Social Tagging und Folksonomies

ein Ansatz zur kollaborativen Indexierung

Bachelorarbeit

im Fach Informatik

angefertigt an der Technischen Fakultät der Christian-Albrechts-Universität zu Kiel

vorgelegt von: Thorge Petersen

Betreuer: Prof. Dr. Michael Hanus

Kiel, 24.03.2013

Ich versichere hiermit, dass die vorliegende Arbeit selbständig und ohne Nutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Weitere Personen waren an der geistigen Erstellung der Arbeit nicht beteiligt. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum Unterschrift

Inhaltsverzeichnis

1 Vorwort

Während zu Beginn des World Wide Web die Ordnung der Webinhalte noch von Hand erledigt wurde und heutige Suchmaschinen als eigenhändig gepflegte Linklisten präsent waren, so machten die zunehmende Anzahl aktiver Internet-Nutzer und die daraus resultierende Menge an Daten Volltextsuchmaschinen notwendig. Heutzutage stoßen in manchen Bereichen auch diese auf Grenzen, und somit stellt sich die Frage, wie man die Fülle an Informationen geeignet ordnen kann.

Eine möglicher Ansatz stellt dabei die gemeinschaftliche Indexierung dar, das *social tagging* oder auch *collaborative tagging*, welches Gegenstand dieser Arbeit ist.

Im Information Retrieval[1] werden bei der Indexierung einer Wissensdomäne Wissensobjekte mittels zugeordneter Deskriptoren beschrieben. So werden z.B. Schlagwörter (engl. Tags) einem Dokument (Webseite, PDF, JPG, AVI etc.) zugeordnet, um dessen Inhalt zu beschreiben.

Wird die Menge der zu verschlagwortenden Wissensobjekte unkontrollierbar groß oder besteht aus anderem Grund keine Möglichkeit, Experten zur Indexierung heranzuziehen, so bieten „social tagging“ und die daraus resultierenden „folksonomies“ einen Ansatz zur Verschlagwortung. Dieser stellt eine Form der freien kollektiven Inhaltserschließung dar und findet in heutigen Web 2.0-Diensten häufig Anwendung, so z.B. im sozialen Bookmarkingdienst Delicious[2].

Die ursprüngliche Intention dieser Arbeit resultierte aus dem Problem einiger Dozenten der Philosophischen Fakultät der Christian-Albrechts-Universität zu Kiel, Videodateien adäquat zu ordnen. Es bestand die Überlegung eine Videosammlung zu erstellen, die primär der Ordnung vorhandener Datenbestände (mit teilweise bereits existierenden Metadaten in verschiedenen Formaten) dienen muss, wobei aber die Indexierung und Abfrage der Daten nicht ausschließlich von Experten vorgenommen werden sollte. Das Ordnungssystem soll also einerseits den Dozenten helfen, schnell bestimmte Videos zu finden, andererseits den Studenten Zugang zu bestimmten Wissenssammlungen bieten, was mich schließlich zu dem Konzept der kollaborativen Indexierung führte. Dabei stellte sich die Frage, warum nicht gleich verschiedene Arten von Inhalten gruppiert und geordnet werden könnten, um Wissensgebiete multimedial zu repräsentieren.

Diese Arbeit soll zunächst eine Begriffsklärung genannter Konzepte bieten, wobei u.a. auf die verschiedenen Arten von typischen Ordnungsystemen eingegangen wird um anschließend vergleichend das Konzept des Social Taggings erörtern zu können. Abschließend wird in Abschnitt 4 einer prototypische Rails-Applikation zur gemeinschaftlichen

[1] Fachgebiet der Informationswissenschaft, Informatik und Computerlinguistik, dass sich der computergestützten Suche komplexer Inhalte widmet.

[2] http://www.delicious.com

Indexierung von inhaltsunabhängigen Dokumenten vorgestellt. Dabei werden Konzepte wie Vorschlagssysteme und Tagclouds umgesetzt.

2 Begriffsbildung

Zunächst werden die grundlegenden Begriffe und Konzepte der Indexierung sowie verschiedene Ordnungssysteme beschrieben, um in Kapitel 3 die Vor- und Nachteile von Social Tagging anhand ihrer speziellen Eigenschaften erläutern zu können. Im Nachfolgenden wird mit den Begriffen *Objekt*, *Ressource* oder auch *information resource* eine beliebige Informationseinheit beschrieben, welche ganz gleich ihres Inhalts, zumindest einen eindeutigen Schlüssel zur Referenzierung besitzen muss. Der inhaltliche Zusammenschluss mehrerer Informationseinheiten zu einem Verbund ergibt eine Wissensdomäne[3].

2.1 Tags/Tagging

Ein Tag (engl. Etikett, Schlagwort, Kennzeichnung) ist ein Deskriptor, der ein Objekt beschreibt. Tags sind also objektbezogene Metadaten.

2.1.1 Eigenschaften und mathematische Struktur

Auf technischer Ebene wird beim Tagging ein Datenbankeintrag erstellt, welcher eine Relation zwischen dem zu taggenden Objekt, dem taggenden Nutzer und dem dazugehörigen Label des Tags ausdrückt. Dabei können auch weitere Informationen in die Relation eigehen, so z.B. der Zeitpunkt, wann ein Tag gesetzt wurde. Bezogen auf diese Arbeit sind aus dem Artikel von Müller-Prove insbesondere folgende Aspekte zu erwähnen (vgl. Müller-Prove 2008, S. 16f):

Eine Tag-Instanz ist eine Relation der Form:

$$(Object, Label, User, ...)$$

Dabei muss das Objekt eindeutig referenzierbar sein. Das Label ist in der Regel eine frei wählbare Zeichenkette, wobei grundsätzlich keine Vorgaben zur Beschreibung exisitieren. Jedoch können Vorschlagssysteme das Vokabular vereinheitlichen.

Der letzte Parameter bezeichnet den taggenden Anwender, wobei nach Müller-Prove dieser Wert immer mitgeführt werden muss, um ein mehrfaches Hinzufügen eines Labels zu einem Objekt zu vermeiden. Dies ist jedoch nicht zwingend notwendig. Ein mehrfaches Hinzufügen lässt sich auch durch geeignete Anwendungs-/Datenbanklogik vermeiden,

[3] In der Künstlichen Intelligenz (KI) ein abgegrenztes Wissensgebiet; das Fachgebiet, das auf ein wissensbasiertes System (Expertensystem) abgebildet wird. (Quelle: http://wirtschaftslexikon.gabler.de/Archiv/54781/wissensdomaene-v7.html, Letzter Zugriff: 22. Feb. 2013)

desweiteren ist es eventuell sogar erwünscht, dass Tags frei vergeben werden, ohne Usern zugehörig zu sein. Somit wäre auch ein Tupel, der nur die elementarsten Eigenschaften eines Tags vereint, eine korrekte Tag-Instanz:

$$(Object, Label)$$

Die Zuweisung mehrerer gleicher Labels zu einem Objekt, bzw. die mehrfache Erstellung von Tag-Instanzen mit gleichem Objekt und Label, kann zudem die Signifikanz des Labels bezüglich des Objekts widerspiegeln. Das Mitführen eines Benutzerobjekts ist i.d.R. aber erwünscht, da aus den Daten Statistiken erhoben werden können und sich die Anwendungslogik handlicher gestaltet.

Aus der Mehrfachvergabe resultierend ergibt sich für die Definition eines Tags der folgende Tupel:

$$(Label_i, \{Object_j \mid \exists(Object_j, Label_i, User_x)\}), bzw.$$
$$(Label_i, \{Object_j \mid \exists(Object_j, Label_i)\})$$

Die sich ergebende Strukur aus Objekten und Tags ist ein Hypergraph[4], bei dem eine Tag-Hyperkante alle Objektknoten verbindet, welche mit dem zugehörigen Tag versehen wurden (siehe Abbildung 2.1).

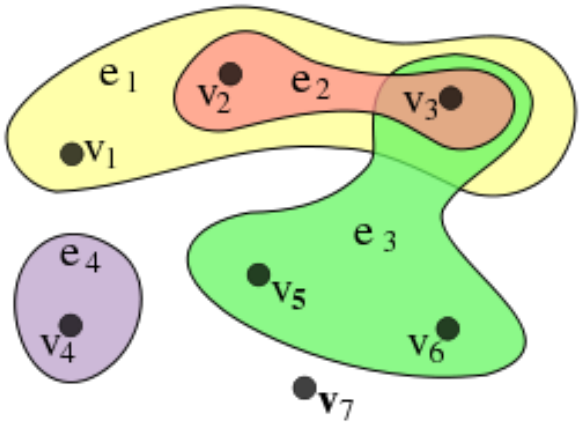

Abbildung 2.1: Graphische Darstellung eines Hypergraphen[5]

Die Knoten $V = \{v_{1,...,}v_7\}$ repräsentieren die Objekte, welche über die farbig hervorgehobenen Hyperkanten $E = \{e_{1,}e_{2,}e_{3,}e_4\} = \{\{v_{1,}v_{2,}v_3\}, \{v_{2,}v_3\}, \{v_{3,}v_{5,}v_6\}, \{v_4\}\}$, den Tags, verbunden sind.

[4]Ein Hypergraph ist ein mathematisches Konstrukt eines Graphen, bei dem eine Kante eine beliebige Anzahl an Knoten verbinden kann. Formal gesehen ist ein Hypergraph H ein Paar $H = (V,E)$, wobei V eine Menge an Knoten und E eine Menge an nicht-leeren Teilmengen von V ist, welche Hyperkanten genannt werden.

[5]Quelle: http://upload.wikimedia.org/wikipedia/commons/5/57/Hypergraph-wikipedia.svg, Letzter Zugriff: 05. Feb. 2013

2.1.2 Die verschiedenen Arten von Tags

Schaut man sich Tags genauer an, so lassen sich gewisse regelmässige Eigenschaften bzw. Funktionen erkennen. So kann beispielsweise im Web-Dienstleistungsportal Flickr[6] ein gemaltes Bild eines Schiffes auf dem Meer mit inhaltlich beschreibenden Begriffen wie „Schiff“, „Meer“, mit Kategorie beschreibenden Begriffen wie „Maritime Malerei“, aber auch mit Tags zur Beschreibung der persönlichen Empfindung wie „schön“ versehen werden.

Innerhalb der Tagging-Gemeinde von Delicious traten Diskussionen darüber auf, ob bestimmte, das Objekt beschreibende Tags im Widerspruch zu jenen stehen, welche die Kategorie, in die das Objekt fällt, beschreiben. Nach Golder und Huberman ist dies jedoch in Bezug auf eben genannte Arten von Tags irrelevant, da Eigenschaften einer Kategorie von Objekten ja schließlich auch Eigenschaften jedes Objekts eben dieser Kategorie sind. Anhand von Delicious gliedern sie Tags in folgende Kategorien (vgl. Golder and Huberman 2006, S.6):

1. *Identifying What (or Who) it is about*

   Tags, die größtenteils den Inhalt beschreiben. So würden zu diesem Dokument Tags wie „Social Tagging“ und „Folksonomies“ unter diese Kategorie fallen.

2. *Identifying What it Is*

   Tags, die den Typ des Objekts beschreiben. Da Delicious mit URLs arbeitet, ist der wahre Typ der Ressource nur schwer zu ermitteln[7]. Zu diesem Dokument passende Tags wären „Bachelorarbeit“ oder „Text“.

3. *Identfying Who Owns It*

   Tags, die angeben wer, das Objekt erstellt hat bzw. wer der Urheber ist. In diesem Falle dann „Thorge Petersen“.

4. *Refining Categories*

   Unter diese Kategorie fallen Tags, die alleine nicht aussagekräftig genug sind und erst mit anderen Tags zusammen einen Sinn ergeben. So sind Zahlen häufig erst in Verbindung mit Straßennahmen oder Maßeinheiten sinnvoll. Delicious lässt keine Leerzeichen zu und somit ergeben Straßenname und Zahl jeweils einzelne Tags.

[6] http://www.flickr.com

[7] Um nicht den kompletten Datensatz der zur URL gehört auslesen zu müssen, lässt sich der Inhalt nur über den Header und dem damit verbundenen Content- bzw. MIME-Type bestimmen. Dieser ist jedoch leicht fälschbar. Eine Überprüfung mittels Magic Numbers könnte hilfreich sein, ist jedoch auch nicht absolut fälschungssicher.

5. *Identifying Qualities or Characteristics*

   Meinungen des Taggers bezüglich des Objekts. Zum Beispiel „leserlich“ oder „interessant“.

6. *Self Reference*

   Tags, die eine Beziehung zwischen dem Tagger und dem Dokument herstellen. Sie beginnen häufig mit „my“, wie beispielsweise „my_documents“.

7. *Task Organizing*

   Aufgabenbezogene Tags, wie z.B. „todo“ oder „toread“.

2.2 Ordnungssysteme

Im Folgenden werden grundlegende Aspekte verschiedener Ordnungssysteme erläutert, um in Kapitel 3 die Unterschiede zu Folksonomien aufzuzeigen zu können:

2.2.1 Thesauri

Ein Thesaurus (altgriechisch $\theta\eta\sigma\alpha\upsilon\rho\acute{o}\varsigma$ thesaurós, Schatz, Schatzhaus; lat. dann thesaurus) besteht aus einem festgelegten kontrollierten Vokabular und einer gegebenen Menge an Relationen, mit denen die einzelnen Vokabeln verknüpft werden können, und dient der Beschreibung und Repräsentation einer Wissensdomäne. Ein Thesaurus ist auch unter dem Begriff „Wortschatzsammlung“ bekannt. Anhand des Vokabulars, also den Deskriptoren (oder auch „Attributwertebereich“), wird ein Themengebiet beschrieben. Die Relationen dienen hauptsächlich der Kategorisierung und der Verwaltung von Synonymen und Ober-/Unterbegriffen innerhalb des Vokabulars. Oft sind Äquivalenzrelationen zur Beschreibung gleicher Bedeutungen, Hierarchierelationen zur *is-a* bzw. *is-part-of* Beschreibung und Assoziationsrelationen zur Definition verwandter Begriffe vorhanden.

„Unter dem Begriff Synonymie versteht man die Sinnverwandtschaft, lexikalische Ähnlichkeit oder Gleichheit zweier Wörter. Analog sind Wörter synonym, wenn sie eine ähnliche oder sogar gleiche Bedeutung haben.“[8]

[8] Quelle: http://synonyme.woxikon.de, letzter Zugriff: 24. Feb. 2012

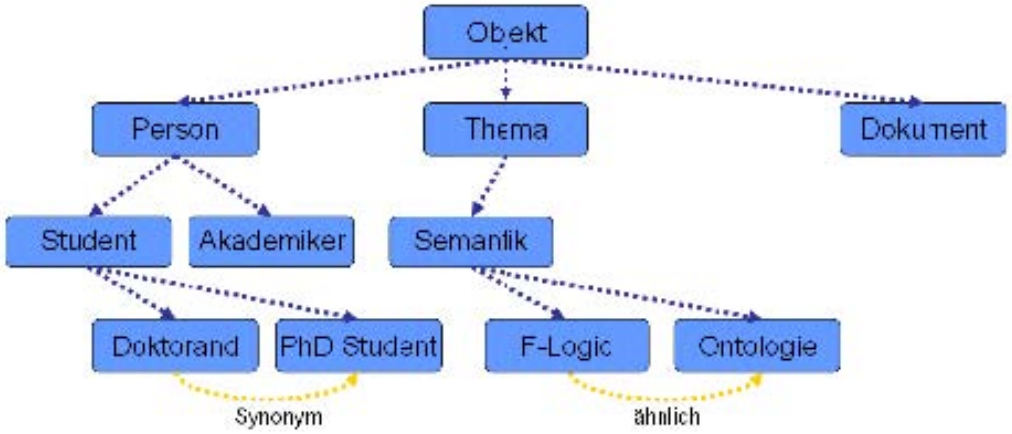

Abbildung 2.2: Beispielhafte graphische Darstellung eines Thesaurus[9]

Die Deskriptoren werden dabei zu einem „descriptor set“ zusammengefasst, bei dem Nicht-Deskriptoren zur Realisierung der Relationen verwendet werden. Letztere dienen nicht der Indexierung selbst, sondern nur der Verknüpfung des Vokabulars. „[. . .] non-descriptors, only refer to the preferred term (descriptor) that must be used to index and retrieve the resource.“ (Peters 2009, S.125)

So zeigt Abbildung 2.3 einen Ausschnitt des NASA Thesaurus, welche die korrekten Terme enthält, die zur Indexierung und Suche innerhalb der „Aeronautics and Space Database“ genutzt werden. Die fett geschriebenen Begriffe stellen die bevorzugten Deskriptoren dar. Diese sollten zur Indexierung verwendet werden. Begriffe wie *galactic winds* oder *galactose* sind Deskriptoren, während *galactic cosmic rays* oder *space plasmas* Nicht-Deskriptoren darstellen.

