causes for climate change**

1. Causes of climate change  
<http://edugreen.teri.res.in/explore/climate/causes.htm>
2. Changing Sun, Changing Climate  
<http://www.aip.org/history/climate/solar.htm>
3. Global Warming Timeline  
<http://www.aip.org/history/climate/timeline.htm>
4. Global Warming Frequently Asked Question  
<http://www.ncdc.noaa.gov/cmb-faq/globalwarming.html>
5. Understanding climate change - Think Change  
<http://www.climatechange.gov.au/climate-change/understanding-climate-change.aspx>

**FolkRank Recommendations for the Activity: Investigate the causes for climate change**

1. Global Warming Frequently Asked Questions  
<http://www.ncdc.noaa.gov/cmb-faq/globalwarming.html>
2. Understanding climate change - Think Change  
<http://www.climatechange.gov.au/climate-change/understanding-climate-change.aspx>
3. Understanding Climate Change | CSIRO  
<http://www.csiro.au/en/Outcomes/Climate/Understanding.aspx>
4. Roanoke College Professor Studies Effects of Global Warming - Roanoke College - Salem, Virginia  
<http://roanoke.edu/x19897.xml>
5. Global Warming Effects - Global Warming 2009 Data - The Daily Green  
<http://www.thedailygreen.com/environmental-news/latest/global-warming-heat-waves-47082601>

#### **AScore Recommendations for the Activity: Give an overview on the history of global warming**

1. Global Warming Timeline  
<http://www.aip.org/history/climate/timeline.htm>
2. HowStuffWorks "Effects of Global Warming: Sea Level Changes"  
<http://science.howstuffworks.com/environmental/green-science/global-warming4.htm>
3. History of climate change science - Wikipedia, the free encyclopedia  
[http://en.wikipedia.org/wiki/History\\_of\\_climate\\_change\\_science](http://en.wikipedia.org/wiki/History_of_climate_change_science)
4. temperature history.jpg (JPEG-Grafik, 600x432 Pixel)  
<http://www.globalwarminglies.com/pics/temperaturehistory.jpg>
5. Global Warming Frequently Asked Questions  
<http://www.ncdc.noaa.gov/cmb-faq/globalwarming.html>

#### **FolkRank Recommendations for the Activity: Give an overview on the history of global warming**

1. Global Warming Frequently Asked Questions  
<http://www.ncdc.noaa.gov/cmb-faq/globalwarming.html>
2. Understanding climate change - Think Change  
<http://www.climatechange.gov.au/climate-change/understanding-climate-change.aspx>
3. HowStuffWorks "Effects of Global Warming: Sea Level Changes"  
<http://science.howstuffworks.com/environmental/green-science/global-warming4.htm>
4. Understanding Climate Change | CSIRO  
<http://www.csiro.au/en/Outcomes/Climate/Understanding.aspx>
5. Roanoke College Professor Studies Effects of Global Warming - Roanoke College - Salem, Virginia  
<http://roanoke.edu/x19897.xml>

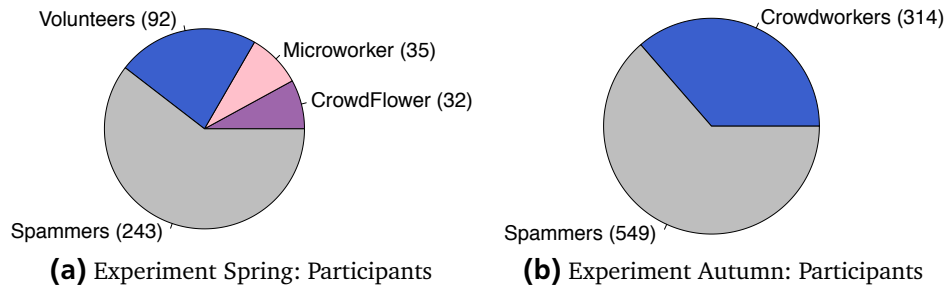
---

## **C.2 Demographics of Participants in Experiment Spring and Experiment Autumn**

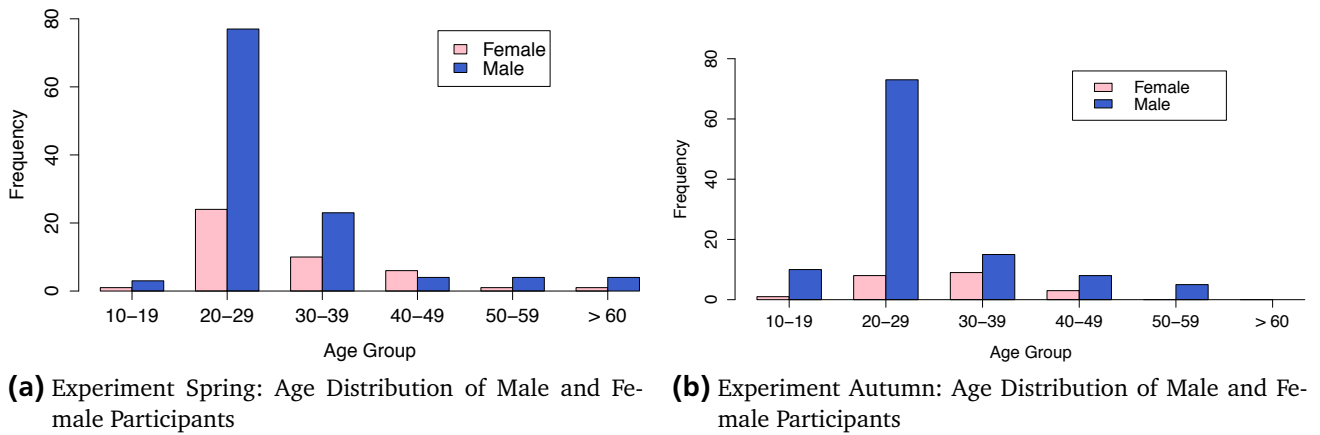
---

The population sample that took part in the experiment are shown in Figure C.6. In Experiment Spring, a total of 402 participants took part in the experiment but after filtering 243 spammers only 159 entries were valid for analysis. In Experiment Autumn, a total of 929 participants took part in the experiment, however 615 spammers were identified and filtered out. The results from 314 valid participants were analysed. The ages of male and female participants in the two experiments are listed in Table C.1 and shown in Figure C.7. The age distribution of participants in both experiments across treatment conditions are shown in Table C.3. The distribution of the gender of participants across treatment conditions for Experiment Spring and Experiment Autumn is shown in Table C.2. The level of education of participants in the experiments across treatment conditions are shown in Table C.4 and plotted in Figure C.8. The countries of participants across treatment conditions for Experiment Spring are shown in Table C.5 and for Experiment Autumn in Table C.6.





**Figure C.6:** Distribution of Participants across Experiments



**Figure C.7:** Distribution of Participants across Experiments

Experiment Spring			
Age	Female	Male	Total
10-19	1	3	4
20-29	24	77	101
30-39	10	23	33
40-49	6	4	10
50-59	1	4	5
> 60	1	4	5
Unknown	0	0	1
<b>Total</b>	<b>43</b>	<b>115</b>	<b>159</b>
Experiment Autumn			
Age	Female	Male	Total
10-19	7	35	42
20-29	22	165	187
30-39	17	43	60
40-49	5	14	19
50-59	1	5	6
> 60	0	0	0
Unknown	0	0	0
<b>Total</b>	<b>52</b>	<b>262</b>	<b>314</b>

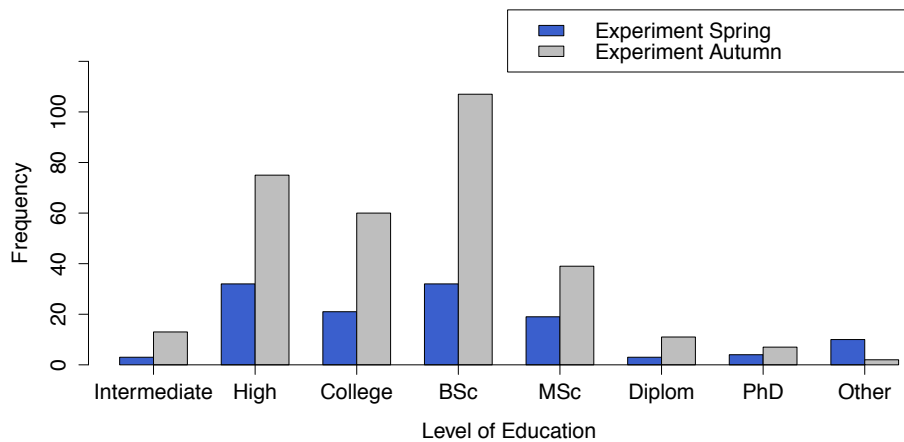
**Table C.1:** Age Distribution of Male and Female Participants Across Experiments

