Hochschule für Technik, Wirtschaft und Kultur Leipzig
Fakultät für Informatik, Mathematik und Naturwissenschaften

Kookkurrenzbasierte Link Discovery am Beispiel von Produkttags

Masterarbeit

Sebastian Marr
mail@sebastianmarr.de
Leipzig, den 12. Januar 2014

Erstgutachter: Dr.–Ing. Toralf Kirsten
Zweitgutachter: M.Sc. Martin Breest

Zusammenfassung

Durch die Möglichkeit der Benutzerbeteiligung an der Beschreibung, Bewertung und Kategorisierung von Inhalten auf Online-Plattformen werden Begriffswelten aufgebaut, deren Auswertung großes Potenzial für die Verbesserung der Benutzererfahrung bietet. Diese Masterarbeit beschreibt ein Verfahren zum Finden von Zusammenhängen zwischen diesen Begriffen. Grundlage dafür stellen die Daten eines Tagging–Systems und die Ermittlung von Kookkurrenz dar. Die Begriffe und ihre Zusammenhänge werden in eine Graphenrepräsentation transformiert und durch Mining und Integration weiterer Datenquellen angereichert. Zur Priorisierung der Beziehungen für einen Anwendungsfall wird ein Verfahren mittels interaktiver evolutionärer Algorithmen vorstellt und angewendet. Die Ergebnisse der Erzeugung von Beziehungen und der Priorisierung werden präsentiert und schließlich die technische Umsetzung der genannten Verfahren beschrieben.

Erklärung

Ich erkläre hiermit, dass ich diese Masterarbeit selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht. Diese Arbeit ist bislang keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht worden. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

.....

Sebastian Marr
Leipzig, den 12. Januar 2014

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Zielsetzung der Arbeit 2

1.2 Aufbau der Arbeit 3

2 Tagging–Systeme **5**

2.1 Grundlagen 5

2.1.1 Datenmodell von Tagging–Systemen 6

2.1.2 Arten von Tagging–Systemen 7

2.2 Tagging–System von Spreadshirt 8

2.2.1 Spreadshirt 9

2.2.2 Eigenschaften des Tagging–Systems 11

2.2.3 Datenqualität des Tagging–Systems 11

2.2.4 Mengengerüst des Tagging–Systems 13

2.3 Zusammenfassung 13

3 Link–Discovery–Framework **15**

3.1 Modell des Weltausschnittes 15

3.2 Link–Discovery–Prozess 17

3.2.1 Integration von Datenquellen 18

3.2.2 Initiale Erzeugung des Weltausschnittes 19

3.2.3 Anreicherung des Weltausschnittes 20

3.2.4 Priorisierung von Beziehungen 21

3.3 Kookkurrenz als Mittel zur Beziehungserzeugung 23

3.3.1 Grundlagen von Kookkurrenz 23

3.3.2 Maße für Kookkurrenz 25

3.3.3 Berechnung von Kookkurrenz 26

3.4 Graphen als Beschreibungsmittel des Weltausschnittes 27

3.4.1 Grundlagen 28

3.4.2 Graphenrepräsentation des Weltausschnittes 29
3.5 Datenquellen zur Anreicherung 32
3.5.1 Lexikalische Quellen 32
3.5.2 Clicktracking–System 33
3.5.3 Verwendete Datenquellen 34
3.6 Evolutionäre Algorithmen als Mittel zur Priorisierung 35
3.6.1 Grundlagen 35
3.6.2 Anwendung zur Priorisierung 37
3.7 Zusammenfassung 38

4 Link–Discovery–Durchführung 41
4.1 Initiale Erstellung des Weltausschnittes aus Tagging–Daten 41
4.1.1 Import 42
4.1.2 Bereinigung 43
4.1.3 Reduktion 44
4.1.4 Transformation 45
4.1.5 Integration 47
4.1.6 Ergebnisse 47
4.2 Anreicherung des Weltausschnittes mit Clicktracking–Daten 48
4.2.1 Import 50
4.2.2 Bereinigung 50
4.2.3 Reduktion 51
4.2.4 Transformation 51
4.2.5 Integration 53
4.2.6 Ergebnisse 53
4.3 Anreicherung des Weltausschnittes durch Zerlegung von Wortgruppen . 54
4.3.1 Vorgehensweise 55
4.3.2 Ergebnisse 55
4.4 Anreicherung des Weltausschnittes mit Wortschatz-Daten 56
4.4.1 Import 57
4.4.2 Bereinigung 59
4.4.3 Reduktion 59
4.4.4 Transformation 60
4.4.5 Integration 61