galactic winds
(added May 1994)
GS extraterrestrial radiation
. galactic radiation
. . **galactic winds**
RT galactic cosmic rays
space plasmas
stellar winds

galactose
GS organic compounds
. carbohydrates
. . sugars
. . . monosaccharides
. . . . hexoses
. . . . . **galactose**

Galatea
(added July 1995)
DEF A natural satellite of Saturn, orbiting at a mean distance of 62,000 kilometers.
GS celestial bodies
. natural satellites
. . Neptune satellites

Galilean satellites
DEF The four largest and brightest satellites of Jupiter (Io, Europa, Ganymede, and Callisto).
GS celestial bodies
. natural satellites
. . Jupiter satellites
. . . **Galilean satellites**
. . . . Callisto
. . . . Europa
. . . . Ganymede
. . . . Io
RT Charon
Galileo project
Galileo spacecraft
icy satellites
Jupiter (planet)
Triton

Galileo mission
USE **Galileo project**

Galileo probe
DEF The NASA Jupiter atmospheric entry

Abbildung 2.3: Auszug aus dem NASA Thesaurus[10]

[9]Quelle: http://www.ullri.ch/download/Ontologien/ttto13.pdf, S.4, Letzter Zugriff: 22. Feb. 2013

Mit einem Thesaurus lassen sich dann bei der Verschlagwortung Objekte indexieren, wobei jedes Objekt eine unbegrenzte Zahl an Deskriptoren zugewiesen bekommen kann. Bei der Suche mit gegebenen Deskriptoren kann dann mittels der Nicht-Deskriptoren die ursprüngliche Suche erweitert werden. So würde eine Suche nach dem Wort „Galileo mission“ über die USE-Beziehung von „Galileo mission“ zu „Galileo project“ auch Ergebnisse liefern können, die statt mit „Galileo mission“ mit „Galileo project“ bezeichnet wurden. Eine Übersicht der nach DIN 1463-1 bzw. dem internationalen Äquivalent ISO 2788 vorgesehenen Relationstypen liefert Tabelle 2.1:

DIN 1463-1		**ISO 2788**	
BF	Benutzt für	**UF**	Used for
BS	Benutze Synonym	**USE/SYN**	Use synonym
OB	Oberbegriff	**BT**	Broader term
UB	Unterbegriff	**NT**	Narrower term
VB	Verwandter Begriff	**RT**	Related term
SB	Spitzenbegriff	**TT**	Top term

Tabelle 2.1: Abkürzungen und Bezeichnungen der Relationstypen von Thesauri[11]

Die Erstellung des Vokabulars und der Beziehungen sowie die Verschlagwortung wird i.d.R. von Experten durchgeführt, da das kontrollierte Vokabular und die relativ strikte hierarchische Ordnung Wissen über den Thesaurus vorraussetzt. Auch die Suche setzt fachliches Wissen voraus. Dem kann zumindest in gewisser Hinsicht mit einem Vorschlagssystem, welches automatisch Begriffe zur Indexierung und Suche vorschlägt, entgegen gegangen werden. Möchte man selbst einen Thesaurus verwenden, so kann es hilfreich sein, auf öffentlich bereitgestellte bereits bestehende Thesauri auszuweichen. So lassen sich z.B. mit Hilfe des „Europäische Thesaurus Internationale Beziehungen und Länderkunde“ Fachpublikationen bezüglich international- sowie regionalwissenschaftlicher Themen in einer Literaturdatenbank wiederauffinden. Der Zugang hierzu wird in Deutschland von IREON[12] über ein Webportal bereitgestellt.

2.2.2 Ontologien

„Ontologies are the most detailed method of knowledge representation and are meant to serve the 'semantic web', mainly to faciliate the interaction between man and computer, as well as between computer and computer.“ (vgl. Peters 2009, S.124f)

Ontologien gehören zum Bereich der Wissensrepräsentation im Teilgebiet der Künstlichen Intelligenz und dienen im Vergleich mit anderen Ordnungssystemen nicht nur primär der Ordnung einer Wissensdomäne, sondern auch dem Datenaustausch verschiedener Wissensbestände. Sie bieten also die Möglichkeit einer Trennung von Meta-Modell und Inhalt und können Zusammenhänge zwischen Objekten verschiedener Ontologien

[10] Quelle: http://www.sti.nasa.gov/thesvol1.pdf, S. 378, Letzter Zugriff: 05. Feb. 2013
[11] vgl. http://de.wikipedia.org/wiki/Thesaurus#Thesaurus_zur_Dokumentation
[12] https://www.ireon-portal.eu/

über Beziehungen, Zuweisungen, logischen Verknüpfungen, etc. ausdrücken, so dass umfangreiche Suchanfragen möglich sind. Aufgrund ihrer stark relationslastigen Struktur lässt sich Wissen sehr ausdrucksstark ordnen.

Ontologien werden ähnlich wie Thesauri i.d.R. von Experten erstellt, wobei jedoch verschiedene Ansätze zur automatisierten Wissensakquisition existieren. Diese fasst man unter dem Begriff „Ontology Learning“ zusammen.

Die Beschreibung der Ontologien erfolgt häufig mittels formaler Sprachen, wie dem RDF-Schema[13] oder OWL[14].

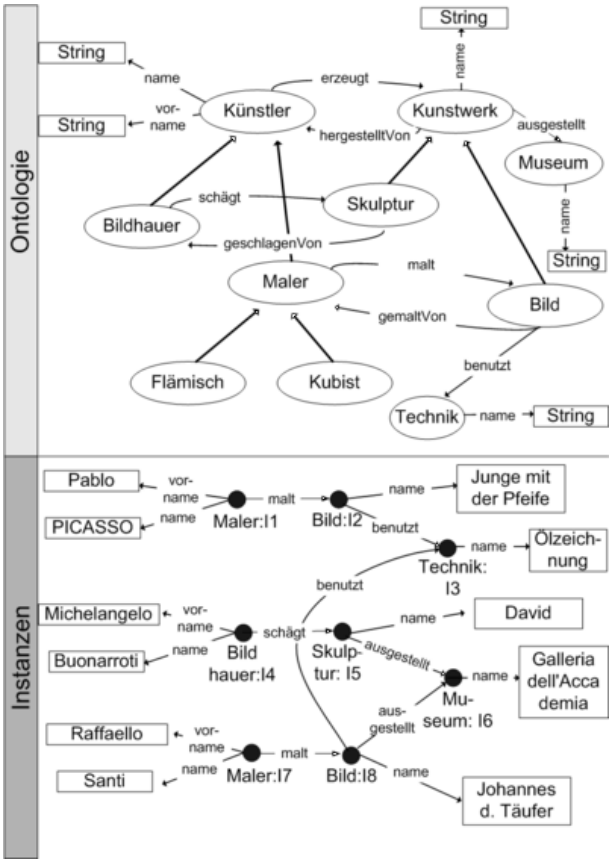

Abbildung 2.4: Graphische Darstellung einer Ontologie eines Museums[15]

Abbildung 2.2.2 zeigt eine graphische Darstellung einer Ontologie eines Museums, bei dem der obere Abschnitt das Meta-Modell und der untere Abschnitt den Inhalt bzw. die Instanzen darstellt. Begriffe sind hier durch Ellipsen gekennzeichnet, wobei Eigenschaften wie Vor- und Nachname von den Künstlern durch Vererbung auch den

[13] „Resource Description Framework Schema“

[14] „Web Ontology Language“, basiert auf dem RDF-Syntax.

[15] Quelle: http://upload.wikimedia.org/wikipedia/commons/e/e3/Ontschichten.gif, Letzter Zugriff: 05. Feb. 2013

Malern zugehörig sind. Die Relationen *gemalt* und *gemaltVon* sind zueinander invers und erben Eigenschaften von den Relationen *erzeugt* und *erzeugtVon*. Sie erweitern also die ursprüngliche Relationen. Die Instanzen stellen die Inhalte dar, welche über das komplexe Meta-Modell ausdrucksstark semantisch strukturiert sind. So hat der Maler Pablo Picasso z.B. das Bild mit dem Namen „Der Junge mit der Pfeife“ gemalt, bei der er die Technik der Ölmalerei benutzt hat.

2.2.3 Taxonomien

Eine Taxonomie (altgriechisch τάξις áxis „Ordnung“ und nómos „Gesetz“) ist ein monohierarchisches[16] Ordnungssystem, welches Objekte in bestimmte Kategorien, auch „Taxa“ genannt, einordnet. Taxonomien sind also Klassifizierungssysteme.

„Unlike thesaurus and ontology, the classification system does not work with natural-language terms but with notations, which are often formed from a combination of numbers and letters“ (Peters 2009, S.126f)

Ein alltägliches Beispiel wäre das Dateisystem eines Computers, bei dem man in Ordnerstrukturen hierarchisch Dateien speichern bzw. einordnen kann. Eine Datei oder ein Ordner ist dann durch die im Pfad vorher angegebenen Ordner klassifiziert.

Taxonomien unterscheiden sich in Bezug auf Thesauri vor allem durch die nicht gegebene Möglichkeit, Relationen zwischen dem Vokabular bzw. den Notationen zu erstellen. „Furthermore, the classification system only displays hierarchical relations between the terms [. . .]“ (Peters 2009, S.127). Graphisch lässt sich eine Taxonomie als Baumstruktur aufzeigen:

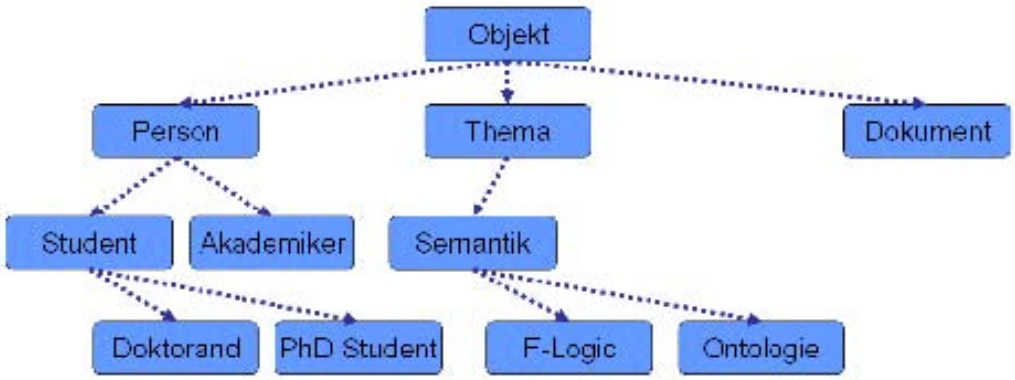

Abbildung 2.5: Graphische Darstellung einer Taxonomie[17]

Resultierend ergibt sich eine sehr genau Einordnung der Objekte in eindeutige Klassen, die eine schnelle Suche ermöglicht, vorausgesetz der Suchende weiß, wo sich ein Objekt befindet. Dies kann problematisch sein. Ebenso ist es mit der Klassifizierung.

[16] In einer Monohierarchie ist jedem Element höchstens ein Element direkt übergeordnet. Man spricht auch von hierarchischer Einfachvererbung.

[17] Quelle: http://www.ullri.ch/download/Ontologien/ttto13.pdf, S.3, Letzter Zugriff: 22. Feb. 2013

Existieren mehrere Kategorien, denen ein Objekt zugeordnet werden könnte, so hat man sich entweder für eine Kategorie zu entscheiden oder man ordnet das Objekt, sowie eine zusätzliche Objekt-Kopie ein und es kommt zu Redundanz. Ersteres impliziert jedoch, dass das Objekt über die nichtpräferierte hierarchische Ordnung unauffindbar ist.

3 Social Tagging/Folksonomies

3.1 Was bedeutet Social Tagging?

„Collaborative tagging describes the process by which many users add metadata in the form of keywords to shared content.“ (Golder and Huberman 2006, S.1)

Social Tagging oder auch *Collaborative Tagging* ist also eine Form der freien Indexierung, da die Verschlagwortung nicht mittels eines kontrollierten Vokabulars geschieht, wie es z.B. bei einem Thesaurus der Fall ist. Die Begriffe, die ein Anwender zur Beschreibung heranzieht, können frei und willkürlich gewählt werden. Der Nutzer indexiert also Objekte, die einer Gemeinschaft oder der Allgemeinheit zugänglich sind. Dabei können einer Ressource beliebig viele Tags zugewiesen werden.

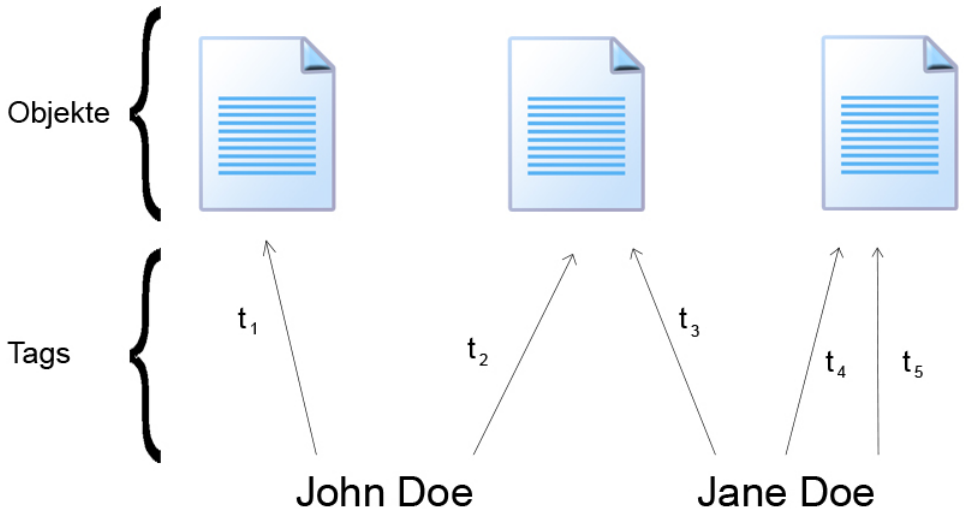

Abbildung 3.1: Graphische Darstellung des Social Taggings

Abbildung 3.1 veranschaulicht Tag-Instanzen[18], also die Beziehungen zwischen Tagger, Tag und Objekt.

Die Tags t_4 und t_5 wurden hier von dem Anwender „John Doe“ einem einzelnen Objekt zugewiesen, t_2 und t_3 von „John Doe“ und „Jane Doe“, welche zusammen eine kollaborative Gruppe darstellen.

[18] vgl. Abschnitt 2.1.1

3.2 Was sind Folksonomies?

Die Gesamtheit der beim Collaborative Tagging entstandenen Tag-Instanzen bilden gemeinsam eine Folksonomie (engl. „folksonomy“). Der Begriff geht dabei ursprünglich auf eine Diskussion zwischen Gene Smith und Thomas Vander Wal innerhalb eines Blogs zurück, wobei letzterer den Begriff neu einführte und wie folgt definierte:

„Folksonomy is the result of personal freetagging of information and objects (anything with a URL) for one's own retrieval. The tagging is done in a social environment (shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information.“ (Wal 2004)

„Folksonomy“ ist eigentlich eine Kombination aus den Wörtern *folk* und *taxonomy* und soll ausdrücken, dass eine Gemeinschaft Objekte mit Tags versieht. Dabei wird in der Literatur des Öfteren kritisiert, dass der Begriff Taxonomie und die damit verbundene hierarchische Struktur nicht auf das Collaborative Tagging zuträfen und somit einen falschen Eindruck erweckten.

“Folksonomies are not classification, since they use neither notations nor relations:“ (Peters 2009, S.154)

Der Begriff *relations* bezieht sich hierbei auf die bei Thesauri gegebene Möglichkeit, Deskriptoren zueinander in Verbindung zu setzen, er bezeichnet nicht die Relationen einer Tag-Instanz. Die in Taxonomien verwendeten Notationen treffen auch nicht auf das Collaborative Tagging zu, welches im Vergleich zu einer Taxonomie nicht-hierarchisch und inklusiv ist. Mit letzterem ist die Mehrfachzuordnung eines Tags zu einer Ressource gemeint. Analog dazu lassen exklusive Ordnungsysteme wie Taxonomien aufgrund ihrer monohierarchischen Struktur nur eine eindeutige Zuordnung zu.