Experiment Spring					
Gender	A_Sub	A_Super	F_Sub	F_Super	Total
Female	14	11	8	10	43
Male	31	28	30	26	115
Unknown	0	0	1	0	1
<b>Total</b>	45	39	39	36	159
Experiment Autumn					
Gender	A_Sub	A_Super	F_Sub	F_Super	Total
Female	11	18	12	11	52
Male	69	55	64	74	262
<b>Total</b>	80	73	76	85	314

**Table C.2:** Gender Distribution of Participants Across Treatment Conditions

Experiment Spring					
Age	A_Sub	A_Super	F_Sub	F_Super	Total
10–19	1	1	1	1	4
20–29	30	26	23	22	101
30–39	8	8	7	10	33
40–49	5	1	3	1	10
50–59	0	2	1	2	5
> 60	1	1	3	0	5
Unknown	0	0	1	0	1
<b>Total</b>	45	39	39	36	159
Experiment Autumn					
Age	A_Sub	A_Super	F_Sub	F_Super	Total
10–19	11	5	14	12	42
20–29	54	43	41	49	187
30–39	10	17	15	18	60
40–49	4	6	5	4	19
50–59	1	2	1	2	6
> 60	0	0	0	0	0
<b>Total</b>	80	73	76	85	314

**Table C.3:** Age Distribution of Participants across Treatment Conditions



**Figure C.8:** Level of Education of Participants across Experiments

<b>Experiment Spring</b>					
<b>Level of Education</b>	<b>A_Sub</b>	<b>A_Super</b>	<b>F_Sub</b>	<b>F_Super</b>	<b>Total</b>
Intermediate School Cert.	1	1	0	1	3
High School Graduate	11	7	10	8	36
College Degree	3	5	9	6	23
Bachelor's Degree (BSc)	13	15	8	12	48
Master's Degree (MSc)	13	5	7	3	28
Diplom Degree	0	3	1	1	5
Doctorate Degree (PhD)	1	1	0	3	5
Other	3	2	4	2	11
<b>Total</b>	<b>45</b>	<b>39</b>	<b>39</b>	<b>36</b>	<b>159</b>
<b>Experiment Autumn</b>					
<b>Level of Education</b>	<b>A_Sub</b>	<b>A_Super</b>	<b>F_Sub</b>	<b>F_Super</b>	<b>Total</b>
Intermediate School Cert.	5	2	1	5	13
High School Graduate	20	16	15	24	75
College Degree	10	13	20	17	60
Bachelor's Degree (BSc)	23	31	24	29	107
Master's Degree (MSc)	12	10	11	6	39
Diplom Degree	7	0	2	2	11
Doctorate Degree (PhD)	2	0	3	2	7
Other	1	1	0	0	2
<b>Total</b>	<b>80</b>	<b>73</b>	<b>76</b>	<b>85</b>	<b>314</b>

**Table C.4:** Level of Education across Treatment Conditions

<b>Experiment Spring</b>					
<b>Country</b>	<b>A_Sub</b>	<b>A_Super</b>	<b>F_Sub</b>	<b>F_Super</b>	<b>Total</b>
Argentina	0	0	0	1	1
Bangladesh	7	10	7	5	29
Bosnia Herzegovina	0	0	1	2	3
Bulgaria	0	2	1	1	4
Canada	0	0	0	1	1
France	0	0	1	1	2
Germany	13	8	8	10	39
India	6	2	2	1	11
Italy	0	1	0	0	1
Macedonia	1	0	1	1	3
Morocco	0	1	0	0	1
Nepal	1	0	0	1	2
Pakistan	0	0	1	0	1
Poland	1	0	0	0	1
Portugal	1	0	0	0	1
Romania	1	2	1	1	5
Russia	0	1	0	0	1
Serbia	1	0	1	0	2
Singapore	0	0	0	1	1
Sri Lanka	0	1	0	1	2
Sweden	0	0	1	0	1
United Kingdom	1	1	0	0	2
USA	12	9	12	8	41
Unknown	0	1	2	1	4
<b>Total</b>	<b>45</b>	<b>39</b>	<b>39</b>	<b>36</b>	<b>159</b>

**Table C.5:** Distribution of Participants across Countries and across Treatment Conditions

Experiment Autumn					
Country	A_Sub	A_Super	F_Sub	F_Super	Total
Algeria	0	0	1	0	1
Bahrain	0	0	1	0	1
Bangladesh	7	5	5	4	21
Bosnia Herzegovina	3	0	1	1	5
Bulgaria	1	0	3	1	5
Canada	4	3	0	3	10
Croatia	0	1	0	1	2
Egypt	0	0	1	0	1
Estonia	0	0	0	1	1
Greece	0	0	0	1	1
Hungary	1	0	1	0	2
India	16	18	6	6	46
Indonesia	2	1	2	3	8
Italy	1	0	1	0	2
Ireland	0	0	0	1	1
Jamaica	1	0	1	0	2
Lithuania	1	0	1	0	2
Macedonia	3	1	4	5	13
Malaysia	3	2	1	1	7
Morocco	1	0	3	0	4
Nepal	13	9	17	24	63
Nigeria	1	1	0	0	2
Norway	0	1	0	0	1
Pakistan	4	3	4	4	15
Philippines	1	0	2	3	6
Poland	0	1	0	3	4
Portugal	0	0	1	1	2
Romania	5	6	3	4	18
Saudi Arabia	0	0	0	1	1
Serbia	1	5	4	3	13
Spain	0	1	0	0	1
Slovenia	2	0	0	0	2
Sri Lanka	5	4	6	5	20
United Kingdom	1	1	0	0	2
USA	1	6	4	5	16
Vietnam	1	1	1	2	5
Unknown	0	3	2	2	7
<b>Total</b>	<b>80</b>	<b>73</b>	<b>76</b>	<b>85</b>	<b>314</b>

**Table C.6:** Distribution of Participants across Countries and across Treatment Conditions

---

### C.3 Descriptive Statistics for Experiment Spring and Experiment Autumn

---

In the following sections, descriptive statistics for Experiment Spring and Experiment Autumn are presented. The following summary statistics are presented for each question [83]:

- The **minimum and maximum** ratings or scores for a particular question  $q$ .
- The **first quartile** which is the middle score between the minimum score and the median of the question.
- The **median** or second quartile which is the middle score of that question.
- The **third quartile** which is the middle score between the median and the maximum score.
- The **Mean (M)** or average which is the sum of the scores for that question divided by the number of scores for that question as shown in Equation C.1.

$$M_q = \frac{\sum_{i=1}^n score_q(i)}{n} \quad (C.1)$$

Where  $n$  is the number of scores for question  $q$  and  $score_q$  is the rating or score given by a single user to question  $q$ .

- The **Standard Deviation (SD)** is the average deviation from the mean. It describes the range and diversity of scores for a question. This is calculated in Equation C.2 as the square root of the average of the sum of squared differences (from the mean) of the scores of that question.

$$SD_q = \sqrt{\frac{\sum_{i=1}^n (score_q(i) - M_q)^2}{n}} \quad (C.2)$$

Where  $n$  is the number of scores for question  $q$ ,  $score_q$  is the score given by a single user to question  $q$  and  $M_q$  is the mean score for question  $q$ .

- The **Standard Error (SE)** is an estimate of the amount of sampling error. It is calculated as the standard deviation divided by the square-root of the number of scores for that question, see Equation C.3

$$SE_q = \frac{SD_q}{\sqrt{n}} \quad (C.3)$$

Where  $SD_q$  is the standard deviation for question  $q$  and  $n$  is the number of scores for question  $q$ .

- The **skew** which describes the asymmetry of the distribution of the scores of the question. A skew of zero indicates a symmetrical distribution. A negative skew indicates an asymmetrical distribution with a tendency of scores to the right, with a long left tail and a positive skew indicates a shift of scores to the left having a long right tail.
- The **kurtosis** which describes the shape of the distribution. A negative kurtosis indicates a platykurtic distribution (a wide peak around the mean) and a positive kurtosis a leptokurtic distribution (a narrow peak around the mean).

---

#### C.3.1 Descriptive Statistics for AScore and FolkRank

---

The descriptive statistics for AScore and FolkRank are presented below. Table C.7 gives the summary statistics for all questions on AScore for Experiment Spring and Experiment Autumn. Table C.8 shows the summary statistics of the answers given to FolkRank for Experiment Spring and Experiment Autumn.

Experiment Spring										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.35	5	7	1.49	-0.03	-0.77	0.07
2)	1	3	4	4.00	5	7	1.58	-0.05	-0.62	0.08
3)	1	4	5	4.55	6	7	1.51	-0.27	-0.41	0.07
1 - 3	1	3	4	4.30	5	7	1.54	-0.13	-0.60	0.04
4)	1	4	4	4.42	5	7	1.51	-0.25	-0.44	0.07
5)	1	3	4	4.09	5	7	1.67	-0.07	-0.85	0.08
6)	1	3	4	4.26	5	7	1.53	0.00	-0.72	0.08
4 - 6	1	3	4	4.26	5	7	1.58	-0.12	-0.68	0.04
7)	1	3	4	4.23	6	7	1.67	-0.06	-0.86	0.08
8)	1	3	4	4.34	5	7	1.62	-0.12	-0.66	0.08
9)	1	3	4	3.91	5	7	1.77	0.03	-0.85	0.09
7 - 9	1	3	4	4.16	5	7	1.69	-0.07	-0.78	0.05
Experiment Autumn										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.30	5	7	1.52	-0.08	-0.80	0.05
2)	1	3	4	4.07	5	7	1.48	-0.04	-0.71	0.05
3)	1	3	4	4.13	5	7	1.47	-0.09	-0.70	0.05
1 - 3	1	3	4	4.17	5	7	1.49	-0.06	-0.73	0.03
4)	1	3	5	4.56	6	7	1.50	-0.22	-0.66	0.05
5)	1	3	4	4.00	5	7	1.59	-0.02	-0.68	0.06
6)	1	3	4	4.37	5	7	1.45	-0.16	-0.60	0.05
4 - 6	1	3	4	4.31	5	7	1.53	-0.15	-0.64	0.03
7)	1	3	4	4.20	5	7	1.51	-0.01	-0.66	0.05
8)	1	4	5	4.47	5	7	1.42	-0.22	-0.42	0.05
9)	1	3	4	4.27	5	7	1.51	-0.27	-0.56	0.05
7 - 9	1	3	4	4.31	5	7	1.48	-0.18	-0.55	0.03

**Table C.7:** Experiment Spring and Experiment Autumn: Summary Statistics for AScore

Experiment Spring										
Question	Min.	1st Quar- tile	Median	Mean	3rd Quar- tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.03	5	7	1.53	-0.18	-0.66	0.08
2)	1	2	4	3.62	5	7	1.59	0.08	-0.84	0.08
3)	1	3	4	4.35	5	7	1.57	-0.06	-0.74	0.08
1 - 3	1	3	4	4.00	5	7	1.59	-0.05	-0.74	0.05
4)	1	3	4	4.10	5	7	1.65	-0.14	-0.74	0.09
5)	1	2	4	3.80	5	7	1.73	0.08	-0.88	0.09
6)	1	3	4	4.00	5	7	1.58	-0.15	-0.71	0.08
4 - 6	1	3	4	3.94	5	7	1.66	-0.07	-0.79	0.05
7)	1	3	4	4.00	5	7	1.64	-0.06	-0.75	0.08
8)	1	3	4	3.99	5	7	1.69	-0.20	-0.92	0.09
9)	1	2	4	3.70	5	7	1.69	-0.07	-0.98	0.09
7 - 9	1	3	4	3.90	5	7	1.67	-0.12	-0.88	0.05
Experiment Autumn										
Question	Min.	1st Quar- tile	Median	Mean	3rd Quar- tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.07	5	7	1.45	0.06	-0.54	0.05
2)	1	3	4	3.80	5	7	1.42	0.03	-0.54	0.05
3)	1	3	4	4.02	5	7	1.38	-0.03	-0.48	0.05
1 - 3	1	3	4	3.96	5	7	1.42	0.02	-0.51	0.03
4)	1	3	4	4.23	5	7	1.42	-0.04	-0.52	0.05
5)	1	2	4	3.98	5	7	1.44	-0.02	-0.49	0.05
6)	1	3	4	4.08	5	7	1.38	-0.02	-0.35	0.05
4 - 6	1	3	4	4.10	5	7	1.41	-0.03	-0.45	0.03
7)	1	3	4	3.98	5	7	1.46	0.03	-0.50	0.05
8)	1	3	4	4.15	5	7	1.41	0.02	-0.33	0.05
9)	1	2	4	3.99	5	7	1.47	-0.10	-0.50	0.05
7 - 9	1	3	4	4.04	5	7	1.45	-0.03	-0.44	0.03

**Table C.8:** Experiment Spring and Experiment Autumn: Summary Statistics for FolkRank



Experiment Spring										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.45	5	7	1.48	-0.03	-0.65	0.10
2)	1	3	4	4.14	5	7	1.61	-0.03	-0.77	0.11
3)	1	4	5	4.72	6	7	1.55	-0.49	-0.24	0.10
1 - 3	1	3	5	4.44	6	7	1.56	-0.19	-0.61	0.06
4)	1	4	5	4.54	6	7	1.48	-0.37	-0.37	0.10
5)	1	3	4	4.12	5	7	1.72	-0.04	-0.88	0.12
6)	1	3	4	4.42	5	7	1.49	-0.09	-0.75	0.10
4 - 6	1	3	4	4.36	5	7	1.57	-0.19	-0.67	0.06
7)	1	3	4	4.29	6	7	1.73	-0.14	-0.97	0.12
8)	1	4	4	4.45	6	7	1.62	-0.23	-0.59	0.11
9)	1	3	4	4.05	5	7	1.79	-0.02	-0.91	0.12
7 - 9	1	3	4	4.27	6	7	1.72	-0.14	-0.83	0.07
Experiment Autumn										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.39	6	7	1.52	-0.09	-0.88	0.08
2)	1	3	4	4.22	5	7	1.46	-0.02	-0.73	0.07
3)	1	3	4	4.20	5	7	1.51	-0.10	-0.75	0.08
1 - 3	1	3	4	4.27	5	7	1.50	-0.07	-0.78	0.04
4)	1	4	5	4.62	6	7	1.53	-0.25	-0.70	0.08
5)	1	3	4	4.05	5	7	1.61	-0.03	-0.77	0.08
6)	1	4	5	4.49	6	7	1.49	-0.24	-0.56	0.07
4 - 6	1	3	4	4.39	6	7	1.56	-0.18	-0.68	0.04
7)	1	3	4	4.36	6	7	1.55	-0.13	-0.71	0.08
8)	1	4	5	4.66	6	7	1.35	-0.22	-0.30	0.07
9)	1	3	5	4.39	6	7	1.51	-0.29	-0.59	0.08
7 - 9	1	3	5	4.47	6	7	1.48	-0.24	-0.52	0.04

**Table C.9:** Experiment Spring and Experiment Autumn: Summary Statistics for A\_Sub

### C.3.2 Descriptive Statistics for Sub-Activities

The descriptive statistics for AScore sub- and super-activities (A\_Sub and A\_Super) as well as for FolkRank sub- and super-activities (F\_Sub and F\_Super) are presented. Table C.9 gives the summary statistics for all questions on A\_Sub for Experiment Spring and Experiment Autumn. Table C.10 gives the summary statistics for all questions on A\_Super for Experiment Spring and Experiment Autumn. Table C.11 gives the summary statistics for all questions on F\_Sub for Experiment Spring and Experiment Autumn. Table C.12 gives the summary statistics for all questions on F\_Super for Experiment Spring and Experiment Autumn.

### C.3.3 Descriptive Statistics for Crowdworkers and Volunteers

The descriptive statistics for crowdworkers and volunteers for Experiment Spring are presented in Table C.13, showing the summary statistics of the answers given to the questions for crowdworkers and volunteers.