„Proponents of collaborative tagging, typically in the weblogging community, often contrast tagging-based systems from taxonomies. While the latter are hierachical and exclusive, the former are non-hierarchical and inclusive.“ (Golder and Huberman 2006, S.1)

Nach der Definition Vander Wals ist eine Folksonomie der Regel nach der Öffentlichkeit zugänglich. (vgl. Wal 2004) Dies ist an sich aber nicht zwingend notwendig, schließlich steht das kollaborative Taggen im Vordergrund und nicht die Verfügbarkeit der Datensätze, wobei aber gerade eine Suche in der Gesamtheit aller Tags die Handlungsbereitschaft der Anwender, Tags zu verteilen, fördert.

3.2.1 Formale Definition

Hotho et al. definieren die grundlegende Struktur einer Folksonomie wie folgt (vgl. Hotho et al. 2006, S.4):

$$F := (U, T, R, Y), \qquad Y \subseteq U \times T \times R$$

U ist hierbei die Menger der User, T die Menger der Tags und R die Menge der Ressourcen. Y ist eine Menge an Tripeln, den Taginstanzen.

3.2.2 Broad/Narrow Folksonomies

„Generally, we can differentiate between two sorts of folksonomies with regard to 'tag scope' (Sen et al., 2006, 183): 1) folksonomies that allow for the multiple allocation of a tag to the same resource and 2) folksonomies that are only generated from the author's tags and may allow the adding of new tags by other users." (Peters 2009, S.164)

Im Folgenden werden beide Typen von Folksonomien anhand der Definition Vander Wals beschrieben (Wal 2005, vgl.):

1. *Broad Folksonomy*

   Eine Broad Folksonomy, also eine breite oder umfassende Folksonomie, wie Delicious eine ist, lässt die Mehrfachzuweisung des selben Tags zu einer Ressource zu. Dadurch lässt sich ein Trend in der Verteilung der Tags bezüglich einer Ressource erkennen, denn man kann anhand der Tag-Instanzen bestimmen, wie oft ein Tag jener Ressource zugeteilt wurde.

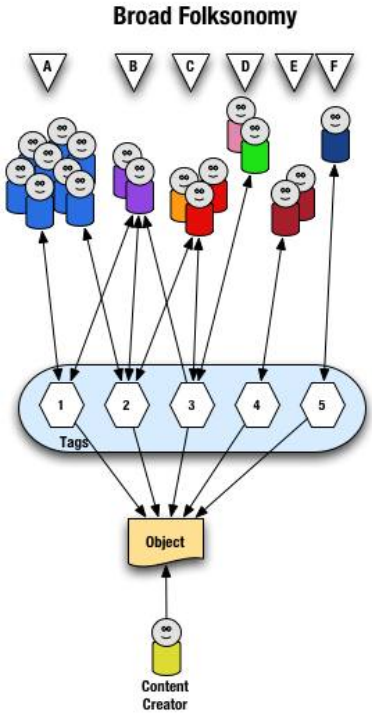

Abbildung 3.2: Broad Folksonomy[19]

Abbildung 3.2 zeigt eine Broad Folksonomy, bei der die Anwendergruppen *A,...,F* die Tags *1,...,5* einem Objekt zugeordnet haben. Ein Pfeil von einer Person/Gruppe zu einem Tag bedeutet, dass dieser Tag vom Anwender bzw. der Gruppe getaggt wurde, während ein Pfeil zurück darstellen soll, dass die Person/Gruppe das Objekt über diesen Deskriptor erhält. Gruppe *A* hat also Tag *1* und *2* dem Objekt hinzugefügt. Gruppe *B* ebenso, wobei letztere das Objekt über die Tags *1*, *2* und *3* suchen. Gruppe *E* und *F* nutzen nur einen einzigen Tag zur Suche, den sie jeweils selbst auch hinzugefügt haben. *E* nutzt Tag *4* und *F* nutzt Tag *5*. Man erkennt, dass Tag *1* von einem größeren Personenkreis getaggt wurde, als Tag *4* oder *5*. Die Suche über letztere ist daher zwar möglich, kann sich aber unter Umständen schwierig gestalten. Mehr dazu in Abschnitt 3.3 über Verteilungskurven von Tags.

2. *Narrow Folksonomy*

   Narrow Folksonomies wie Flickr hingegen lassen keine Mehrfachzuweisung des selben Tags zu. Häufig vergibt der Author der Ressource eine Grundmenge an Tags, welche dann von der Tagging-Gemeinschaft erweitert werden kann. Letzteres ist nicht immer der Fall, denn Anwendungen wie Youtube[20] z.B. erlauben nur die Tags des Authors bzw. des „Content Creators“, wobei genau genommen dadurch die Kompenente des kollaborativen Indexierens verloren geht und man nicht mehr von einer Folksonomie sprechen kann. (vgl. Peters 2009, S.165)

   Die Häufigkeitsverteilung der Tags bezüglich einer Ressource lässt sich in Narrow Folksonomies nicht bestimmen. „In Narrow Folksonomies, there is no possibility of counting tag frequency on a resource level and to observe distributions.“ (Peters 2009, S.165) Es lassen sich trotzdem Häufigkeitsverteilungen der Tags erstellen in Bezug auf die Gesamtheit aller vergebenen Tags. Man kann daran z.B. erkennen, wie wichtig Tags in Bezug auf das Gesamtsystem sind.

   Abbildung 3.3 zeigt eine Narrow Folksonomy, bei der die Anwendergruppe *A* das Objekt über den vom Content Creator erstellten Tag *1* erhält. Tag *2* wurde von Gruppe *B* und Tag *3* vom Anwender *F* erstellt. Da Anwender *F* nur über Tag *3* das Objekt erhält, bedeutet dies, dass er das Objekt zu Beginn nicht über die Suche erhalten hat, sondern z.B. über einen Link eines Freundes. Gruppe *E* kann das Objekt gar nicht finden, da es keine Tags bereitstellt, die mit der Suche übereinstimmen.

[19] Quelle: http://vanderwal.net/images/broadfolksonomy.jpg, Letzter Zugriff: 22. Feb. 2013
[20] http://www.youtube.com

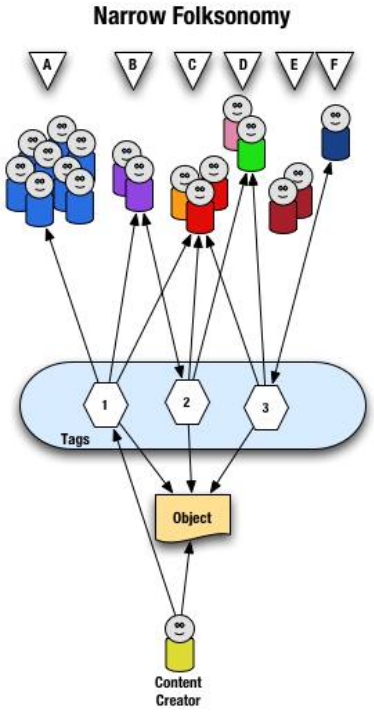

Abbildung 3.3: Narrow Folksonomy[21]

3.3 Häufigkeitsverteilung der Tags

Aus der Analyse von Tags und ihren Häufigkeitsverteilungen lassen sich verschiedene Informationen gewinnen, z.B. die am häufigsten genutzten Tags eines Nutzers, einer Gruppe von Nutzern oder der gesamten Nutzermenge. Es lassen sich Rückschlüsse auf das Taggingverhalten ziehen, sowie das Ranking der Tags bezüglich einer Ressource oder der gesamten Folksonomie bestimmen.

Die Betrachtung der Tagverteilung kann generell in Broad sowie Narrow Folksonomies erfolgen, wobei eine Analyse der ressourcenbezogenen Verteilungen in letzteren nicht möglich ist. Die meisten Tagverteilungen ergeben sich aus den Potenzgesetzen oder sind invers-logistischer Natur.

3.3.1 Power-Law Verteilung

Aus wissenschaftlichen Artikeln ist bekannt, dass die Häufigkeitsverteilungen i.d.R. den Gesetzmäßigkeiten des Potenzgesetzes (engl. power law) nach Lotkas Gesetz[22] folgen. (vgl. Peters 2009, S.171) Dieses Gesetz zeigt eigentlich die Beziehung zwischen der Anzahl von Publikationen einer Person und der Anzahl von Personen mit einem ebenso hohen

[21] Quelle: http://vanderwal.net/images/narrowfolksonomy.jpg, Letzter Zugriff: 22. Feb. 2013

[22] Lotkas Gesetz ist ein Skalengesetz, das in der Szientometrie, dem „Messen der Wissenschaft“, Anwendung findet.

Publikationsausstoß, lässt sich jedoch auf Tags übertragen. Abbildung 3.8 gibt solch eine Verteilung wieder. Eine beispielhafte Verteilung nach Lotkas Gesetz ist in Abbildung 3.4 dargestellt.

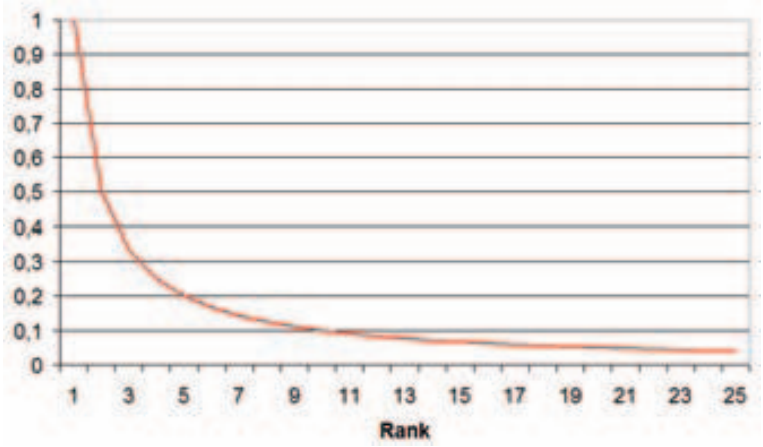

Abbildung 3.4: Relevanzverteilung nach Lotkas Gesetz[23]

Mathematisch betrachtet verhält sich die Verteilung nach folgender Formel

$$f(x) = C \cdot \left(\frac{1}{x}\right)^a = \frac{C}{x^a}$$

„bei der C eine Konstante, x der Rang des gegebenen Tags und a ein konstanter Wert (normalerweise zwischen 1 und 2) ist." (Peters and Stock 2008, S.79)

Daraus ergibt sich der Begriff „Long Tail", der die Tags bezeichnet, die nahezu die gleiche Häufigkeit aufweisen und sich am rechten Ende der Kurve befinden. Oft ist es der Fall, dass auf Ressourcenebene, Tags am Anfang der Kurve ein Objekt allgemeiner und adäquater beschreiben (Power Tags), während der Long Tail Deskriptoren zur spezielleren Beschreibung enthält. Im Fall einer Power Law-Verteilung können die ersten n Tags als Power Tags genutzt werden. Dabei ist n in Abhängigkeit zu dem Exponenten a zu wählen. Für $a = 1$ bietet sich beispielsweise $n = 4$, bei $a = 2$ etwa $n = 2$ an. (vgl. Peters and Stock 2008, S.79)

Es existieren in der Literatur mehrere Erklärungen dafür, dass sich Tagverteilungen nach genannten Gesetzen verhalten, wobei die meist verbreitete, den Yule-Prozess sowie den Yule-Simon Prozess als Erklärung heranziehen. „The underlying Yule process describes the generation of different biological taxa and is the most widespread model for the explanation of Power Law development." (Peters 2009, S.173)

Diese Prozesse besagen, dass an jeder Stelle innerhalb eines Textes ein Wort die Wahrscheinlichkeit p besitzt, ein neues Wort zu sein, insofern, dass es noch nicht vorher aufgetreten ist. Die Wahrscheinlichkeit *1-p* hingegen gibt an, dass das Wort eine Kopie eines bereits vorkommenden Wortes ist. Dieser Wert hängt also davon ab, wie oft ein Wort

[23]Quelle: (Peters and Stock 2008, S.79)

in einem Text schon vorgekommen ist, wobei angenommen wird, dass je öfter ein Wort schon aufgetreten ist, desto höher die Wahrscheinlichkeit ist, dass es erneut auftritt. (vgl. Peters 2009, S.173) Dieses Prinzip lässt sich ebenso auf Tags beziehen.

3.3.2 Invers-logistische Verteilung

Eine andere Art der Verteilung stellt die invers-logistische Verteilung dar, welche sich aus der Formel

$$f(x) = e^{-C'(x-1)^b}$$

ergibt, „bei der e die Euler'sche Zahl und x der Rang des Tags ist. C' [...] ist eine Konstante und der Exponent b ist stets ungefähr 3." (Peters and Stock 2008, S.80) In Abbildung 3.5 gilt ungefähr *C' = 0.1.*

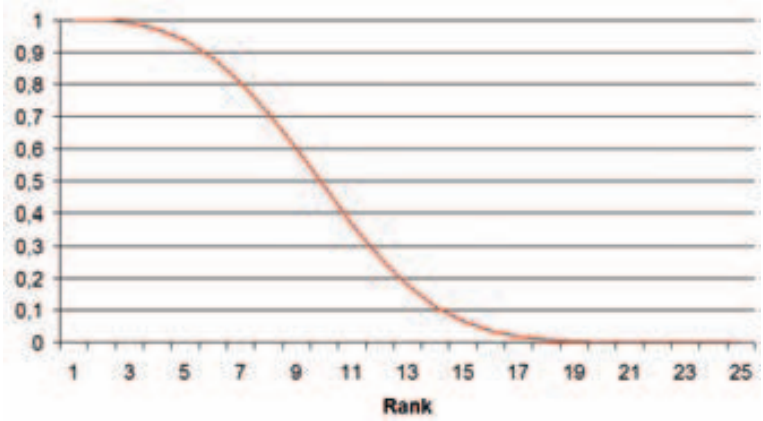

Abbildung 3.5: Invers-logistische Relevanzverteilung[24]

Im Gegensatz zur Power-Law Verteilung existiert hier nicht nur ein Long Tail, sondern auch ein Long Trunk (langer Rumpf), der den linken Teil der Verteilungskurve ergibt. Die Tags grenzen sich also in Bezug auf ihre Häufigkeiten nicht genug ab. An einer gewissen Stelle exisitert ein Wendepunkt, der den Long Trunk vom Long Tail abgrenzt. Im Fall der invers-logistischen Verteilung können alle Tags des Long Trunks als Power Tags dienen.

3.4 Tagclouds

Eine Tagcloud oder auch Schlagwortwolke ist eine graphische Darstellung der Tags und der Häufigkeit ihres Auftretens.

[24]Quelle: (Peters and Stock 2008, S.80)

3.4.1 Reguläre Tagclouds

Müller-Prove beschreibt eine Tag-Wolke wie folgt:

„Die Anzahl der vorhandenen Tag-Instanzen für jedes Tag kann leicht aufaddiert werden, so dass sich für jedes Tag ein Häufigkeitswert ergibt. Normiert man die Werte, um sie auf Zeichensatzgrößen umzurechnen und stellt die Label in einer alphabetisch fortlaufenden Liste dar, so gelangt man zu den so genannten TagWolken (Tag-Clouds)." (Müller-Prove 2008, S.17)

Eine Tagcloud ist also eine Liste an Tags, wobei die Darstellungsgröße der einzelnen Tags anhand ihrer Auftretenshäufigkeit ermittelt wird. Häufig auftretende Tags werden dabei größer dargestellt. Die grundsätzliche Ordnung der Tags innerhalb einer Tagcloud ist nicht festgelegt. Alphabetische, randomisierte bzw. geshufflete oder häufigkeitsrelevante Ausgaben sind denkbar.