Experiment Spring										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.23	5	7	1.50	-0.04	-0.94	0.11
2)	1	3	4	3.84	5	7	1.53	-0.11	-0.53	0.11
3)	1	3	4	4.35	5	7	1.44	-0.03	-0.45	0.10
1 - 3	1	3	4	4.14	5	7	1.50	-0.08	-0.57	0.06
4)	1	3	4	4.30	5	7	1.53	-0.10	-0.48	0.11
5)	1	3	4	4.05	5	7	1.62	-0.11	-0.86	0.12
6)	1	3	4	4.09	5	7	1.57	0.10	-0.68	0.11
4 - 6	1	3	4	4.15	5	7	1.57	-0.05	-0.67	0.07
7)	1	3	4	4.17	5	7	1.59	0.04	-0.71	0.11
8)	1	3	4	4.21	5	7	1.61	0.00	-0.71	0.12
9)	1	3	4	3.75	5	7	1.74	0.08	-0.79	0.12
7 - 9	1	3	4	4.04	5	7	1.66	0.01	-0.70	0.07
Experiment Autumn										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.21	5	7	1.51	-0.07	-0.72	0.08
2)	1	3	4	3.91	5	7	1.49	-0.04	-0.73	0.08
3)	1	3	4	4.04	5	7	1.43	-0.10	-0.66	0.07
1 - 3	1	3	4	4.05	5	7	1.48	-0.06	-0.69	0.04
4)	1	3	5	4.49	6	7	1.46	-0.21	-0.63	0.08
5)	1	3	4	3.94	5	7	1.57	-0.02	-0.60	0.08
6)	1	3	4	4.24	5	7	1.40	-0.11	-0.65	0.07
4 - 6	1	3	4	4.22	5	7	1.50	-0.13	-0.59	0.05
7)	1	3	4	4.04	5	7	1.44	0.09	-0.55	0.08
8)	1	3	4	4.26	5	7	1.47	-0.16	-0.58	0.08
9)	1	3	4	4.13	5	7	1.49	-0.27	-0.54	0.08
7 - 9	1	3	4	4.14	5	7	1.47	-0.12	-0.56	0.04

**Table C.10:** Experiment Spring and Experiment Autumn: Summary Statistics for A\_Super

Experiment Spring										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.09	5	7	1.43	-0.07	-0.50	0.10
2)	1	3	4	3.67	5	7	1.47	0.06	-0.73	0.10
3)	1	3	5	4.59	6	7	1.53	-0.33	-0.53	0.11
1 - 3	1	3	4	3.95	5	7	1.51	-0.13	-0.60	0.06
4)	1	3	4	4.27	5	7	1.55	-0.33	-0.48	0.11
5)	1	2	4	3.67	5	7	1.60	-0.01	-0.82	0.11
6)	1	3	4	4.19	5	7	1.42	-0.11	-0.67	0.10
4 - 6	1	3	4	3.97	5	7	1.55	-0.23	-0.65	0.06
7)	1	3	4	4.24	5	7	1.66	-0.14	-0.84	0.12
8)	1	3	4	4.13	5	7	1.55	-0.26	-0.72	0.11
9)	1	2	4	3.79	5	7	1.70	-0.06	-0.88	0.12
7 - 9	1	3	4	3.96	5	7	1.61	-0.19	-0.80	0.07
Experiment Autumn										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.08	5	7	1.44	0.04	-0.46	0.07
2)	1	3	4	3.89	5	7	1.40	-0.04	-0.46	0.07
3)	1	3	4	4.13	5	7	1.35	-0.08	-0.57	0.07
1 - 3	1	3	4	4.04	5	7	1.40	-0.03	-0.47	0.04
4)	1	3	4	4.24	5	7	1.39	-0.05	-0.50	0.07
5)	1	3	4	3.97	5	7	1.45	-0.02	-0.50	0.07
6)	1	3	4	4.11	5	7	1.37	-0.06	-0.29	0.07
4 - 6	1	3	4	4.11	5	7	1.41	-0.05	-0.42	0.04
7)	1	3	4	4.07	5	7	1.44	-0.02	-0.43	0.07
8)	1	3	4	4.17	5	7	1.40	0.01	-0.33	0.07
9)	1	3	4	3.98	5	7	1.48	-0.14	-0.45	0.08
7 - 9	1	3	4	4.07	5	7	1.44	-0.06	-0.38	0.04

**Table C.11:** Experiment Spring and Experiment Autumn: Summary Statistics for F\_Sub

Experiment Spring										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.11	5	7	1.60	-0.18	-0.81	0.12
2)	1	2	4	3.74	5	7	1.70	0.14	-0.96	0.13
3)	1	3	4	4.31	6	7	1.66	0.04	-0.95	0.12
1 - 3	1	3	4	4.05	5	7	1.67	0.00	-0.89	0.07
4)	1	3	4	4.09	5	7	1.75	-0.03	-0.90	0.13
5)	1	2	4	3.78	5	7	1.81	0.24	-0.99	0.13
6)	1	3	4	3.84	5	7	1.71	-0.04	-0.88	0.13
4 - 6	1	3	4	3.91	5	7	1.76	0.06	-0.92	0.08
7)	1	3	4	4.03	5	7	1.68	-0.05	-0.89	0.13
8)	1	2	4	3.88	5	7	1.78	-0.10	-1.05	0.13
9)	1	2	4	3.59	5	7	1.73	0.06	-0.96	0.13
7 - 9	1	3	4	3.83	5	7	1.74	-0.04	-0.95	0.07
Experiment Autumn										
Question	Min.	1st Quar-tile	Median	Mean	3rd Quar-tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.05	5	7	1.47	0.08	-0.62	0.07
2)	1	3	4	3.72	5	7	1.44	0.11	-0.58	0.07
3)	1	3	4	3.91	5	7	1.40	0.01	-0.41	0.07
1 - 3	1	3	4	3.90	5	7	1.44	0.07	-0.53	0.04
4)	1	3	4	4.22	5	7	1.44	-0.04	-0.56	0.07
5)	1	3	4	3.99	5	7	1.43	-0.02	-0.49	0.07
6)	1	3	4	4.05	5	7	1.38	0.01	-0.42	0.07
4 - 6	1	3	4	4.09	5	7	1.42	-0.01	-0.48	0.04
7)	1	3	4	3.91	5	7	1.47	0.07	-0.56	0.07
8)	1	3	4	4.13	5	7	1.42	0.02	-0.35	0.07
9)	1	3	4	3.99	5	7	1.46	-0.06	-0.57	0.07
7 - 9	1	3	4	4.01	5	7	1.45	0.01	-0.49	0.04

**Table C.12:** Experiment Spring and Experiment Autumn: Summary Statistics for F\_Super

Crowdworkers										
Question	Min.	1st Quar- tile	Median	Mean	3rd Quar- tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4.5	4.49	6	7	1.56	-0.13	-0.71	0.09
2)	1	3	4	4.14	5	7	1.65	-0.04	-0.85	0.09
3)	1	3	5	4.52	6	7	1.52	-0.29	-0.61	0.08
1 - 3	1	3	4	4.38	6	7	1.59	-0.17	-0.73	0.05
4)	1	3	5	4.52	6	7	1.61	-0.24	-0.81	0.09
5)	1	3	4	4.25	5	7	1.70	-0.11	-0.81	0.09
6)	1	3	4	4.41	6	7	1.58	-0.14	-0.72	0.09
4 - 6	1	3	4	4.39	6	7	1.63	-0.17	-0.77	0.05
7)	1	3	4	4.22	6	7	1.82	-0.04	-1.01	0.10
8)	1	3	4	4.32	6	7	1.80	-0.21	-0.92	0.10
9)	1	3	4	4.16	6	7	1.85	-0.13	-0.94	0.10
7 - 9	1	3	4	4.23	6	7	1.82	-0.13	-0.95	0.06
Volunteers										
Question	Min.	1st Quar- tile	Median	Mean	3rd Quar- tile	Max.	Standard Deviation	Skew	Kurtosis	Standard Error
1)	1	3	4	4.05	5	7	1.47	-0.15	-0.60	0.07
2)	1	2	4	3.63	5	7	1.58	0.00	-0.80	0.07
3)	1	3	5	4.49	6	7	1.57	-0.21	-0.54	0.07
1 - 3	1	3	4	4.06	5	7	1.58	-0.11	-0.64	0.04
4)	1	3	4	4.14	5	7	1.55	-0.29	-0.46	0.07
5)	1	2	4	3.72	5	7	1.68	0.08	-0.89	0.08
6)	1	3	4	3.97	5	7	1.54	-0.11	-0.71	0.07
4 - 6	1	3	4	3.94	5	7	1.60	-0.11	-0.73	0.04
7)	1	3	4	4.03	5	7	1.61	-0.11	-0.77	0.08
8)	1	3	4	4.09	5	7	1.61	-0.19	-0.71	0.08
9)	1	2	4	3.64	5	7	1.71	0.00	-1.00	0.08
7 - 9	1	3	4	3.92	5	7	1.65	-0.11	-0.83	0.04

**Table C.13:** Summary Statistics for Crowdworkers and Volunteers

	Hypothesis	Experiment Spring	Experiment Autumn
AScore vs. FolkRank	1A: Relevance	d=0.19	d=0.14
	2A: Novelty	d=0.20	d=0.14
	3A: Diversity	d=0.16	d=0.19
A_Sub vs. F_Sub	1A: Relevance	d=0.21	d=0.16
	2A: Novelty	d=0.20	d=0.19
	3A: Diversity	d=0.12	d=0.27
A_Super vs. F_Super	1A: Relevance	d=0.06	d=0.11
	2A: Novelty	d=0.17	d=0.09
	3A: Diversity	d=0.12	d=0.09
A_Sub vs. A_Super	1B: Relevance	d=0.20	d=0.15
	2B: Novelty	d=0.11	d=0.11
	3B: Diversity	d=0.13	d=0.22
F_Sub vs. F_Super	1B: Relevance	d=0.04	d=0.10
	2B: Novelty	d=0.09	d=0.02
	3B: Diversity	d=0.13	d=0.04

**Table C.14:** Experiment Spring and Experiment Autumn: Results of Cohen's d

#### C.4 Inference Statistics - Cohen's d Effect Size

The Cohen's d effect sizes for all hypotheses are shown in Table C.14, comparing AScore with FolkRank, A\_Sub with F\_Sub, A\_Super with F\_Super, A\_Sub with A\_Super and F\_Sub with F\_Super.



---

## D Details of the PageRank Algorithm and the Random Surfer Model

---

In the following sections of this appendix, details of the Random Surfer Model and the PageRank algorithm are explained.

---

### D.1 The PageRank Algorithm

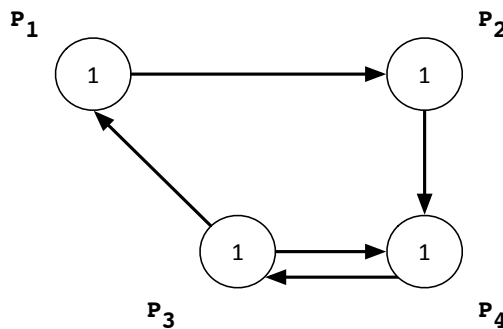
---

PageRank [41, 185] is an example of a graph-based approach based on random walk similarity [61]. It is an example of a link-based ranking algorithm for the Web. Link-based algorithms explore the graph structure of the Web rather than the content of documents [234]. The main idea in PageRank is that a web page is important if many other web pages are linking to it, especially if these web pages are important themselves [41, 185]. The hyperlink structure of the Web can be considered as a directed graph, where the nodes of the graph represent the web pages and the directed edges the hyperlinks between the web pages. The weight of a node contains the page's rank or PageRank value at the end of the PageRank calculation. PageRank can be explained as an iterative process [142] and starts with an initialization of the graph in Round 0, as shown in Figure D.1. The nodes are initialized with a value of 1. The choice of the initial weight in the nodes is irrelevant for the calculation, as this only influences the speed of convergence to a final weight [142]. Then Round 1 in Figure D.2 commences, iteratively increasing the weights of the nodes depending on the links leading to the individual nodes. This weight spreading process is shown in Equation D.1.

$$r_{k+1}(P_i) = \sum_{P_j \in \text{Incoming}(P_i)} \frac{r_k(P_j)}{|P_j|} \quad (\text{D.1})$$

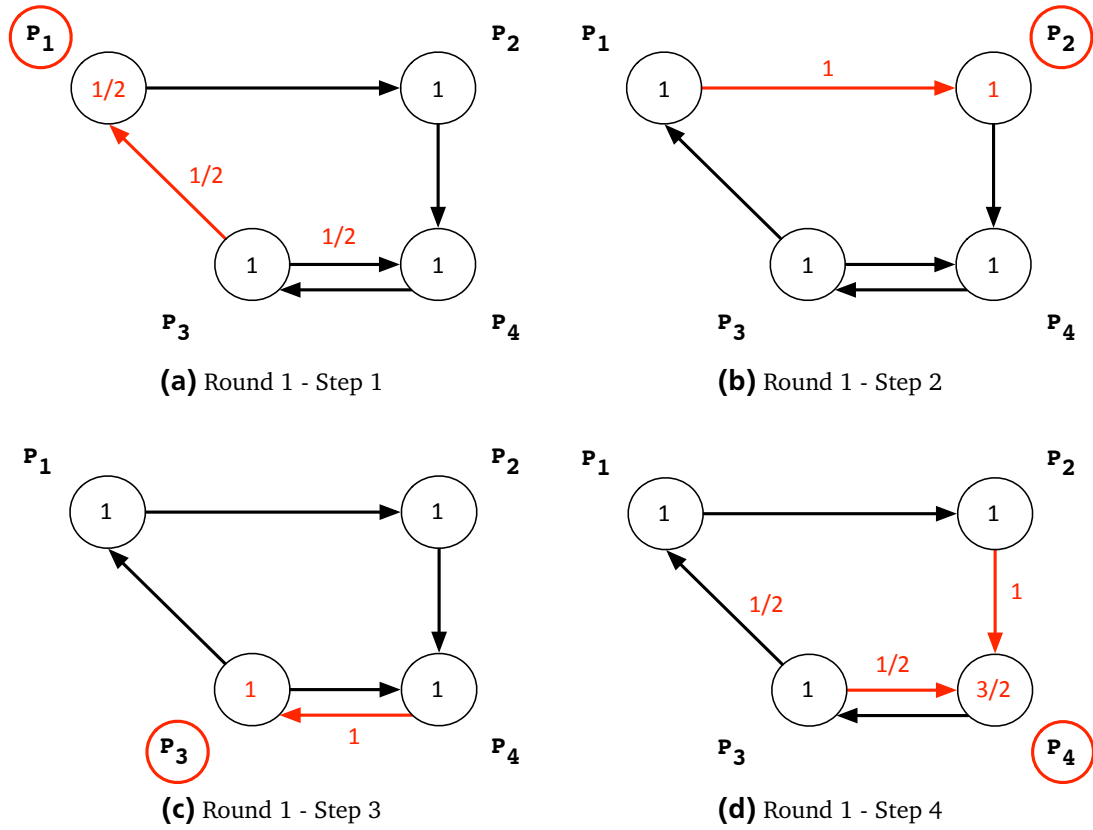
For example, in Round 1 - Step 1, node  $P_3$  has two outgoing links. Therefore the weight spread from  $P_3$  is  $1/2$  to node  $P_1$  and node  $P_4$ .  $P_1$  thus gets the weight  $1/2$  for this round as it has no other incoming links as the one from  $P_3$ . In contrast, in Round 1 - Step 4,  $P_4$  gets weights from  $P_1$  as well as from  $P_2$  and combines these weights to have  $3/4$  at the end of Round 1. Round 1 concludes after 4 iterative steps in the state shown in Figure D.3. Round 2 in Figure D.4 is performed correspondingly but with the initial weights from the final state from Round 1, see Figure D.3. Round 2 concludes with the weights shown in Figure D.5.

**The Power Method:** The graph can also be represented as a square matrix  $H$  called the *transition probability matrix* [142]. Each entry  $h_{ij}$  of  $H$  is the probability of following a hyperlink from page  $i$  to