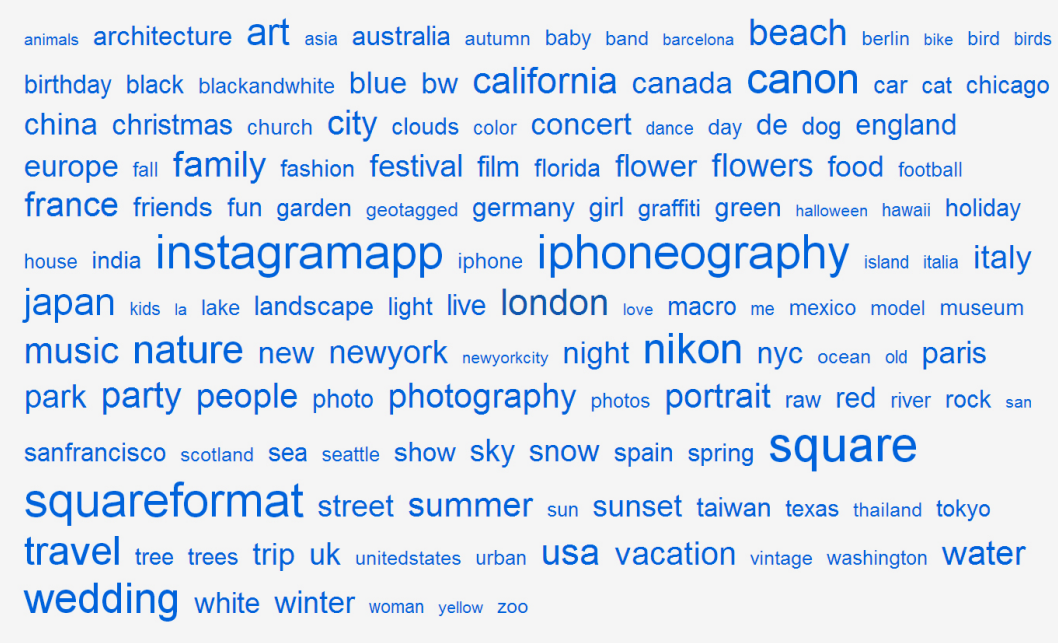

Abbildung 3.6: Tagcloud mit den beliebtesten Tags von Flickr[25]

Abbildung 3.6 zeigt eine einfache Tagcloud, welche die beliebtesten Tags von Flickr auflistet. Die Anwender haben also Tags wie *animals*, *water* und *iphoneography* vergeben, wobei zuletzt genannte eine größere Häufigkeit aufweisen und daher größer dargestellt werden. Man erkennt zusätzlich, dass die zu Grunde liegende Folksonomie Tags auf den lowercase (Kleinschreibung) reduziert oder lediglich die Darstellung demnach erfolgt. Die Auflistung der Tags ist alphabetisch sortiert.

Nimmt man die Zeit, zu der ein Tag gesetzt wurde, zu einer Tag-Instanz hinzu, so lässt sich das aktuelle Geschehen veranschaulichen, indem z.B. nur die fünfzig häufigsten Tags der letzten Woche präsentiert werden.

[25]Quelle: http://www.flickr.com/photos/tags, Letzter Zugriff: 10. Mar 2013

Tagclouds sind in Broad sowie Narrow Folksonomies umsetzbar, wobei in Broad Folksonomies weitere Möglichkeiten existieren, Wissen graphisch darzustellen. So sind u.a. Darstellungen auf Ressourcenebene realisierbar, also Tagclouds, die die Häufigkeiten der Tags eines bestimmten Objektes präsentieren.

Oft realisieren Tagclouds eine Navigation. Dies geschieht i.d.R. dadurch, dass, wenn man einen Tag aus der Tagcloud anklickt, dieser der Suche bzw. den Suchfiltern hinzugefügt wird.

3.4.2 Generische Tagclouds

Generische Tagclouds erweitern die Funktion einer einfachen Tagcloud um eine generische Komponente. So lassen sich Tagclouds implementieren, die abhängig von den bereits verwendeten Tags zur Suche nur weitere Tags präsentieren, die die Suche ereitern könnten. Der Inhalt generiert sich also anhand der Suchfilter.

Abbildung 3.7 zeigt zwei Tagclouds der Suchmaschine KLICKDRAUF[26], die nach genannten Prinzipien funktioniert, wobei die linke Seite die Tagcloud zu Beginn des Seitenbesuch und die rechte Seite die Tagcloud nach Auswahl von Tags ins Suchkriterium darstellt. Auf der linken Seite wurde also noch kein Tag ausgewählt, und die dargestellten Tags repräsentieren nur die häufigsten und neuesten Deskriptoren, um das aktuelle Geschehen aufzuzeigen.

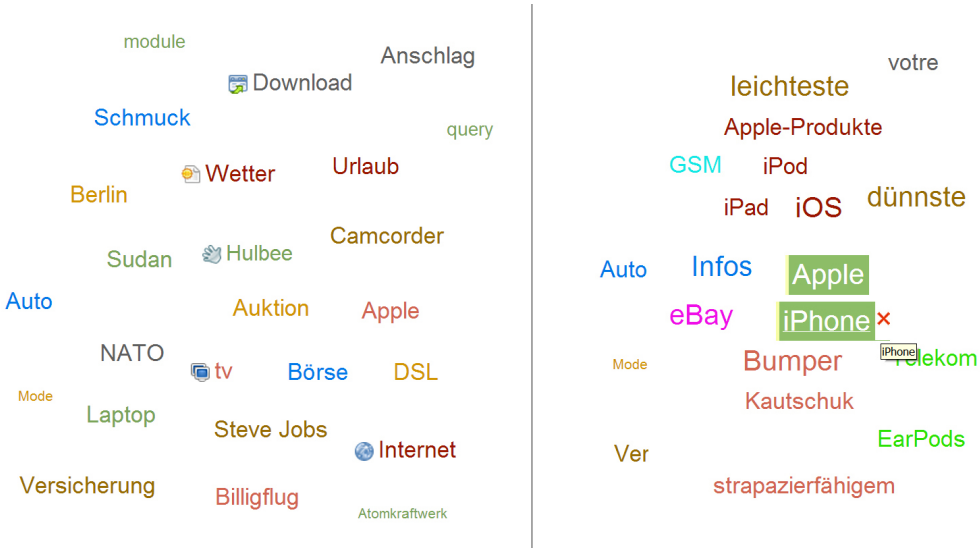

Abbildung 3.7: Generische Tagcloud vor und nach dem Hinzufügen von Tags zu den Suchfiltern

Die rechte Seite der Abbildung zeigt die Tagcloud, nachdem zunächst der Tag *Apple* und in einem zweiten Schritt der Tag *iPhone* angeklickt wurde. Die Tagcloud passt sich den gegeben Suchfiltern an und stellt nur Tags dar, die gemeinsam mit den bereits

[26] http://www.klickdrauf.de

ausgewählten Tags bestimmten Objekten zugehörig sind. Diese Objekte sind wiederum das Ergebnis der Suche. Eine Möglichkeit zur Rückwärtsnavigation ist ebenfalls gegeben, indem bereits gewählte Tags in der Tagcloud weiterhin angezeigt werden und man diese per Mausklick wieder aus den Suchfiltern entfernen kann.

Generische Tagclouds bieten den Vorteil, dass über sie eine sehr umfangreiche Navigation stattfinden kann. Im Vergleich zu einfachen Tagclouds kann eine Navigation durch mehrere Ebenen hinweg erfolgen, da sich die dargestellten Tags der aktuellen Suche anpassen können. Dies kann zu einer explorativen Suche führen.

3.4.3 Berechnung der Schriftgrößen

Da die Häufigkeitsverteilung der Tags i.d.R. dem Potenzgesetz folgt, sollte eine logarithmische Normierung vorgenommen werden, um eine gleichmässigere Darstellungsform zu erhalten.[27]

Auf Wikipedia findet man einen Ansatz zur Berechnung der Schriftgröße, zunächst mit linearer Normierung, den Tabelle 3.1 darstellt.

f_i:	anzuzeigende Schriftgröße
f_{max}:	maximale Schriftgröße
t_i:	Häufigkeit des Tags
t_{min}:	Häufigkeit, ab der ein Tag angezeigt werden soll
t_{max}:	Häufigkeit des häufigsten Tags

$$f_i = \left\lceil f_{max} \cdot \frac{t_i - t_{min}}{t_{max} - t_{min}} \right\rceil \qquad \text{, für } t_i > t_{min};$$

$$f_i = 1 \qquad \text{, sonst.}$$

Tabelle 3.1: Algorithmus zur Ermittlung der Schriftgröße eines Tags in einer Tagcloud mit linearer Normierung[28]

Die Schriftgröße des anzuzeigenden Tags ist also durch die Werte $f_{max}, t_i, t_{min}, t_{max}$ zu bestimmen. Diesen Algorithmus kann man durch eine logarithmische Normierung einfach erweitern. Tabelle 3.2 stellt den erweiterten Algorithmus dar.

Hierbei wurde ein Wert f_{min} hinzugefügt, der die minimale Schriftgröße angibt. Die Addition mit 2 ist nötig, da sonst in den beiden Extremfällen $t_i = T_{min}$ bzw. $t_i = T_{max}$ der Logarithmus oder die Division undefiniert ist.

Einen ähnlichen Algorithmus stellt Kentbye in einem Blog vor. Dieser besteht aus zwei Schritten. (vgl. kentbye 2005) Zunächst werden Schwellenwerte, ab denen Schriftgrößenänderungen vorgenommen werden sollen, logarithmisch berechnet. Abbildung 3.8 veranschaulicht die Schwellenwerte graphisch.

[27] vgl. Abschnitt 3.3

[28] Quelle: http://de.wikipedia.org/wiki/Schlagwortwolke, Letzter Zugriff: 20. Feb 2013

f_i: anzuzeigende Schriftgröße
f_{min}: minimale Schriftgröße
f_{max}: maximale Schriftgröße
t_i: Häufigkeit des Tags
t_{min}: Häufigkeit, ab der ein Tag angezeigt werden soll
t_{max}: Häufigkeit des häufigsten Tags

$$f_i = f_{min} + \left\lceil (f_{max} - f_{min}) \cdot \frac{\log(t_i - t_{min})}{\log(t_{max} - t_{min})} \right\rceil \quad \text{, für } t_i > t_{min};$$
$$f_i = f_{max} \quad \text{, für } t_i \geq t_{max};$$
$$f_i = f_{min} \quad \text{, sonst.}$$

Tabelle 3.2: Algorithmus zur Ermittlung der Schriftgröße eines Tags in einer Tagcloud mit logarithmischer Normierung

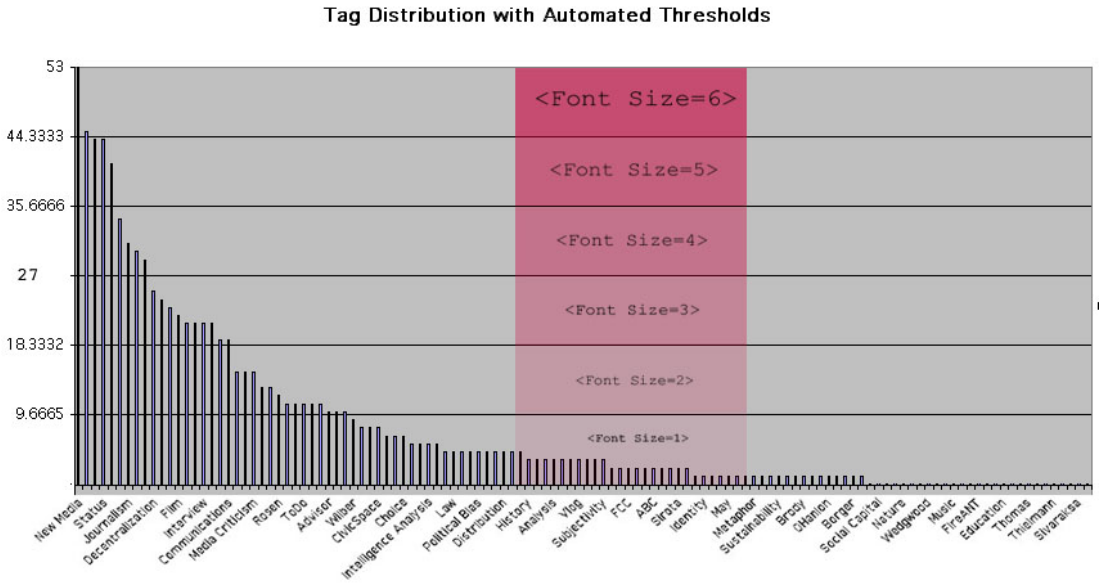

Abbildung 3.8: Tagverteilung nach dem Potenzgesetz mit Schwellenwerten[29]

In einem zweiten Schritt werden den Schwellenwerten Schriftgrößen zugeordnet und die Tags in die richtige Schwellenwertkategorie eingeordnet. Danach kann die Ausgabe erfolgen.

[29]Quelle: http://www.echochamberproject.com/files/images/autotagdist.jpg, Letzter Zugriff: 10. Mar 2013

3.5 Halbautomatische Indexierung/Vorschlagssysteme

Nicht nur in Folksonomien finden Systeme, die Vorschläge an Tags bereitstellen (recommendation systems), einen großen Anklang. Diese Systeme bieten viele Vorteile, da sie nicht nur die Taggingbereitschaft der Nutzer erhöhen, sondern auch eine Konsolidierung des Vokabulars ermöglichen: „Recommending tags can serve various purposes, such as: increasing the chances of getting a resource annotated, reminding a user what a resource is about and consolidating the vocabulary across the users." (Jäschke et al. 2007, S.1)

Oft werden die Tagvorschläge anhand von *auto-completition* Mechanismen, also Mechanismen zur Autovervollständigung, dem Tagging- oder Suchformular bereitgestellt.

Jäschke et al. beschreiben dabei formal ein Set an Tagvorschlägen aus einer Menge an Tags T zu einem Nutzer $u \in U$ und einer Ressource $r \in R$ als $T(u, r) \subseteq T$. (vgl. Jäschke et al. 2007, S.2)

Dabei existieren verschiedene Ansätze und Algorithmen zur Bestimmung der Tagvorschläge. Eine simple Methode ist, erst bei Eingabe eines Zeichens durch den Nutzer die ersten n Tags aus der Datenbank, die mit der bereits eingegebenen Zeichenkette übereinstimmen, vorzuschlagen. Dabei sind dann sowohl der Nutzer als auch die Ressource zur Berechnung der Vorschläge nicht von Bedeutung. Desweiteren existieren Varianten, die anhand der Ressource die Tagvorschläge bestimmen. Diese Systeme basieren i.d.R. auch auf einer Rankingpräferenz; so werden z.B. die n häufigsten bzw. populärsten Tags jener Ressource vorgeschlagen. „Schlägt ein System nämlich dem Indexer die jeweils bereits am häufigsten vergebenen Tags eines Dokuments vor und orientieren sich die indexierenden Nutzer tatsächlich daran, so entsteht – in einer Art self-fulfilling prophecy – stets eine Tag-Verteilung nach dem Power Law." (Peters and Stock 2008, S.84)

Beide Ansätze vereinfachen den Taggingprozess und fördern somit die Bereitschaft der Nutzer, Tags zu vergeben. Desweiteren können Rechtschreibfehler vermindert werden und eine Förderung von einheitlicherem Vokabular findet statt. Wenn man es jedoch genau nimmt, muss der Nutzer im Prinzip schon wissen, welche Deskriptoren er verteilen möchte. Er stößt somit nicht auf neue Tags, die beschreibend sein könnten.

Einen komplexeren Ansatz bieten Jäschke et al., die in ihrem Papier über Vorschlagssysteme in Folksonomien zwei Algorithmen präsentieren und evaluieren. (vgl. Jäschke et al. 2007) Ein Algorithmus basiert dabei auf einem nutzerbasierten Ansatz des *collaborative filterings*[30], also dem kollaborativen Filtern, der andere auf einer graphenbasierten Variante aufbauend auf dem *FolkRank* Algorithmus, der eine Erweiterung des *PageRank*[31] Algorithmus darstellt. Abbildung 3.9 zeigt eine Kurve, die die Wiederaufrufe von Seiten mit der Anzahl der vorhandenen vorgeschlagenen Tags in Bezug auf die verschiedenen Algorithmen relativiert.

[30] „Collaborative filtering" bezeichnet den Prozess der Informationsfilterung unter Einbeziehung mehrerer kollaborativ arbeitender Agenten, Datenquellen etc.

[31] Der PageRank Algorithmus wurde Google Inc. Gründern Larry Page und Sergei Brin entwickelt und dient der Bewertung bzw. Gewichtung einer Menge verlinkter Dokumente (z.B. dem World Wide Web). Dabei wird jedem Element anhand der Verlinkungsstruktur ein Gewicht (PageRank) zugeordnet. Der PageRank Algorithmus dient der Suchmaschine Google als Grundlage für die Bewertung von Webseiten.