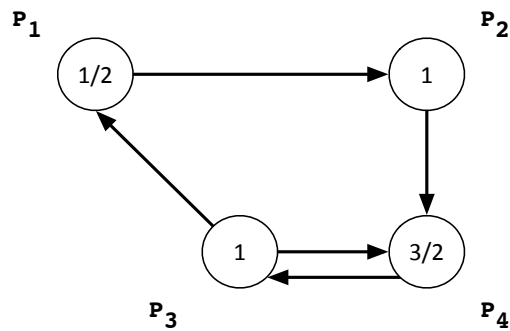


**Figure D.1:** PageRank Iterations - Initialization Round 0

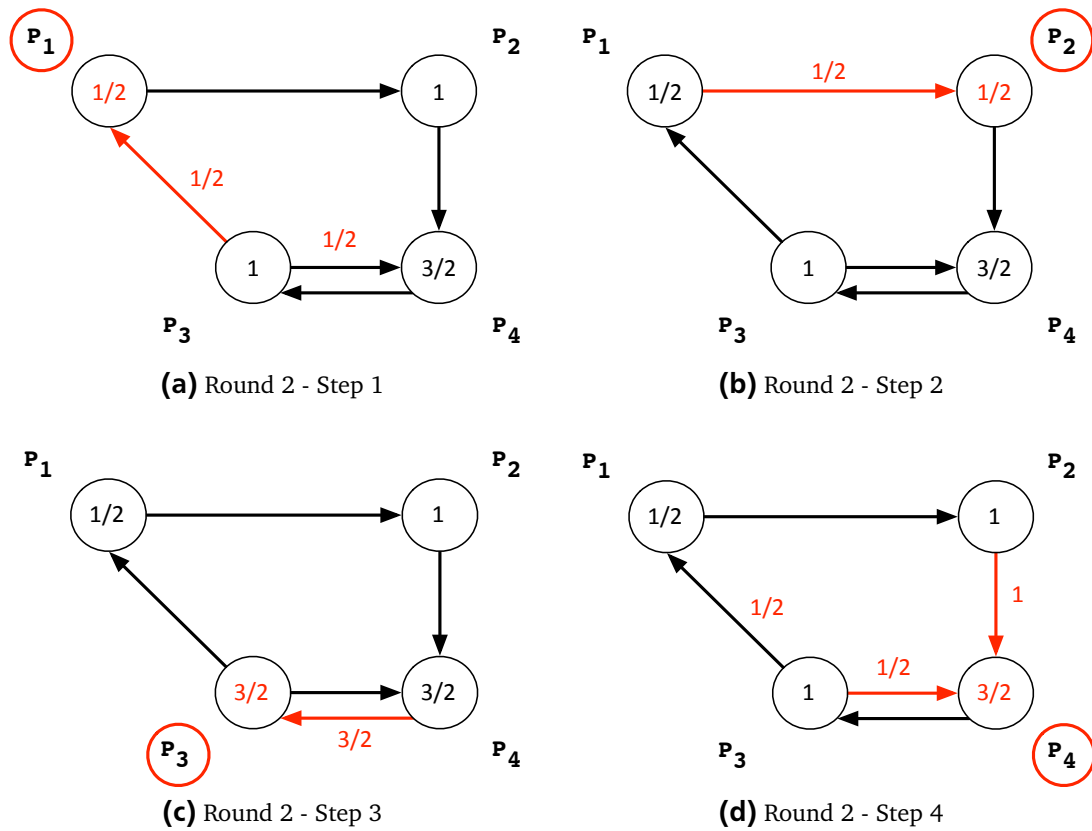




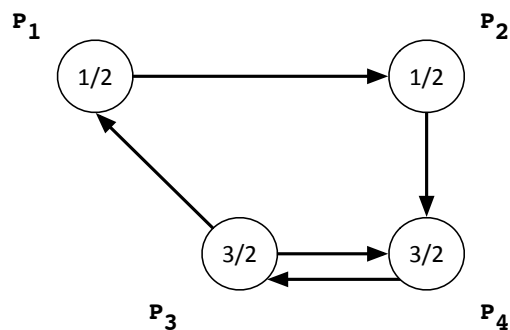
**Figure D.2:** Round 1 - PageRank Iterations Steps



**Figure D.3:** PageRank Iterations - After Round 1



**Figure D.4:** Round 2 - PageRank Iterations Steps



**Figure D.5:** PageRank Iterations - After Round 2

page  $j$ . Therefore, the PageRank vector is calculated as shown in Equation D.2. This calculation is called the *power method* [142].

$$\vec{w}^{(k+1)T} = \vec{w}^{(k)T} H \quad (\text{D.2})$$

Equation D.3 shows the transition probability matrix  $H$  of the graph in Figure D.3

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (\text{D.3})$$

and in Equation D.4 an initial PageRank vector.

$$\vec{w}^{(0)T} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \quad (\text{D.4})$$

$$\vec{w}^{(1)T} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1/2 & 1 & 1 & 3/2 \end{pmatrix} \quad (\text{D.5})$$

Equation D.5 shows the first round of PageRank iterations:  $\vec{w}^{(1)T} = \vec{w}^{(0)T} H$ .

$$\vec{w}^{(2)T} = \begin{pmatrix} 1/2 & 1 & 1 & 3/2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 3/2 & 3/2 \end{pmatrix} \quad (\text{D.6})$$

Equation D.6 shows the second round of PageRank iterations:  $\vec{w}^{(2)T} = \vec{w}^{(1)T} H$ .

$$\vec{w}^{(k+1)T} = \vec{w}^{(k)T} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} = (\dots) \quad (\text{D.7})$$

Finally, as shown in Equation D.7, further PageRank iterations will be needed until the matrix converges to the final PageRank values.

---

## D.2 The Random Surfer Model - Markov Chains

---

The following problems could arise when calculating the power method on the transition probability matrix  $H$  as explained above in Equation D.2 [142]:

- *Rank Sinks*: a node or set of nodes that monopolize the PageRank and do not share their weights thereby causing other nodes to starve or have the PageRank value of zero. This could be caused by *dangling nodes* which are nodes that do not lead to any other nodes i.e. have no outgoing links [142].
- *Cycles*: when the PageRank between nodes flip-flop between iterations. The PageRank value will therefore never be stable and will never converge [142].

There are several questions that arise when considering the PageRank iterative calculation [142] explained above:

- Will the calculation of the power method always converge to a final PageRank vector?
- Will it converge to a positive PageRank vector?
- For which properties of  $H$  is it guaranteed to converge?
- Does this convergence depend on the starting vector  $r_0(P_i)$ ?

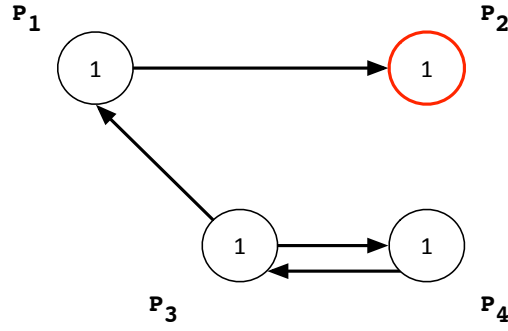
PageRank creates a *Markov chain* by exploiting the Web's hyperlink structure. Markov chains provide answers to the questions stated above [142]. Markov chains can be described by using the *Random Surfer Model* [41, 142, 153, 185]. The random surfer model describes the behaviour of a surfer on the Web who randomly follows a hyperlink on a page thus moving from page to page. Sometimes the random surfer does not follow the hyperlinks on the page but rather types in a URL and thus jumps to a totally unconnected page. Markov chains guarantee that this random walk with the random jumps will converge to a stable state (i.e. a PageRank for every page exists) regardless of where the surfer begins to surf from. Markov chains have the property that for any starting vector, the power method applied to the Markov transition probability matrix converges to a unique positive vector [142]. This ensures that the PageRank vector exists and is independent of the starting vector. For this to be true, the transition probability matrix  $H$  has to be *stochastic* and *primitive*. A primitive matrix is *irreducible* and *aperiodic*<sup>1</sup>. Therefore the following adjustments are made to the transition probability matrix  $H$  [142]:

**Stochasticity Adjustment:** A *stochastic* (or row stochastic) matrix means the sum of the values in each row of the matrix adds up to 1 [142]. This ensures that there are no rows having all zero values. For the random surfer, this means he can get from each node to every other node. The rows with zeros will be replaced with rows of  $1/n$  (where  $n$  is the number of nodes in the graph) thus representing the probability of getting to every node in the graph (other implementations of PageRank simply remove these zero rows). The transition probability matrix is therefore modified to  $S = H + \vec{a} (1/n e^T)$ , where  $a_i$  is 1 if page  $i$  is a dangling node and 0 if not, and  $e$  is a vector with all 1s [142]. In this way, all  $0^T$  rows of  $H$  are replaced with  $1/ne^T$ . If the example above had no link between  $P_2$  and  $P_4$ , then  $P_2$  would be a dangling node, as it leads to no other node as shown in Figure D.6. The matrix  $H$  would therefore look like this, as shown in Equation D.8:

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (D.8)$$

---

<sup>1</sup> The power method applied to a stochastic and irreducible matrix converges to a unique positive vector. Aperiodicity ensures that this converges regardless of the starting vector.



**Figure D.6:** Graph from Figure D.1 but with a dangling node  $P_2$

$H$  would have a row with complete zeros.  $H$  is therefore not a stochastic row matrix. To convert  $H$  to a stochastic matrix  $S$ , the following changes need to be made. First of all, the binary dangling node vector  $\vec{d}$  needs to be determined in Equation D.9:

$$\vec{d} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad (\text{D.9})$$

and  $1/n e^T$  will be  $1/4 e^T$ , where  $n = 4$  (the total number of nodes in the graph) and  $e$  is a vector with all 1s in Equation D.10:

$$1/ne^T = 1/4 \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \quad (\text{D.10})$$

Then the matrix  $S$  is calculated thus:  $S = H + \vec{d} (1/n e^T)$  as shown in Equation D.11

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \quad (\text{D.11})$$

The outer product  $\vec{d} (1/n e^T)$  results in Equation D.12:

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{D.12})$$

The matrix addition results in the row stochastic matrix  $S$  in Equation D.13:

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (\text{D.13})$$

**Primitivity Adjustment:** A matrix is said to be irreducible when every node can be reached from every other node in its graph [142]. An irreducible matrix has each node directly connected to every other node. This means the random surfer can get from one node to every other node. This property is ensured by implementing a *teleportation* matrix [142]. This teleportation matrix models how the random surfer can at anytime type in a URL and jump to any random page. To model this random decision, a parameter  $0 < \alpha < 1$  called the teleportation operator is introduced [142]. For example if  $\alpha$  is 0.85, then this means for 85% of the time, the random surfer will follow the hyperlinks to another node and for 15% of the time he teleports to a random new page. The larger  $\alpha$  gets, the longer it takes for the power method to converge. Finally, aperiodicity is ensured through the teleportation matrix, which ensures all entries in the matrix have non-zero values [142]. The Google Matrix now looks like this:  $G = \alpha S + (1 - \alpha) E$  where  $E = 1/n e e^T$  is the so called teleportation matrix (formed from the outer product of  $e$  and  $e^T$ ). For the example above, the teleportation matrix  $E$  is shown in Equation D.14:

$$E = 1/n e e^T = 1/4 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \quad (D.14)$$

and the Google Matrix  $G$  for this example will be calculated as shown in Equation D.15:

$$G = 0.85 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{pmatrix} + (1 - 0.85) \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \quad (D.15)$$

To finally calculate the PageRank, the power method is applied to  $G$ :  $\vec{w}^{(k+1)T} = \vec{w}^{(k)T} G$  [142].

---

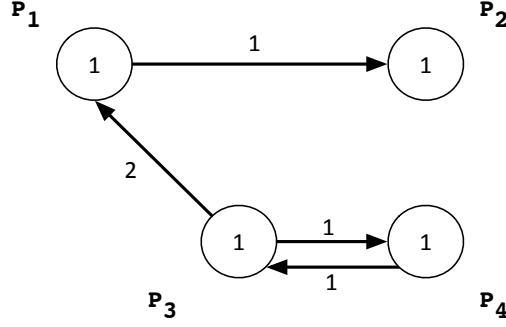
### D.3 The Intelligent Surfer Model - Personalized PageRank

---

The PageRank as it is will calculate the same global PageRank values for every user. This does not make it attractive for personalized rankings. Therefore to personalize PageRank, two main changes need to be made: one to the calculation of the transition probability matrix  $H$  and one to the teleportation matrix  $E$  [142].

**The Hyperlink Transition Probability Matrix  $H$ :** Above, all out-links from a page are given the same transition probability, assuming that the random surfer considers following each out-link with the same probability. However, the surfer on the Web is not necessarily a random surfer but rather an *intelligent surfer* who has preferences and makes informed decisions on which page to visit next [142]. This choice may be influenced by his present location, interests or surfing history. The surfer should therefore perhaps rather be modeled as an intelligent surfer. As a result, the transition probabilities in  $H$  do not have to be uniform, they could rather form a weighting scheme, giving preference to certain out-links in the graph. In order to ensure the row stochastic property of the Markov transition probability matrix is maintained, the values in the rows of the matrix need to be normalized so they sum up to 1 [142]. For example, supposing in Figure D.7,  $P_1$  is preferred twice as much as  $P_4$ , then the out-links of  $P_3$  will need to be adjusted as seen fitting as shown in Equation D.16.

$$H' = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 2/3 & 0 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (D.16)$$



**Figure D.7:** Weighted Graph

**The Teleportation Matrix E:** Furthermore, the teleportation matrix E need not be as uniformly defined as above:  $E = 1/nee^T$ . This matrix could be used to personalize the PageRank calculation by introducing a probability vector called the *personalization vector*  $\vec{p}^T > 0$ . With this vector, the single, global, query-independent PageRank vector  $\vec{w}^T$  becomes an individualized ranking vector. Therefore  $G' = \alpha S + (1 - \alpha)e\vec{p}^T$  [142]. For the example above, with  $H'$  and a personalization vector  $\vec{p}^T$  giving more preference to  $P_1$  and  $P_3$  ( $\vec{p}^T$  however needs to retain the stochastic property) the following Equation D.17 results:

$$\vec{p}^T = \begin{pmatrix} 1/3 & 1/6 & 1/3 & 1/6 \end{pmatrix} \quad (\text{D.17})$$

The teleportation matrix  $E' = e \vec{p}^T$  will then be as shown in Equation D.18.

$$E' = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1/3 & 1/6 & 1/3 & 1/6 \end{pmatrix} = \begin{pmatrix} 1/3 & 1/6 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/3 & 1/6 \end{pmatrix} \quad (\text{D.18})$$

The Google Matrix G will then be as shown in Equation D.19.

$$G' = 0.85 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 2/3 & 0 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{pmatrix} + (1 - 0.85) \begin{pmatrix} 1/3 & 1/6 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/3 & 1/6 \\ 1/3 & 1/6 & 1/3 & 1/6 \end{pmatrix} \quad (\text{D.19})$$

#### D.4 The Biased Surfer Model - Topic-Sensitive PageRank

Similar to personalized PageRank, topic-sensitive PageRank [153] adjusts the teleportation matrix E by introducing a *topic-sensitive probability vector*  $\vec{t}^T > 0$ . This vector represents the query and each entry in this vector represents a node (or topic) in the graph. Therefore to create a topic-sensitive PageRank, some of these nodes (topics) should not belong to the query. Hence if you don't want to query for a certain node (topic) its value in the vector will be set to 0. Topic-sensitive PageRank can be said to have a *biased surfer model* [96], as the surfer is biased towards certain nodes. Analogue to the explanation above,  $G' = \alpha S + (1 - \alpha)e\vec{t}^T$ . And for the example above with  $H'$  and a topic-sensitive vector  $\vec{t}^T$  that is sensitive to the topics  $P_1$  and  $P_3$ , all other nodes (topics) will be given the value 0.  $\vec{t}^T$  however remains stochastic as shown in Equation D.20.

$$\vec{t}^T = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \end{pmatrix} \quad (\text{D.20})$$

This leads further to a teleportation matrix  $E' = e \vec{t}^T$  as shown in Equation D.21.

$$E' = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \end{pmatrix} = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix} \quad (\text{D.21})$$

The Google Matrix will then be as shown in Equation D.22. All parameters needed for the calculations are summarized below.

$$G' = 0.85 \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 2/3 & 0 & 0 & 1/3 \\ 0 & 0 & 1 & 0 \end{pmatrix} + (1 - 0.85) \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix} \quad (\text{D.22})$$

- $H$  Hyperlink transition probability matrix
- $S$  The Stochastic matrix
- $G$  The Google Matrix  $G = \alpha S + (1 - \alpha) E$
- $E$  Teleportation matrix  $E = 1/n e e^T$
- $n$  Number of nodes in the graph
- $e$  A vector having all 1s
- $\alpha$  The teleportation operator having values between 0 and 1
- $\vec{w}$  The PageRank vector
- $\vec{d}$  The binary dangling node vector
- $\vec{p}$  The personalization vector  $\vec{p} > 0$
- $\vec{t}$  The topic-sensitive probability vector  $\vec{t} > 0$