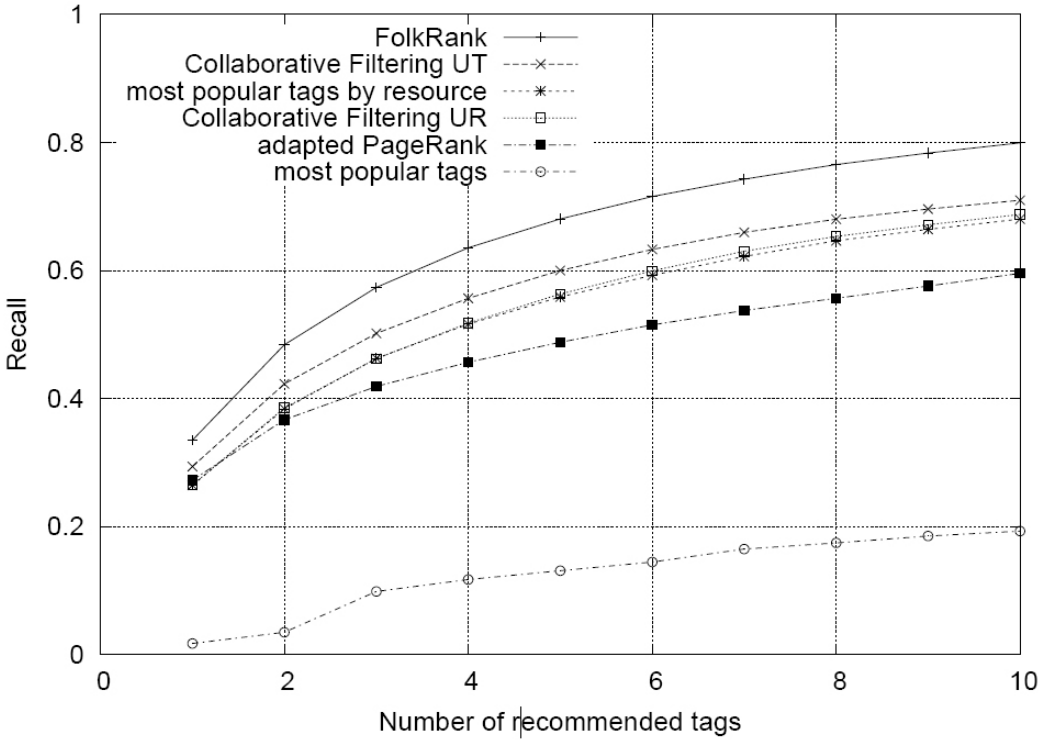

Abbildung 3.9: Wiederaufrufe von Seiten im Vergleich zur Anzahl vorgeschlagener Tags[32]

Man erkennt, dass der *FolkRank* Algorithmus die besten Ergebnisse erzielt. Dieser Algorithmus funktioniert nach dem Prinzip, dass eine Ressource, die mit wichtigen Tags von wichtigen Nutzern verknüpft ist, automatisch selbst wichtig wird. Dasselbe Prinzip gilt wiederum für Nutzer und Tags. Für den sogenannten *cold start* bzw. Kaltstart einer Anwendung ist jedoch ein System, basierend auf den häufigsten Tags, zunächst gut geeignet. „The adapted PageRank profits also from this good performance of the 'most popular tags' on small datasets." (Jäschke et al. 2007, S.6)

3.6 Ranking

Die Relevanz einer Ressource in den Ergbnissen einer Suche kann, wie in Abschnitt 3.5 erwähnt wurde, z.B. durch den *FolkRank* Algorithmus geschehen, der eine Anpassung des *PageRank* Algorithmus auf folksonomietypische Strukturen ist. (vgl. Hotho et al. 2006) Dabei werden Graphalgorithmen auf die in einen ungerichteten, gewichteten, tripartiten Graph umgeformte Folksonomie angewendet, um eine Relevanz der Ressourcen zu erhalten.

[32]Quelle:(Jäschke et al. 2007, S.6)

Im Allgemeinen sind viele Algorithmen zur Bestimmung des Rankings nach Relevanz denkbar. „del.icio.us ranks reverse-chronologially according to the date of the bookmarks entry into the system and the tags according to popularity“ (Peters 2009, S.339).

3.7 Folksonomie und Ontologien

Da eine Ontologie aufgrund der relationalen Struktur mehr Ausdrucksstärke besitzt als eine Folksonomie, existieren verschiedene Ansätze, die kollaborativ erstellten Metadaten aus einer Folksonomie in eine Ontologie zu überführen, welche anschließend präzisiert werden kann. Dies ist nicht Gegenstand dieser Arbeit, aus informativen Gründen werden trotzdem einige Aspekte erläutert.

1. Überführung

   Van Damme et al erläutern einen Ansatz zur Überführung einer Folksonomie in eine Ontologie und Validierung der extrahierten Daten durch die Gemeinschaft. (vgl. Damme et al. 2007) Dabei werden zunächst die Tags gesäubert. Dies beinhaltet die Reduktion von Wörtern auf ihre Grundform. Manche User haben schließlich die Pluralform eines Nomens getaggt, andere den Singular. Die Reduktion geschieht mittels sogenannter Stemming Algorithmen bzw. Algorithmen zur Stammformreduktion. Unter diesen Begriffen fasst man im Information Retrieval und der linguistischen Informatik Verfahren zusammen, mit denen verschiedene morphologische Varianten eines Wortes auf ihren Wortstamm zurückgeführt werden können.[33]

   „It is important not to loose the context of the tags, therefore the stemming process of tags should be limited to plural nouns and conjugated verbs.“ (Damme et al. 2007, S.7)

   Anschließend wird über die lexikalischen Ressourcen Leo Dictionary[34], Wordnet[35], Google[36] und Wikipedia[37] die Rechtschreibung der einzelnen Tags überprüft. Werden keine Ergebnisse gefunden, wird die Häufigkeit des Auftretens innerhalb des Systems ermittelt. Bei einem hohen Auftreten lässt sich davon ausgehen, dass ein neues Wort innerhalb der Tagging-Gemeinschaft entstanden ist. Bei einem geringen Auftreten wird das Wort wohl falsch geschrieben worden sein und kann eventuell noch mit der richtigen Form zusammengeführt werden, ansonsten wird der Tag verworfen.

   Im nächsten Schritt werden diverse Algorithmen angewendet, um Tag-Paare, hierarchische Relationen sowie die Tagging-Gemeinschaft ansich zu analysieren. Die gewonnenen Informationen werden anhand oben genannter lexikalischer Ressourcen

[33] Ein bekannter Stemming Algorithmus ist der Porter-Stemmer-Algorithmus, bei dem eine Menge an Verkürzungsregeln auf ein Ausgangswort angewendet wird, bis dieses eine Minimalanzahl von Silben enthält.
[34] http://dict.leo.org
[35] http://wordnet.princeton.edu
[36] http://www.google.com
[37] http://www.wikipedia.org

dann mit zusätzlichen Informationen bereichert und mittels einer Ontologiesprache ausgedrückt.

2. Modellierung

   Ein anderen Ansatz bieten Echarte et al, die eine Methode vorstellen, Folksonomies mittels Ontologien zu modellieren. „This method consists of: (1) an ontology able to be used to represent any kind of folksonomy, and (2) an algorithm to transform folksonomies into the proposed ontology and to update the resulting ontology as the folksonomy evolves in time.“ (Echarte et al. 2007, S.8)

   Hierbei wird die Struktur einer Folksonomie in einer Ontologie abgebildet, wobei diese um bestimmte Relationen erweitert wird. Dabei werden Probleme von Folksonomien, wie die unterschiedlichen Schreibweisen von Deskriptoren und die schlechte Unterscheidbarkeit von persönlichen und allgemeinen Tags, gelöst. Jedoch wird dazu wiederum zusätzliches Wissen benötigt, welches erst akquiriert werden muss, bevor es in Relationen abgebildet werden kann.

3.8 Ein Vergleich mit anderen Ordnungssystemen

Die Erstellung einer Folksonomie erfolgt durch den Nutzer selbst. Daraus und aus der inklusiven Mehrfachzuweisung von Tags ergibt sich eine semantische Bedeutungsvielfalt, woraus bestimmte Vor- und Nachteile in Bezug auf die in Abschnitt 2.2 vorgestellten Ordnungssysteme entstehen.

„The greatest strength of folksonomies, their linguistic ans semantic variety, is also their greatest weakness“ (Peters 2009, S.218)

Grundsätzlich lässt sich sagen, dass Folksonomies von ihrer Nutzerzahl abhängig sind, „the more users tag, the better it is for the system; if the number of tagging users breaches 'critical mass,' the system will lift off and establish itself as a standard.“ (Peters 2009, S.216)

Es muss noch erwähnt werden, dass sich die einzelnen Ordnungssysteme nicht gegenseitig ausschließen. Hybridartige Modelle sind denkbar und konkurrenzartige Vergleiche von z.B. Ontologien und Folksonomien sind nicht ganz zutreffend.

3.8.1 Vorteile

Folksonomies binden den Anwender an den Prozess der Indexierung und können dadurch dem Nutzer den Sinn, Zweck und die Probleme der Verschlagwortung näher bringen. Sie bieten Vorteile nicht nur in Bezug auf die Wissensdarstellung, sondern auch in dem Erhalt von Informationen, dem Information Retrieval.

1. *Nutzerfreundlichkeit*

   „One of folksonomies‘ advantages is often identified as the fact that they are easy to use.“ (Peters 2009, S.161)

Die Verwendung von Folksonomien gestaltet sich i.d.R. ziemlich einfach. Beim Tagging schreibt man einfach die Begriffe, die man mit dem zu taggenden Objekt verbindet herunter. Die kognitiven Modelle, die sich bei der Betrachtung eines Objekts bilden, können also direkt in Tags umgewandelt werden. Nach Sinha besteht ein großer Unterschied zu hierarchischen Ordnungssystemen oder Ordnungssystemen mit festen Vokabular darin, dass im kognitiven Prozess eines Anwenders die nachträgliche Analyse und Paralyse der möglichen Beschreibungkonzepte nicht notwendig ist. (vgl. Sinha 2006) Abbildung 3.10 und 3.11 veranschaulichen diesen Prozess in Bezug auf Tagging und Kategorisierung:

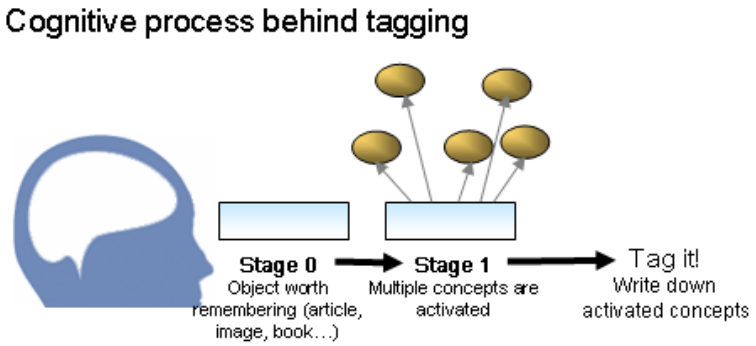

Abbildung 3.10: Kognitiver Vorgang des Taggens[38]

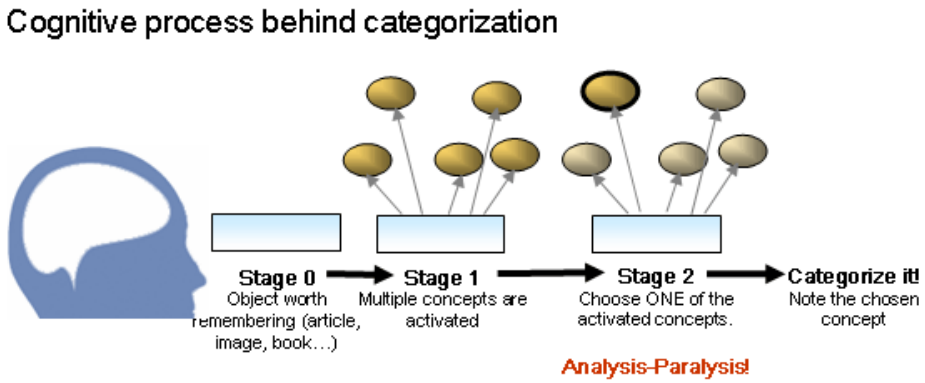

Abbildung 3.11: Kognitiver Vorgang des Kategorisierens[39]

2. *Umgang mit großen Datenbeständen*

   Im Vergleich zur traditionellen Indexierung, bei der Experten die Verschlagwortung vornehmen, erfüllen die Anwender einer Folksonomie nicht zwangsläufig die Anforderungen, eben genannte kognitive Modelle oder auch Möglichkeiten der Beschreibungen zu ordnen und zusammenzuführen. Nicht jeder Nutzer ist ein Experte. Das muss er jedoch auch nicht sein, denn der Tagging Prozess in Folksonomies

[38] Quelle: http://www.flickr.com/photos/riddle/57209550/
[39] Quelle: http://tagging.files.wordpress.com/2007/04/cognitive_categorization.gif

soll ja gerade das freie Verschlagworten förden. Die Gänze der Tags stellt dann die eindeutige Beschreibung des Objekts dar, welche in einer Taxonomie z.B. durch die Notation bestimmt ist. Das bedeutet, dass Folksonomies aufgrund ihrer Möglichkeit, eine große Tagging-Gemeinschaft zu etablieren, eine viel größere Menge an Datenbeständen verschlagworten kann, als Ordnungssysteme, die auf Experten zurückgreifen. Besonders im World Wide Web, wo die Datenmengen stetig wachsen, kann daher eine Folksonomie einen Ansatz darstellen, jene Daten zu ordnen.

3. *Durchsuchbarkeit*

   Die freie Indexierung sowie die Mehrfachzuweisung von Tags, ermöglichen eine effektive und vor allem breitgefächerte Suche, die auch unter Laienbegriffen[40] Erfolg haben kann. In den meisten Anwendungen findet eine Kombination aus einer Suche in den Tag-Instanzen, also der Folksonomie, und der Suche in den bereits gegebenen Datenobjekten und ihren Metadaten statt. Letzteres würde z.B. das Erstellen eines Volltextindexes beinhalten, der Inhalte von Objekten wie PDF-Dokumenten oder Webseiten für Volltextsuchmaschinen indexiert. Auch Metadaten der Objekte, wie Größe und Format eines Bildes und dessen Dateityp und Autor, könnten dabei indexiert werden. Einem Objekt sind dann eine Menge an Tags sowie Wörter aus dem Volltextindex zugehörig. „D.h. die tagbasierte Suche könnte mit anderen Worten auch als Filtern bezeichnet werden, wobei aus allen vorhandenen Elementen diejenigen mit den zutreffenden Tags herausgefiltert werden." (Frohner 2009, S.27)

   Wie sehr die Tags den Themenbereich und das Objekt beschreiben, hängt letztlich von dem Personenkreis ab, der Zugang zu den Objekten und ein Interesse daran hat, diese zu indexieren. Daraus resultierend vermag die Suche in einer Folksonomie gegebenenfalls breit gefächert sein, trotzdem haben hierarchische Ordnungssysteme durchaus ihre Vorzüge. Kennt man sich in einer Hierarchie gut aus, so ergibt sich aus dem festen Platz eines Objektes innerhalb dieser Struktur eine sehr effiziente Suche.

   In einer gut funktionierenden Collaborative Tagging Anwendung wie Delicious sind zu den wichtigen Objekten i.d.R. alle Arten von Tags[41] vertreten. Da das Tagging auch von Laien erfolgt, ist über Begriffe, die ein Experte eher nicht verwenden würde, trotzdem eine Suche möglich.

   „Since the users of collaborative information services become indexers themselves, and attach their thoughts, association and descriptions to the information resource in their own language, via tags, the tags then directly reflect the users' wishes regarding the descriptions." (Peters 2009, S.214)

   Dies ist ein großer Vorteil gegenüber auf Expertenwissen basierenden Ordnungssystemen. Das Ganze ist aber gleichzeitig mit einer Verminderung der Präzision von Suchergebnissen verbunden. Dazu mehr in Abschnitt 3.8.2.

[40] Hier: Begriffe unter denen potentielle Sucher, ohne fachbezogenes Wissen, suchen würden.
[41] Siehe Abschnitt 2.1.2, um eine Übersicht der verschieden Arten von Tags zu erhalten.

Eine weitere Möglichkeit der Suche bieten die in Abschnitt 3.4 erläuterten Tagclouds. Besonders die generischen Tagclouds können das aktuelle Geschehen präsentieren sowie eine Navigation anhand der Tags ermöglichen, was wiederum eine explorative Suche fördert.

4. *Flexibilität und Aktualität*

   Da der Taggingprozess fortwährend stattfindet, kann eine Folksonomie sehr schnell auf Veränderungen reagieren, indem einer Ressource neue Deskriptoren hinzugefügt werden. Aus der Flexibilität folgt die Aktualität, unter der Voraussetzung, dass eine Anwendung über genug aktiv taggende Nutzer verfügt. Dies kann auch zur Übersichtlichkeit beitragen. So lässt sich das aktuelle Geschehen, z.B. neu getaggte Dokumente oder kürzlich gesetzte Tags, unter anderem über eine Listenansicht oder eine Tagcloud[42] präsentieren.

5. *Nutzerperspektive*

   Über die Tag-Instanzen ist ein Anwender aus Systemebene eng an Tags und Ressourcen gebunden. Jene Relationen bieten Möglichkeiten Informationen aus dem Nutzerverhalten zu generieren, die zu Verbesserungen der Folksonomie und Statistikzwecken genutzt werden können.

3.8.2 Nachteile

1. *Synonyme und Homonyme*

   Zur Erinnerung: Bei Synonymen werden verschiedene sprachliche oder lexikalische Bezeichnungen für den selben Begriff verwendet.[43]

   „Homonym (gleichnamig) heißen Dinge, die nur den Namen gemein haben, während der zum Namen gehörende Wesensbegriff verschieden ist.“ (Aristoteles 1995, Kategorien 1, 1a)

   Homonyme sind also das Gegenstück zu Synonymen und stellen Wörter dar, die gleich geschrieben werden, aber eine unterschiedliche Bedeutung haben. So z.B. „Ball“, welches einerseits den kugelförmigen Ball eines Spiels wie Fußball bezeichnen kann oder aber auch die Festlichkeit, wie einen Tanzball.

   Solche Begriffe lassen sich in einer Folksonomie grundsätzlich semantisch nicht auseinanderhalten, da im Vergleich mit Ordnungssystemen wie Ontologien oder Thesauri keine Relationen zwischen den Deskriptoren existieren. Es gibt jedoch NLP[44]-Ansätze, welche zur Erkennung von Homonymen/Synonymen und weiteren Relationen angewendet werden können. „Während der Erkennung von Homonymen und Synonymen muss man Wissensordnungen wie beispielsweise WordNet (Miller,

[42] Siehe Abschnitt 3.4.
[43] vgl. Abschnitt 2.2.1
[44] Natural Language Processing (NLP) ist ein Fachgebiet der Informationswissenschaft, Informatik und Computerlinguistik und behandelt die Interaktionen zwischen Computern und menschlicher natürlicher Sprachen.

1998) zur Hilfe nehmen. Außerdem könnte es hilfreich sein, bei der Homonymunterscheidung auch Co-Occurrence-Statistiken der Tags einzubeziehen (Butterfield et al., 2006)“ (Peters and Stock 2008, S.84)

2. *Präzision* Die Vermischung von verschiedenen Sprachen, Spamtags und eben genannte Probleme mit Homonymen und Synonymen führen zu einer Verschlechterung der Präzision der Suchergebnisse und der Folksonomie im Allgemeinen. So kann es vorkommen, dass falsche Ergebnisse angezeigt werden.

3. *Fehlen eines kontrollierten Vokabulars* Die fehlende einheitliche Vorschrift bei der Verschlagwortung ist einer der größten Vorteile, aber auch Nachteile von Folksonomien.

4. *Vermischung der Arten von Tags* Die in Abschnitt 2.1.2 erwähnten Arten von Tags lassen sich nicht grundsätzlich voneinander auseinanderhalten.

5. *Fehlende Hierarchie und fehlende Relationen* Die fehlende hierarchische Struktur bzw. die nicht-relationale Struktur von Folksonomien erschweren eine semantische Interpretation der Daten.

3.8.3 Gegenüberstellung

Im Folgenden werden nochmal knapp einige Vor- und Nachteile tabellarisch gegenübergestellt:

Vorteile	**Nachteile**
• Dursuchbarkeit mittels breitem Vokabular • Spiegeln das Vokabular der Anwender wieder • Nutzerfreundlichkeit und einfache Benutzung • Umgang mit großen Datenbeständen • Flexibilität und Aktualität durch fortwährende Evolution • Tagclouds	• Mangelnde Präzision durch Homonyme, Synonyme, Vermischung der Sprachen, Rechtschreibfehlern, Spam-Tags, etc. • Fehlen eines kontrollierten Vokabulars • Vermischung der Arten von Tags • Fehlende Hierarchie und fehlende Relationen zur semantischen Interpretation

Tabelle 3.3: Gegenüberstellung der Vor- und Nachteile von Folksonomien im Vergleich mit anderen Ordnungssystemen

4 URURI - Ein Konzept zur kollaborativen Indexierung von URIs

Im folgenden wird ein Prototyp einer Social Tagging Anwendung zur kollaborativen Indexierung von Inhalten vorgestellt. Dabei wurden Konzepte wie Tagclouds und Vorschlagssysteme umgesetzt.

4.1 Was sind URIs?

Damit die Datenobjekte möglichst inhaltsunabhängig referenziert werden können, wird eine URI zur Identifizierung verwendet. Daher auch der Anwendungsname „URURI“, was für „your uri“ stehen soll. URIs sind in RFC 1630 und RFC 3986 beschrieben:

„A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource.“ (Berners-Lee 2005, in Abstract)

Ein Uniform Resource Identifier hat dabei folgende syntaktische Komponenten (vgl. Berners-Lee 2005, S.15):

URI = scheme „:“ hier-part [„?“ query] [„#“ fragment]

hier-part = „//“ authority path-abempty
/ path-absolute
/ path-rootless
/ path-empty

Die Scheme und Path Komponenten sind notwendig, auch wenn der Pfad leer ist, also keine Zeichen enthält. Der *hier-part* steht für eine optionale Autorität und den Pfad. Falls eine Autorität vorhanden ist, beginnt der hier-part mit //, und der Pfad muss mit einem / beginnen, anderfalls darf der *hier-part* nicht mit // beginnen Berners-Lee verdeutlicht die Komponenten an einem Beispiel (vgl. Berners-Lee 2005, S.15):

foo://example.com:8042/over/there?name=ferret#nose

Hierbei stellt *foo* das Schema, *example.com:8042* die Authorität, */over/there* den Pfad, *name=ferret* die Anfrage und *nose* das Fragment dar. Dass eine Authorität vorhanden ist, erkennt man daran, dass der hier-part mit // beginnt. Das RFC beschreibt weiterhin die jeweiligen zulässigen Zeichenketten, Normalisierungen und Sicherheitsbedenken, auf die hier nicht weiter eingangen wird.

Uniform Resource Identifier bieten einem also die Möglichkeit, verschiedene Ressourcen inhaltsunabhängig zu referenzieren. URIs bestehen aus zwei Unterarten, den *Uniform Resource Locators* (URL) und den *Uniform Resource names* (URN) in welche ursprünglich Schemata wie *ftp* (URL), *isbn* (URN) oder *mailto* eingeordnet werden sollten. Schemata wie letzteres lassen sich jedoch nicht genau in eine der beiden Kategorien einordnen, sie befinden sich eher in der Schnittmenge von URIs und URNs, womit eine strenge Aufteilung aufgegeben wurde. Beispiele für URIs sind:

1. Webseiten (http, https)
   http://tools.ietf.org/html/rfc3986
2. FTP-Server
   ftp://ftp.rz.uni-kiel.de/pub/
3. E-Mail Adressen
   mailto:john.doe@gmail.com
4. Bücher über ISBN-Nummern
   isbn:3598251795
5. Das eigene Dateisystem
   file:///C:/folder/file.ext
6. Geodaten
   geo:42.359616,-71.09377
7. Git-Repositories
   git://github.com/rails/rails.git

Eine Erweiterung der nur aus ASCII-Zeichen bestehenden URIs stellen die *Internationalized Resource Identifiers* (IRIs) dar.

4.2 Anforderungen an URURI

URURI soll die grundlegenden Aspekte einer Folksonomie umsetzen. Dazu gehören die Möglichkeiten, öffentliche Daten zu referenzieren und diese durch eine Gemeinschaft indexieren zu lassen. Die vergebenen Tags sollen dabei graphisch durch eine Tagcloud visualisiert werden. Weiterhin sollen Möglichkeiten implementiert werden, die den in Abschnitt 3.8.2 erläuterten strukturbedingten Nachteilen entgegenwirken. So kann eine halbautomatische Indexierung mittels Vorschlagssystemen (engl. recommendation systems) das Vokabular in Ansätzen kontrollierbarer machen.

Die von Jakob Voß erörterte Typologie wird im folgenden auf URURI angewendet, um sich über grobe Eigenschaften bewusst zu werden (vgl. Voß 2007, S.6):

- *Tagging Rights*
  Jeder Nutzer soll Tags vergeben können, wobei Restriktionen in Bezug auf eine zu schnelle Vergabe oder falsche Vergabe zur Beeinflussung der Rankings nötig sind

(IP-Sperre). Zusätzlich sollte es zu einem späteren Zeitpunkt die Möglichkeit geben, eingeschränkte Bereiche zu kreieren, wo es z.B. nur einer bestimmten Nutzergruppe erlaubt ist zu taggen, um Expertengemeinschaften zu etablieren. Auch denkbar ist, dass zwar grundsätzlich jeder taggen darf, aber eine Unterscheidung in Experten bzw. Gruppenmitgliedern mit jeweiligen Rechten und normalen Nutzern getroffen wird.

- *Source of Resources*
  Vorerst wird über URIs referenziert, um eine große Vielfalt an Inhaltstypen referenzieren zu können. Später ist denkbar, auch selbst Hoster von Ressourcen zu werden. Eine Möglichkeit, etwas abseits der Folksonomie, die Ordnerstruktur von Teilen des eigenen Dateisystems auf URURI zu speichern, könnte dem Nutzer zudem eine Motivation geben, die Anwendung vermehrt zu nutzen. Diese Daten entsprechen dann zwar nicht mehr den kollaborativen Daten, können jedoch trotzdem der Folksonomie an sich nützlich sein.

- *Resource Representation*
  Für die Präsentation der jeweiligen Datentypen sollen, soweit möglich, einzelne Darstellungen implementiert werden.

- *Tagging Feedback*
  Dem Nutzer sollen die Tags anderer Nutzer präsentiert werden, wobei eine Liste der am häufigsten vergebenen Tags ausreichend sein sollte. Das Tagging selbst soll mittels Ajax[45] umgesetzt werden, um z.B. während der Präsentation eines Videos die Seite nicht neu laden zu müssen. Dabei sollen dem Nutzer anhand der schon im System vorliegenden Tags mögliche Deskriptoren vorgeschlagen werden.

- *Tag Aggregation*
  URURI soll eine Broad Folksonomy sein, also die Mehrfachvergabe von Tags zu einer Ressource erlauben.

- *Vocabulary Control*
  Zunächst soll das Vokabular gänzlich frei wählbar sein, bis darauf ein Vorschlagssystem anhand der schon vergeben Tags implementiert werden soll. Zudem werden die n wichtigsten Tags zu einer Ressource ausgegeben.

- *Vocabulary Connectivity*
  Vorerst soll keine Verknüpfung der Tags untereinander vorgenommen werden. Wenn die Folksonomie wächst, sollten Möglichkeiten zur Entfernung bzw Reduzierung falsch geschriebener Tags auf die richtigen Begriffe in Betracht gezogen werden. Ebenso wäre eine Beziehung für Homonyme, Synonyme und Klassifikationen interessant, da dadurch die Suche noch erweitert werden könnte.

- *Resource Connectivity*

[45] Ajax steht für „Asynchronous JavaScript and XML“ und stellt ein Konzept der asynchronen Datenübertragung zwischen Browser und Server dar.

Die Ressourcen bzw. die URIs, die die Ressource referenzieren, sollen in Kategorien, so genannten Wissensdomänen eingeordnet werden.

- *Automatic Tagging*
  Eine direkte automatische Zuweisung von Tags muss nicht erfolgen; jedoch sollte die Suche anhand von weiteren Metadaten, die der Autor erstellt, erweitert werden. Zukünftig ist denkbar, dass aus den Inhalten selbst weitere Deskriptoren extrahiert werden.

4.3 Grundlegendes zur Anwendungsaufbau

Bisher habe ich Webanwendungen stets in PHP geschrieben, da es einerseits einfach ist, schnell dynamischen Inhalt zu generieren, und andererseits die Verbreitung der Sprache relativ hoch ist und daher nahezu alle Webspaceprodukte PHP unterstützen. In Artikeln liest man jedoch immer wieder von Problemen jeglicher Art, wobei manche PHP sogar als „broken by design“[46] betiteln. Ohne hier näher darauf einzugehen, habe ich dies als Anlass dazu genommen, URURI in Ruby bzw. Ruby on Rails zu implementieren.

Ruby on Rails[47] (auch Rails oder RoR) ist ein in der Programmiersprache Ruby geschriebenes Web Application Framework, mit dem sich Webapplikationen nach MVC-Prinzipien[48] erstellen lassen. Rails ist open-source und verhält sich nach der Präferenz "convention over configuration", welches u.a. besagt, dass statt einer variablen Konfiguration, Konventionen für die Namensgebung von Objekten einzuhalten sind.

Da für die Suche der in Abschnitt 4.6 erläuterte Open Source Search Server Sphinx[49] verwendet wird, sollte dieser auf dem System vorhanden sein. In dieser Arbeit wurde mit dem „2.0.6-release“ aus dem Oktober 2012 gearbeitet.

URURI nutzt diverse RubyGems[50], über die u.a. Programmfunktionalitäten eingebunden werden. Einige dieser sind im Nachfolgenden aufgeführt. Eine Übersicht aller Gems ist im Gemfile einzusehen.

- *rails*
  Als Framework, hier wurde Version 3.2.11 verwendet.
- *mysql2*[51]
  Als grundlegende Datenbank.
- *jquery-rails*[52]

[46] vgl. http://helmbold.de/it/php/, Letzter Zugriff: 5. Mar 2013
[47] http://rubyonrails.org/
[48] MVC steht für model-view-controller (deutsch: Modell-Präsentation-Steuerung) und stellt ein Muster zur Strukturierung von Software dar, welches die Applikationsstruktur in die drei Module Datenmodell, Präsentation und Programmsteuerung einteilt.
[49] http://sphinxsearch.com
[50] RubyGems oder auch Gems ist das offizielle Paketsystem für die Programmiersprache Ruby.
[51] http://www.mysql.com
[52] https://github.com/rails/jquery-rails

Um jQuery[53] nutzen zu können.

- *thinking-sphinx*[54]
  Zur Kommunikation mit dem Sphinx Search Server, hier wurde Version 3.0.0 verwendet.
- *devise*[55]
  Zur Authentifizierung von Nutzern.
- *cancan*[56]
  Zur Autorisierung von Nutzern.
- *twitter-bootstrap-rails*[57]
  Zur graphischen Darstellung der Applikation.
- *will_paginate*[58]
  Um Seitenumbrüche in der Darstellung von Daten zu ermöglichen.
- *google_books*[59]
  Um Anfragen an die Google Book API übersichtlich durchführen zu können.

Die Darstellung der Anwendung wurde mit von dem von Twitter hergestellten Toolkit *Bootstrap* aus angepasst. Dieses enthält u.a. *Cascading Style Sheets* (CSS), Stilvorlagen zu strukturierten Dokumenten wie HTML oder XML. Anpassungen und eigene CSS Definitionen sind in *app/assets/stylesheets/* zu finden. Hierbei wurde oft LESS[60] zur Beschreibung verwendet.

4.4 Die Modelle

URURI wurde nach MVC-Prinzipien konzipiert, um die Klarheit von Code sowie die Isolation von Anwendungslogik und Darstellung zu fördern. Die Modelle, Präsentationen und Steuerungen befinden sich nach Rails Struktur in den Ordnern *app/model/*, *app/view/* und *app/controller/*.

In Rails korrespondiert in den meisten Fällen ein Modell mit einer Tabelle in der Datenbank, wobei jedes Modell ein zusätzliches Feld `id` erhält, welches den Primärindex darstellt. Das Modell hält dazu Methoden zur Manipulation der Daten. Die relationale Struktur der Modelle zueinander wird in Rails durch Assoziationen wie `has_one`,

[53] jQuery ist eine freie JavaScript-Bibliothek, welche Funktionen zur Manipulation des Document Object Models (DOM), einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente, zur Verfügung stellt.
[54] https://github.com/pat/thinking-sphinx
[55] https://github.com/plataformatec/devise
[56] https://github.com/ryanb/cancan
[57] https://github.com/seyhunak/twitter-bootstrap-rails
[58] https://github.com/mislav/will_paginate
[59] https://github.com/mislav/will_paginate
[60] LESS erweitert CSS mit dynamischem Verhalten wie Variablen, Mixins, Berechnungen und ähnelt SASS (Syntactically Awesome Stylesheets).

`has_many`, `has_and_belongs_to_many`, etc. ausgedrückt. Abbildung 4.1 zeigt ein Modell-Diagramm von URURI, welches mit dem RubyGem *rails-erd*[61] erstellt wurde. Eine größere Ansicht ist in *erd.pdf* zu finden.