---

## E Author's Publications

---

### E.1 Main Publications

---

1. Mojisola Erdt and Christoph Rensing: *Evaluating Recommender Algorithms for Learning using Crowdsourcing*. In: Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies, ICALT 2014, Pages 513–517, IEEE Computer Society, July 2014.
2. Mojisola Erdt, Florian Jomrich, Katja Schüler and Christoph Rensing: *Investigating Crowdsourcing as an Evaluation Method for TEL Recommenders*. In: Monica Divitini, Tobias Ley, Stefanie Lindstaedt, Viktoria Pammer, Michael Prilla: Proceedings of ECTEL meets ECSCW 2013, the Workshop on Collaborative Technologies for Working and Learning, Volume 1047, Pages 25–29, CEUR Workshop Proceeding Series, September 2013. ISSN 1613-007.
3. Mojisola Anjorin, Thomas Rodenhausen, Renato Domínguez García, Christoph Rensing: *Exploiting Semantic Information for Graph-based Recommendations of Learning Resources*. In: Andrew Ravenscroft, Stefanie Lindstaedt, Carlos Kloos and Davinia Hernández-Leo: 21st Century Learning for 21st Century Skills. Proceedings of the 7th European Conference on Technology Enhanced Learning, EC-TEL 2012, Lecture Notes in Computer Science (LNCS), Volume 7563, Pages 9–22, Springer, September 2012. ISBN 978-3-642-33262-3.
4. Mojisola Anjorin, Ivan Dackiewicz, Alejandro Fernández, and Christoph Rensing: *A Framework for Cross-Platform Graph-based Recommendations for TEL*. In: Nikos Manouselis, Hendrik Drachsler, Katrien Verbert, Olga C. Santos: Proceedings of the 2nd Workshop on Recommender Systems for Technology Enhanced Learning, RecSysTEL 2012, Volume 896, Pages 83–88, CEUR Workshop Proceeding Series, September 2012. ISSN 1613-0073.
5. Mojisola Anjorin, Christoph Rensing, Kerstin Bischoff, Christian Bogner, Lasse Lehmann, Anna Lenka Reger, Nils Faltin, Achim Steinacker, Andy Lüdemann, Renato Domínguez García: *CROKODIL - a Platform for Collaborative Resource-Based Learning*. In: Carlos Delgado Kloos, Denis Gillet, Raquel M. Crespo Garcia, Fridolin Wild, Martin Wolpers: Towards Ubiquitous Learning, Proceedings of the 6th European Conference on Technology Enhanced Learning, EC-TEL 2011, Lecture Notes in Computer Science (LNCS), Volume 6964, Pages 29–42, Springer, September 2011. ISBN 9783642239847.
6. Mojisola Anjorin, Doreen Böhnstedt, Christoph Rensing: *Towards Graph-Based Recommendations for Resource-Based Learning using Semantic Tag Types*. In: Steffen Friedrich, Andrea Kienle, Holger Rohland: DeLFI 2011: Die 9. e-Learning Fachtagung Informatik - Poster Workshops Kurzbeiträge, TUD press, September 2011. ISBN 9783942710367.
7. Mojisola Anjorin, Christoph Rensing, Ralf Steinmetz: *Towards Ranking in Folksonomies for Personalized Recommender Systems in E-Learning*. In: Marco de Gemmis, Ernesto William De Luca, Tommaso Di Noia, Aldo Gangemi, Michael Hausenblas, Pasquale Lops, Thomas Lukasiewicz, Till Plumbaum, and Giovanni Semeraro: Proceedings of the second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation, SPIM 2011, Volume 781, Pages 22–25, CEUR Workshop Proceeding Series, October 2011. ISSN 1613-0073.

- 
8. Mojisola Anjorin, Renato Domínguez García, Christoph Rensing: *CROKODIL: a platform supporting the collaborative management of web resources for learning purposes*. In: Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education, ITICSE 2011, Pages 361, Special Interest Group on Computer Science Education (SIGSE), ACM, June 2011. ISBN 978-1-4503-0697-3.

---

## E.2 Co-Authored Publications

---

9. Alejandro Fernández, Mojisola Erdt, Ivan Dackiewicz and Christoph Rensing: *Recommendations from Heterogeneous Sources in a Technology Enhanced Learning Ecosystem*. In: Nikos Manouselis, Hendrik Drachsler, Katrien Verbert, Olga C. Santos: *Recommender Systems for Technology Enhanced Learning: Research Trends & Applications*, Pages 251–256, Springer, 2014. ISBN 978-1-4939-0529-4
10. Guibing Guo, Mojisola Helen Erdt and Bu Sung Lee: *A Hybrid Recommender System based on Material Concepts with Difficulty Levels*. In: Wong, L. - H. et al.: *Proceedings of the 21st International Conference on Computers in Education, ICCE 2013*, November 2013.
11. Markus Migenda, Mojisola Erdt, Michael Gutjahr, Christoph Rensing: *Semantische Graph-basierte Empfehlungen zur Unterstützung des Ressourcen-basierten Lernens*. In: Andreas Breiter, Dorothee Meier, Christoph Rensing: *Proceedings der Pre-Conference Workshops der 11. e-Learning Fachtagung Informatik - DeLFI 2013*, Pages 67–68, Logos Verlag, September 2013. ISBN 978-3-8325-3470-7.
12. Thomas Rodenhausen, Mojisola Anjorin, Renato Domínguez García, Christoph Rensing, Ralf Steinmetz: *Ranking Resources in Folksonomies by Exploiting Semantic Information*. In: *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies, i-KNOW '12*, ACM, September 2012. ISBN 978-1-4503-1242-4.
13. Thomas Rodenhausen, Mojisola Anjorin, Renato Domínguez García, Christoph Rensing: *Context Determines Content - An Approach to Resource Recommendation in Folksonomies*. In: Bamshad Mobasher, Dietmar Jannach, Werner Geyer, Andreas Hotho: *Proceedings of the 4th ACM RecSys workshop on Recommender Systems and the Social Web*, Pages 17–24, ACM, September 2012. ISBN 978-1-4503-1638-5.
14. Renato Domínguez García, Matthias Bender, Mojisola Anjorin, Christoph Rensing, Ralf Steinmetz: *FReSET - An Evaluation Framework for Folksonomy-based Recommender Systems*. In: Bamshad Mobasher, Dietmar Jannach, Werner Geyer, Andreas Hotho: *Proceedings of the 4th ACM RecSys workshop on Recommender Systems and the Social Web*, Pages 25–28, ACM, September 2012. ISBN 978-1-4503-1638-5.
15. Christoph Rensing, Christian Bogner, Thomas Prescher, Renato Domínguez García, Mojisola Anjorin: *Aufgabenprototypen zur Unterstützung der Selbststeuerung im Ressourcen-basierten Lernen*. In: Holger Rohland, Andrea Kienle, Steffen Friedrich: *DeLFI 2011 - Die 9. e-Learning Fachtagung Informatik*, Pages 151–162, Köllen Verlag, September 2011. ISBN 9783885792826.
16. Sebastian Harrach, Mojisola Anjorin: *Optimizing collaborative learning processes by using recommendation systems*. In: *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education, ITICSE 2011*, Pages 389, Special Interest Group on Computer Sci-

---

ence Education (SIGSE), ACM, June 2011. ISBN 978-1-4503-0697-3.

17. Christoph Rensing, Stephan Tittel, Mojisola Anjorin: *Location based Learning Content Authoring and Content Access in the docendo platform*. In: Franco Zambonelli, Scott F. Midkiff: PerCom-WORKSHOPS 2011: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops, Pages 165–170, IEEE eXpress Conference Publishing, March 2011. ISBN 9781424495283.



---

## F Supervised Student Theses

---

### F.1 Master Theses

---

- KOM-D-0464. Clément Benaych. *Exploiting Social Networks to Recommend Resources in Social Bookmarking Applications*. Master Thesis, Technische Universität Darmstadt, April 2013.
- KOM-D-461. Markus Migenda. *Supporting Resource-based learning using Semantic Graph-based Recommendations*. Master Thesis, Technische Universität Darmstadt, March 2013.
- KOM-D-0445. Thomas Rodenhausen. *Ranking Resources in Folksonomies by Exploiting Semantic and Context-specific Information*. Master Thesis, Technische Universität Darmstadt, January 2012.

### F.2 Bachelor Theses

---

- KOM-B-0465. Florian Jomrich. *Crowdsourcing as an Online Evaluation Method for Recommender Systems for E-learning*. Bachelor Thesis, Technische Universität Darmstadt, July 2013.
- KOM-S-0428. Gustavo Rocha. *Automatic Extraction of RDF Data from HTML Webpages using Web Mining Techniques*. Bachelor Thesis, Technische Universität Darmstadt, March 2012.



---

## G Curriculum Vitae

---

### Personal Details

Name	Mojisola Helen Erdt geb. Anjorin
Date of Birth	16. September 1980
Place of Birth	Zaria, Nigeria
Nationality	German, Nigerian

### Education

07/2010–present	<b>Doctoral researcher,</b> KOM - Multimedia Communications Lab, Technische Universität Darmstadt
10/2003–02/2006	<b>Diplom in Computer Science</b> with subsidiary course Psychology, Technische Universität Darmstadt
10/2002–09/2003	<b>Year Abroad,</b> Computer Science, Technische Universität Wien, Austria
10/1999–09/2002	<b>Vordiplom,</b> Computer Science, Technische Universität Darmstadt
09/1998–09/1999	<b>Studienkolleg,</b> Preparatory course for foreigners, Johann Wolfgang Goethe Universität, Frankfurt am Main
01/1992–10/1997	<b>Secondary School,</b> Abeokuta Girls' Grammar School, Ogun state, Nigeria

### Academic and Working Experience

07/2010–12/2013	<b>Research Assistant,</b> Knowledge Media Research Group, KOM - Multimedia Communications Lab, Technische Universität Darmstadt
09/2006–06/2010	<b>Functional Specialist/ Business Analyst,</b> Human Resources IT, Deutsche Bank, Frankfurt am Main





---

## **H Erklärung laut §9 der Promotionsordnung**

---

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe. Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

*Darmstadt, 2014*

---

Dipl.-Inform. Mojisola Helen  
Erdt geb. Anjorin