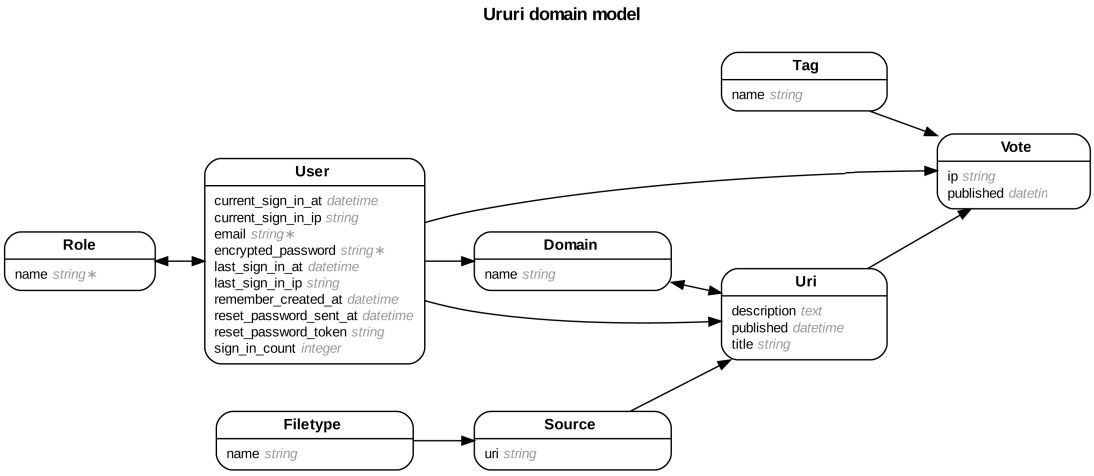

Abbildung 4.1: Modell-Diagramm von URURI

Das Kernstück der Anwendung ist das Modell *Uri*, welches eine Ressource repräsentiert. Die dem Modell zugrunde liegende Datenbankstruktur enthält eine ID der Tabelle *Source*, welche wiederum einem *Filetype*[62] zugeordnet ist und die letztlich tatsächliche URI als Zeichenkette enthält. In *Uri* sind zudem weitere Beschreibungsdaten vorhanden, wie z.B. *title*, *description* oder das Erstelldatum *published*. Diese Daten soll jeder Nutzer selber neu erstellen dürfen; sie sollen nicht exklusiv der tatsächlichen URI zugeordnet sein, daher wurde ein extra Modell *Source* erstellt, welches die tatsächliche URI Zeichenkette enthält. Eine *Uri* ist zudem einem *User* zugeordnet, wobei dieses Feld auch *NULL* sein kann, denn es soll auch unangemeldeten Nutzern möglich sein, eine URI der Anwendung hinzuzufügen. Zur Verwaltung von Wissensdomänen, also Kategorien, in denen ein Nutzer seine URIs ordnen kann, wurde das Modell *Domain* kreiert, welches letztendlich nur einen Domänennamen enthält. Dabei kann ein Datensatz des *Uri* Modells gleichzeitig mehreren Domänen zugewiesen werden. Das Modell *User* hält die registrierten Benutzer, sowie ihre Daten und weitere Informationen, zur Authentifizierung. Ein Nutzer kann dabei gleichzeitig mehreren Rollen zugeordnet sein, welche in *Role* beschrieben sind. Rollen enthalten nur eine Zeichenkette, die den Namen repäsentiert.

[61] http://rails-erd.rubyforge.org

[62] Anstatt ein ENUM-Feld für den Filetype dem Uri Modell zuzuweisen, wurde eine eigene Tabelle erstellt. ENUM-Felder werden nicht von jeder Datenbank unterstützt, zudem kann eine spätere Änderung der ENUM-Werte bei großen Datenmengen zu erweitertem Rechenaufwand führen.

Diesem Namen werden bestimmte Rechte zuteil, die in Abschnitt 4.7.1 genauer erläutert werden.

4.5 Die Datenbankstruktur

Abbildung 4.2 zeigt ein von MySQL Workbench[63] erstelltes *enhanced entity-relationship* Modell, welches hier zur Vollständigkeit aufgeführt wird. Das EER-Modell gibt die MySQL Datenbankstruktur von URURI wieder. Wie in Abschnitt 4.4 erläutert, ist die relationale Struktur in Rails selber abgebildet, wodurch in der Grafik keine Beziehungen vorkommen.

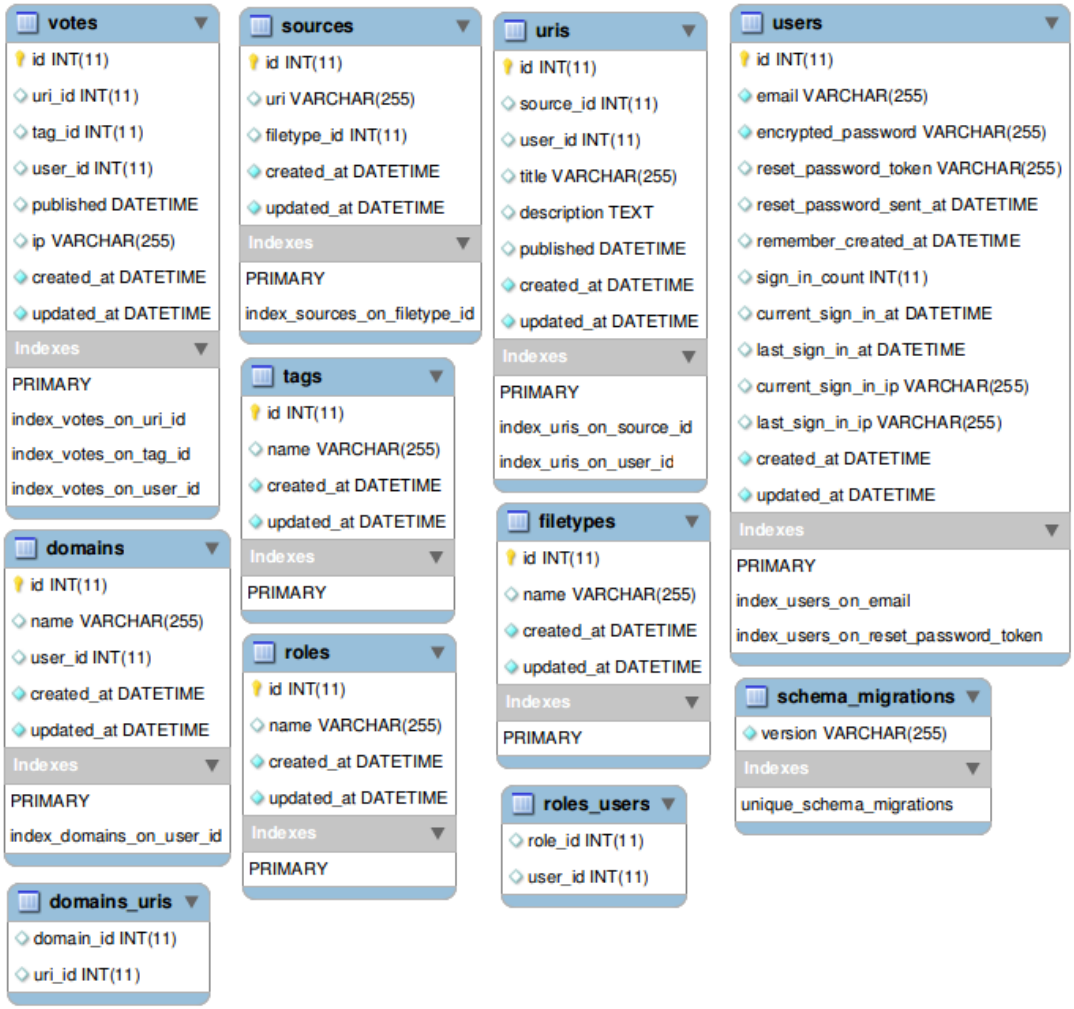

Abbildung 4.2: EER-Diagramm von URURI

[63] http://www.mysql.de/products/workbench/

Die relationalen Strukturen ließen sich auch auf Datenbankebene durch *foreign keys* implementieren, was jedoch die Modularität der Anwendung negativ beeinflussen würde und zu Problemen führen könnte; so unterstützt das relationale Datenbanksystem SQLite[64] Fremdschlüsselbeziehungen nur ansatzweise. Es stellt sich allgemein die Frage, ob eine relationale Datenbank die beste Lösung bietet und ob die Datenbank selber oder die Anwendung die Verantwortung über die gespeicherten Daten tragen solle. NoSQL[65] Lösungen brechen dort mit der traditionellen relationalen Struktur von Datenbanksystemen. NoSQL Datenbanken können mit vielen Schreib-/Leseanfragen in kurzen Zeiträumen umgehen, welches gerade bei der datenintensiven Indexierung von großen Dokumentmengen und Webseiten mit hohen Lastaufkommen einen entscheidenen Vorteil gegenüber SQL Lösungen bietet. Da URURI jedoch eine prototypische Applikation darstellt, wurde ein relationaler Ansatz auf Basis einer MySQL Datenbank gewählt.

4.6 Indexierung mittels Sphinx

Zunächst verwendet URURI eine MySQL Datenbank, wo, wie bereits in Abschnitt 4.5 erwähnt, eine NoSQL Datenbank bei Folksonomies effizienter wäre. Die eigentliche Suche geschieht jedoch mit dem quelloffenen Volltext Suchserver *Sphinx*[66]. Dieser wurde in *C++* geschrieben und ist auf nahezu jedem Betriebssystem installierbar. Mit Sphinx ist es möglich, Daten, die in SQL, sowie NoSQL Datenbanken oder Dateien (HTML, Text-Dateien, Mailboxen etc.) gespeichert sind zu indexieren. Eine Suche in den indexierten Datenbeständen sowie die Indexierung selbst können gleichzeitig erfolgen. Die Arbeit mit Sphinx gleicht der mit einem normalen Datenbankserver. Abfragen erfolgen in der eigenen Sprache *SphinxQL*, welche der SQL Syntax ähnelt. Sphinx ist im Vergleich zu normalen SQL Anfragen deutlich schneller, wenn es um große indexierte Datenbestände geht. Desweiteren bietet es die Möglichkeit mehrere Datenbanken in einem System zu indexieren, sowie eine effiziente Volltextsuche durchzuführen. Einige Vorteile von Sphinx sind im Folgenden aufgelistet:

- *Schnelle Indexierung*
- *Schnelle Suche innerhalb der Datenbestände*
- *Verteilte Suche*
  Um die Skalierbarkeit zu gewährleisten, besitzt Sphinx Möglichkeiten zur verteilten Suche in *multi-server*, *multi-core* oder *multi-cpu* Umgebungen, welches Latenzzeiten sowie die Anzahl der Querys pro Sekunde, die ausgeführt werden können, verbessern kann.
- *Ranking nach Relevanz*

[64] http://www.sqlite.org/
[65] NoSQL (Not only SQL) bezeichnet Datenbanksysteme, die einen nicht-relationalen Ansatz verfolgen.
[66] http://www.sphinxsearch.com

Sphinx enthält viele mitgelieferte *ranker*. Ein Ranker stellt eine Funktion dar, die einem Dokument bzw. einem Datensatz einen Relevanzwert zuordnet. Diese Funktionen basieren häufig auf der *phrase proximity*, ein Wert, der sich anhand der Abstände einzelner Wörter im Text berechnet, und dem *BM25* Ranking, welches anhand der Häufigkeit eines Schlagwortes generiert wird. URURI verwendet den voreingestellten Ranking Modus *SPH_RANK_PROXIMITY_BM25*, welcher beide Ansätze vereint.

- *Unterschiedliche Gewichtung einzelner Felder*
- *Multi-queries*
  Sphinx unterstützt multiple Anfragen in einer einzelnen Anfrage.
- *SQL-ähnlicher Syntax*
- *Vorhandensein diverser APIs*
  Es existieren z.B. offizielle APIs für PHP, Ruby oder Java. Drittanbieter stellen u.a. APIs für Haskell oder Ruby On Rails bereit.)

Um in Ruby On Rails *Sphinx* einfach verwenden zu können, wurde das *thinking sphinx* Gem installiert, welches von Pat Allen entwickelt wurde und unter MIT-Lizenz verfügbar ist. Die Attribute und Felder, die zur Indexierung herangezogen werden sollen, werden in *app/indices* definiert. Listing 4.1 zeigt dies anhand des Uri Modells:

Listing 4.1: app/indices/uri_index.rb

```
# Defines indexes for sphinx search engine.
# A Cronjob for 'rake ts:index' is recommended, because
# otherwise the newly added uris won't be searchable.
ThinkingSphinx::Index.define :uri, :with => :active_record do
  # fields
  indexes title, :sortable => true
  indexes description, :sortable => true
  indexes source.uri, :as => :source_uri, :sortable => true
  indexes source.filetype.name, :as => :source_filetype,
    :sortable => true
  indexes votes.tag.name, :as => :uri_tags, :sortable => true
  indexes user.email, :as => :user_email, :sortable => true

  # attributes
  has published, :sortable => true
  has user_id, :sortable => true
  has source_id
  has domains(:id), :as => :domain_ids
end
```

Nach Feldern wie `title` oder `description` kann in einer späteren Suchanfrage sortiert werden. Bei Feldern, die in anderen Tabellen vorliegen wie `source.uri`, muss ein alternativer Bezeichner zur Referenzierung mittels `:as` betitelt werden. Attribute dienen der Erweiterung von Suchanfragen. So möchte man z.B. alle Ressourcen auflisten, die in einem bestimmten Zeitraum oder von einem bestimmten Nutzer erstellt worden sind, und danach erst ein Ranking nach Relevanz durchführen.

Die eigentliche Suchanfrage wird in der `index` Methode in *app/controllers/uris_controller.rb* durchgeführt, welche in Listing 4.2 dargestellt ist.

Listing 4.2: Suche mittels thinking sphinx

```
uris = Uri.search(search_str,
  :conditions => condition_hash,
  #:group_by => :source_uri,
  :match_mode => :extended, :per_page => 8,
  :page => params[:page],
  :order => sort_column + "␣" + sort_direction)
```

Eine Gruppierung anhand der URI Zeichenkette im `Source`-Modell, um keine doppelten URIs anzuzeigen, falls zwei Nutzer dieselbe Quelle referenzieren, wurde hier zunächst auskommentiert. Die Helfermethoden `sort_column` und `sort_direction` sind ebenfalls im URIs Controller definiert und stehen auch den Views bereit, um Links zur Sortierung etc. zu generieren. Der `condition_hash` ergibt sich aus den verschiedenen Suchfiltern (Filetype, User etc.). Sphinx ermöglicht eine Volltextsuche und indexiert auch Wörter aus der Beschreibung einer URI. Das Ranking der Ergebnisse kann beeinflusst werden, wird regulär durch den Ranker *SPH_RANK_PROXIMITY_BM25* bestimmt und erfolgt u.a. anhand der Popularität der Indexe, also der Tags und zusätzlich akquirierten Deskriptoren.

Um den Datenbestand zu indexieren, die Datenstruktur neu zu analysieren oder Sphinx zu starten/stoppen, muss eine `rake task` über *thinking sphinx* ausgeführt werden. Dies geschieht aus dem Anwendungsordner z.B. über `rake ts:index`, `rake ts:rebuild` oder `rake ts:start` und `rake ts:stop`. Um eine größtmögliche Aktualität der Suche gewährleisten zu können, sollte in Abhängigkeit der sich ändernden Datenbestände ein Cronjob[67] oder Ähnliches erstellt werden.

4.7 Funktionalitäten

Im Folgenden sollen ein paar Funktionalitäten etwas genauer betrachtet werden.

4.7.1 Authentifizierung und Autorisierung

Um zwischen Administratoren und Benutzern unterscheiden zu können, sowie später verschiedene Rollen anlegen zu können, wurden eine Authentifizierung mittels dem Gem

[67] Cronjobs sind wiederkehrende Aufgaben, die in Unix und unixartigen Betriebssystemen wie Linux, BSD oder Mac OS X über den Cron-Daemon verwaltet und ausgeführt werden können.

devise und eine Autorisierung mittels dem Gem *cancan* implementiert. Die verschiedenen Rollen können unterschiedlichen Nutzen bringen; so sind Moderatorenrollen, die bestimmte Wissensdomänen managen können, denkbar. Devise ermöglicht es, verschiedene Nutzer anzulegen, und bringt Fähigkeiten wie Registrierung, Passwortrücksetzung, E-Mail Bestätigungen etc. mit sich. Es erlaubt multiplen Rollen, zur gleichen Zeit eingeloggt zu sein. Um die Befähigungen der einzelnen Rollen, Ressourcen abfragen zu dürfen, geordnet definieren zu können, wurde *cancan* verwendet, welches die Zugriffserlaubnisse in der *app/models/ability.rb* definiert. Dadurch müssen die einzelnen Berechtigungen nicht verstreut in den Controllern und Views implementiert werden. Listing 4.3 zeigt eine simple Struktur, die vorerst für URURI genutzt wird.

Listing 4.3: Rolleneigenschaften aus app/models/ability.rb

```
# define alias
alias_action :index, :show, :to => :read
alias_action :new, :to => :create
alias_action :edit, :to => :update

# guest user?
user ||= User.new

# assign abilities to roles
if user.role? :root
  can :manage, :all
elsif user.role? :admin
  can :manage, [Uri, Domain, Vote, Tag, Source]
  can [:show, :update, :destroy], User, :id => user.id
else
  can [:read, :create], Uri
  can [:destroy, :update], Uri, :user_id => user.id
  can [:get_all_names, :index], User
  can [:show, :update, :destroy], User, :id => user.id
  can :manage, Domain, :user_id => user.id
  can [:read, :create], Vote
  can [:index, :get_popular_tags], Tag
  can [:update_content_type], Source
end
```

Hierbei werden zunächst verschiedene Aliase hinzugefügt, die es ermöglichen, mehrere Methoden bzw. Aktionen unter einem Begriff zusammenzufassen. Existiert aktuell kein Nutzer, wird ein neues Nutzerobjekt erstellt. Danach wird überprüft, welche Rollen vorliegen (aktuell: root, admin, sonst normaler Nutzer), und verschiedene Rechte definiert. Ein Admin kann z.B. die Modelle `Uri`, `Domain`, `Vote`, `Source` und `Tag` managen. Dies erlaubt ihm auf genannten Modellen, jede Methode, die öffentlich ist, aufzurufen. Beim User Modell darf er lediglich die Methoden `:show`, `:edit`, `:update` und `:destroy` aufrufen und auch nur auf seinen eigenen Nutzerdaten.

In den Controllern, die zu schützende Methoden enthalten, muss dann noch `authorize_resource` nach Beginn der Klassendefinition eingefügt werden. Werden Bedingungen wie `:user_id => user.id` angeknüpft, muss die Ressource zunächst per `load_ressource` geladen werden.

4.7.2 Vorschlagssystem

In URURI wurde ein Vorschlagssystem implementiert, welches z.B. bei Eingabe eines Zeichens in ein Taggingfeld die Datenbanktabelle der Tags abfragt und nach Tags sucht, die mit der bereits eingegebenen Zeichenkette übereinstimmen. Selbiges funktioniert auch für Benutzernamen, womit sich später z.B. Freunde finden lassen könnten.

Wie bereits erwähnt, fördert dieser Ansatz zwar nicht die Vorschläge neuer Tags, jedoch verbessert er die Vereinheitlichung des Vokabulars, sowie die Bereitschaft der Anwender Tags zu verteilen. Zudem werden die populärsten Tags einer Ressource dargestellt, welches eventuell auch Nutzer inspirieren kann.

Die Darstellung des Formularfeldes zum setzen von Tags wurde als über eine von James Smith aus dem Jahre 2009 unter GPL oder MIT Lizenz geschriebene Bibliothek bereitgestellt.[68] Diese stellt ein jQuery Plugin dar, welches Nutzern erlaubt, mehrere Einträge einer vorgefertigten Liste mittels Autovervollständigung auszuwählen, und es wurde im Ordner *vendor* installiert, welcher Bibliotheken von Drittanbietern enthält.

Die `index` Methode des Tag Controllers stellt hierfür auch JSON[69] bereit, wie in Listing 4.4 zu sehen ist.

Listing 4.4: app/controllers/tags_controller.rb

```
# GET /tags
# GET /tags.json
#
# The index method for listing tags.
# It's also used for tag recommendation auto completition
# in assets/javascripts/uris.js.coffee
def index
  # get tags
  # params[:q] is used for auto-completition
  if params[:q]
    @tags = Tag.find :all, :conditions => ["name␣LIKE␣?",
          "#{params[:q]}%"]
  else
    @tags = Tag.all
  end

  respond_to do |format|
```

[68] http://loopj.com/2009/04/25/jquery-plugin-tokenizing-autocomplete-text-entry/

[69] Die JavaScript Object Notation (JSON) ist ein kompaktes Datenformat um den Datenaustausch zwischen Anwendungen zu ermöglichen.

```
    format.html # index.html.erb
    format.json { render :json => @tags }
  end
end
```

Indem die Rails Syntax verwendet wurde, sollte die Anfrage sicher gegenüber SQL Injection[70] über den Parameter `params[:q]` sein. Das eigentliche Formularfeld wird dann in der *app/assets/javascripts/uris.js.coffee* dem `div`-Element mit der ID `#form_tag_wrap` der *app/views/uris/show.html.haml* bereitgestellt.[71]

4.7.3 Pagination, Sorting und Filtering

Damit die Webanwendung vom Gefühl eher einer Desktopapplikation ähnelt wurde Ajax zur Naviergierbarkeit durch Seiten von URIs, sowie zur Sortierung und Filterung verwendet. Nicht alle Views werden aktuell per Ajax geladen, jedoch die wichtigen Elemente bei der Suche und dem Tagging, so z.B. auch die Darstellung einer Tagcloud. Dies geschieht hauptsächlich mit jQuery.

Das Gem *will_paginate* ist dabei für die Sortierung und Erstellung von Seitenumbrüchen in der URI Listendarstellung zuständig, welches jedoch um eine asynchrone Komponente erweitert wurde, indem eine *app/views/uris/index.js.erb* angelegt wurde. Listing 4.5 zeigt den Code, welcher mit der jQuery-Funktion das `div`-Element mit der ID `#uris` der eigentlichen View *app/views/uris/index.js.erb* so manipuliert, dass das Partial[72] *app/views/uris/_uris.html.erb* dorthin gerendert wird.

Listing 4.5: app/views/uris/index.js.erb

```
$("#uris").html("<%= escape_javascript(render("uris")) %>");
```

Um eine schnelle Suche nach bestimmten Inhaltstypen wie Bildern oder Videos suchen zu können, wurden in der Navigationsleiste, welche in der *app/views/layout/application.html.erb* definiert wurde, Icons zur Filterung angelegt. Per jQuery werden der reguläre Event des Links unterbunden, Informationen über den angeklickten Link angebunden und die Aktion des Suchformulars ausgeführt, welches wiederum das Partial *app/views/uris/_uris.html.erb* rendert.

4.7.4 Tagcloud

Es wurde eine reguläre Tagcloud, wie sie in Abschnitt 3.4.1 beschrieben wurde, umgesetzt. Dabei ist eine logarithmische Normierung nach dem Algorithmus von Tabelle

[70] SQL Injection bezeichnet das Ausnutzen von Sicherheitslücken von SQL Datenbanksystemen, welches anhand mangelnder Überprüfens von Nutzereingaben möglich sein kann.

[71] CoffeeScript ist eine Sprache, welche in JavaScript kompiliert und die Übersichtlichkeit von Code verstärken soll.

[72] Partial templates dienen in Ruby on Rails der Strukturierung von Programmcode in den Views. Es lassen sich damit modulare Teile einer View in eine externe Datei verlagern. Partials beginnen in Rails immer mit einem Unterstrich, welcher in der render Anweisung nach Konvention weggelassen wird, um sie von den regulären Views abzugrenzen.

3.2 vorgenommen worden. Zudem lässt sich per Mausklick auf einen Tag dieser dem Suchfeld hinzufügen. Die Grundstruktur zur Berechnung generischer Tagclouds steht soweit. Die Funktionen zur Erstellung der Tagcloud liegen in *app/assets/javascripts/jquery.tagcloud.js.coffee*. Eingebunden wird eine Tagcloud dann in der *app/assets/javascripts/application.js.coffee* über den in Listing 4.6 dargestellten Code.

Listing 4.6: Tagcloud Erstellung in app/assets/javascripts/application.js.coffee

```
# load tagcloud if there is a div with id #tagcloud
if $("#tagcloud").length > 0
  $("#tagcloud").tagcloud
    tag_count: 50
    source: "/tags/get_popular_tags.json"
    search_form: $("#uris_search")
    search_input: $("#search")
    f_min: 14
    f_max: 40
```

Dies geschieht in der applikationszugehörigen Datei, da beabsichtigt ist, in Zukunft auch Tagclouds außerhalb der Uri Views darzustellen.

4.7.5 Modulare Präsentationen

Die View *app/views/uris/show.html.haml* rendert über eine in *app/helpers/uris_helper.rb* beschriebene Methode `present(filetype)` in Abhängigkeit zu dem Inhaltstyp einer Ressource unterschiedliche modulare Partials, welche in *app/views/uris/presentations/* definiert sind. Bei Youtube Links wird z.B. der passende Einbindungscode von Videos generiert. Für ISBN Nummern wird eine Anfrage an die Google Book API gesendet, welche, falls möglich, zusätzliche Metadaten wie Inhaltsbeschreibungen oder Vorschaubilder bereitstellt. Die Anfragen erfolgen über das Gem *google-books*. Die Methoden zur Erkennung von ISBN Nummern sind in *lib/isbn.rb* definiert.

4.7.6 Zusammenfassung und Ausblick

URURI deckt aktuell die grundlegenden Aspekte einer Folksonomie ab. Simple Algorithmen zur Generierung von Tagvorschlägen und Tagclouds sind implementiert. Diese könnten in einer zweiten Version überarbeitet werden. Die aktuellen Tagvorschläge ähneln eher einer Autovervollständigung von Wörtern auf Basis der schon vergebenen Tags. Ein weiteres Aplikationselement, welches neue Tags vorschlägt, kann die Folksonomie effizienter gestalten. Ebenso kann eine generische Tagcloud die Navigierbarkeit und die explorative Suche fördern. Zusätzlich ist es eventuell sinnvoll, Teile der Struktur des Dateisystems eines Nutzers in URURI abzubilden und dem Anwender dadurch die Möglichkeit zu geben, seine lokalen Dateien zu ordnen. Dies bringt zwar keinen gemeinschaftlichen Nutzen, kann aber die Anwendung für potentielle Nutzer attraktiver machen. Zudem ließen sich so für einen Nutzer private Wissensdomänen erstellen, welche lokale Inhalte mit öffentlichen gruppieren.

Eine weiterer Aspekt, der URURI für den Nutzer interessanter gestalten würde, wären Browserplugins, welche über eine Bookmarkfunktion besuchte Webseiten zusammen mit Tags in URURI eintragen können. Eine Suchleiste für URURI im Browser ist dabei ebenso denkbar. Da die meisten Methoden in URURI auch JSON anbieten, ließe sich eine API relativ schnell implementieren.

Wie man sieht, ergeben sich viele Möglichkeiten zur Verbesserung. Dabei sind hybrid-artige Modelle, die zusätzlich Aspekte von Ontologien und anderen Ordnungssystemen mit einbeziehen, nicht einmal eingeschlossen.

5 Schlusswort

Es wurden die grundlegenden Aspekte von Social Tagging und Folksonomien erörtert. Jene Konzepte beherrschen den Umgang mit großen Datenmengen und sind nutzerfreundlich, besitzen aber weniger Ausdrucksstärke als z.B. Ontologien. Es existieren jedoch Möglichkeiten, Folksonomien zu erweitern, indem Algorithmen zur ansatzweisen Überführung in ausdrucksstärkere Systeme verwendet werden. Es wurde gezeigt, dass es relativ schnell möglich ist, eine prototypische Anwendung, basierend auf einer Folksonomie, zu erstellen, welche inhaltsunabhängig Ressourcen referenzieren kann, modulare Präsentationen der Ressourcen generiert und die typischen Aspekte des Social Taggings beinhaltet. Die Gruppierung von Ressourcen verschiedenen Inhalts zu Wissensdomänen und eine Durchsurchbarkeit derer nach Benutzerrechten erfüllt die ursprüngliche Intention, den Dozenten sowie Studenten, einen adäquaten Zugang zu den Ressourcen zu verschaffen, wobei beiden Parteien eine Suche innerhalb der Datenbestände anhand ihres eigenen Vokabulars möglich ist. Zusammenfassend lässt sich sagen, dass Folksonomien zwar nicht die Präzision von Thesauri oder die semantische Ausdrucksstärke von Ontologien besitzen, sie jedoch einen guten Grundstein zur Erstellung jener bilden. Grundsätzlich bieten sich der kollaborativen Indexierung viele Anwendungsgebiete, wobei gerade Anwendungen, die nicht das Maß an deskriptiver Präzision voraussetzen, eine Möglichkeit, Objekte relativ pflegeleicht zu ordnen. Der enorme Vorteil, bei der Indexierung nicht auf Experten angewiesen sein zu müssen, kann der zunehmenden Fülle an Inhalten eher gerecht werden als ein expertenbasierter Ansatz. Hybridartige Anwendungen könnten die „triviale“ Taggingarbeit durch den normalen Nutzer kollaborativ erledigen lassen, aus der wiederum über Algorithmen weitere Informationen und Relationen extrahiert werden, die Experten eine Basis zur weiteren Ausarbeitung bieten könnten.

6 Anhang: Verzeichnisse

Literaturverzeichnis

Aristoteles. *Philosophische Schriften in 6 Bänden*. Felix Meiner Verlag, Hamburg, 1995. ISBN 978-3787312439. URL `http://www.uni-erfurt.de/fileadmin/public-docs/Philosophie/TheoPhil/Silvere_Schutkowski/Aristoteles/Aristoteles%20Kategorien.pdf`. Übersetzt von Eugen Rolfes.

T. Berners-Lee. Uniform Resource Identifier (URI): Generic Syntax, 2005. URL `http://tools.ietf.org/html/rfc3986`. Letzter Zugriff am 29. Feb. 2012.

Céline Van Damme, Martin Hepp, and Katharina Siorpaes. FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies, 2007. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.6592`.

Francisco Echarte, José Javier Astrain, Alberto Córdoba, and Jesús Villadangos. Ontology of Folksonomy: A New Modeling Method, 2007. URL `http://www.gsd.unavarra.es/gsd/files/condep/EcAsCoVisaakm07f.pdf`. Letzter Zugriff am 10. Feb. 2012.

Herbert Frohner. *Social Tagging: Grundlagen, Anwendungen, Auswirkungen auf Wissensorganisation und soziale Strukturen der User*. Werner Hülsbusch, Berlin, 2009. ISBN 978-3598251795.

Scott A. Golder and Bernardo A. Huberman. The Structure of Collaborative Tagging Systems, 2006. URL `http://hpl.hp.com/research/idl/papers/tags/tags.pdf`. Letzter Zugriff am 18. Feb. 2012.

Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking, 2006. URL `http://www.kde.cs.uni-kassel.de/stumme/papers/2006/hotho2006information.pdf`. Letzter Zugriff am 10. Feb. 2012.

Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies, 2007. URL `http://www.kde.cs.uni-kassel.de/hotho/pub/2007/kdml_recommender_final.pdf`. Letzter Zugriff am 10. Feb. 2012.

kentbye. Tag Cloud Font Distribution Algorithm, 2005. URL `http://www.echochamberproject.com/node/247`. Letzter Zugriff am 27. Feb. 2012.

Matthias Müller-Prove. Modell und Anwendungsperspektive des Social-Taggings. In B. Gaiser, T. Hampel, and S. Panke, editors, *Good Tags - Bad Tags. Social Tagging in*

der Wissensorganisation, pages 15–22. Waxmann, 2008. ISBN 978-3830920397. URL http://www.waxmann.com/fileadmin/media/zusatztexte/2039Volltext.pdf.

Isabella Peters. *Folksonomies. Indexing and Retrieval in Web 2.0 (Knowledge and Information)*. Saur de Gruyter, Boizenburg, 2009. ISBN 978-3940317032.

Isabella Peters and Wolfgang G. Stock. Folksonomies in wissensrepräsentation und information retrieval. *Information - Wissenschaft und Praxis*, 59(2):77–90, 2008. URL http://www.phil-fak.uni-duesseldorf.de/infowiss/mitarbeiter/wissenschaftliche-mitarbeiter-hilfskraefte/isabella-peters/012-folksonomies-in-wissensrepraesentation-und-information-retrieval/.

Rashmi Sinha. Findability with tags: Facets, clusters, and pivot browsing, 2006. URL http://rashmisinha.com/2006/07/27/findability-with-tags-facets-clusters-and-pivot-browsing/. Letzter Zugriff am 10. Feb. 2012.

Jakob Voß. Tagging, Folksonomy Co - Renaissance of Manual Indexing?, 2007. URL http://arxiv.org/abs/cs/0701072. Letzter Zugriff am 27. Feb. 2012.

T. Vander Wal. Folksonomy, 2004. URL http://vanderwal.net/folksonomy.html. Letzter Zugriff am 18. Feb. 2012.

T. Vander Wal. Explaining and Showing Broad and Narrow Folksonomies, 2005. URL http://www.vanderwal.net/random/entrysel.php?blog=1635. Letzter Zugriff am 18. Feb. 2012.

Abbildungsverzeichnis

Tabellenverzeichnis