4.4.6 Ergebnisse 62
4.5 Ergebnisse der Integrations– und Anreicherungsschritte 63
4.5.1 Anzahl der Knoten und Kanten 63
4.5.2 Verteilung der Kanten 64
4.6 Priorisierung der Beziehungen 67
4.6.1 Vorgehensweise 68
4.6.2 Ergebnisse 72
4.7 Zusammenfassung 75

5 Link–Discovery–System 77
5.1 Anforderungen an das System 77
5.1.1 Funktionale Anforderungen 77
5.1.2 Nichtfunktionale Anforderungen 78
5.2 Architektur des Systems 79
5.3 Technologieauswahl 81
5.3.1 Datenbanksystem 82
5.3.2 Implementierung der Komponenten der Architektur 85
5.3.3 Datenverarbeitung 86
5.4 Zusammenfassung 89

6 Schlussbetrachtung 91

A Ergebnisse der Priorisierung 95

Abbildungsverzeichnis 99

Tabellenverzeichnis 101

Listings 103

Literatur 105

1 Einleitung

Immer mehr Online–Plattformen geben ihren Benutzern die Möglichkeit, sich an der Beschreibung, Bewertung und Kategorisierung von Inhalten zu beteiligen. Zu diesen Beteiligungsmöglichkeiten gehören beispielsweise die Vergabe von Tags, Produktbewertungen in Online–Shops oder Kommentarfunktionen in Blogs und auf Nachrichtenseiten.

Speziell Tagging–Systeme bieten ein großes Potenzial, die Organisation von Inhalten auf Websites nachhaltig zu verändern [Shi05]. Sie erlauben den Benutzern einer Website, Inhalte mit Begriffen zu versehen, um diese zu beschreiben oder zu kategorisieren. Die Eingabe der Tags unterliegt dabei möglichst wenigen Regeln, um dem Benutzer zu ermöglichen, den Inhalt in einer für ihn natürlichen Art zu beschreiben. Die Benutzer bauen dabei eine Begriffswelt auf, die ihre Sicht auf die Inhalte der Website beschreibt.

Diese Begriffswelt besteht in einer losen Ansammlung von Begriffen und deren Verknüpfung mit Inhalten. Jedoch liegt die Vermutung nahe, dass auch zwischen den Begriffen selbst Zusammenhänge existieren. Diese werden zwar von den Benutzern nicht explizit in das System eingegeben, jedoch bei der Vergabe von Tags bedacht.

Das Finden dieser Zusammenhänge bietet einige Nutzungsmöglichkeiten, die die Benutzererfahrung auf der Website verbessern können. Denkbar sind beispielsweise Navigationsstrukturen, die diese Zusammenhänge berücksichtigen, um den Benutzer zu für ihn relevanten Inhalten zu führen. Auch die Suchfunktion einer Website kann maßgeblich verbessert werden, wenn zu einem Suchbegriff weitere Begriffe bekannt sind, die den Suchraum erweitern.

Die Suche nach Zusammenhängen in einer gegebenen Datenmenge wird als *Link Discovery* bezeichnet. Die Methoden, die dazu angewendet werden, hängen stark von der Art der Daten und der geplanten Anwendung ab. Für Tagging–Daten bietet es sich an, Kookkurrenzen zu ermitteln. Diese ergeben sich aus der Verwendung von mehreren Tags zur Beschreibung des gleichen Inhaltes. Werden Tags häufig zusammen verwendet, besteht eine hohe Wahrscheinlichkeit, dass zwischen diesen ein Zusammenhang besteht.

In den Arbeiten von Schmitz [Sch06] und Knautz, Soubusta und Stock [KSS10] wurde dieser Ansatz, angewendet auf Tagging–Systeme, beschrieben. Diese konzentrierten sich auf die Ableitung von Themenclustern aus den ermittelten Beziehungen. In der Arbeit von Schmitz [Sch06] wird ein Ansatz beschrieben, eine Ontologie aus den Daten der Foto–Plattform *Flickr* herzustellen. Dazu wird versucht, ebenfalls auf Basis von Kookkurrenz, hierarchische Beziehungen zwischen den Begriffen zu ermitteln.

In der vorliegenden Arbeit wird ebenfalls ein kookkurrenzbasierter Ansatz beschrieben, Zusammenhänge zwischen den Begriffen eines Tagging–Systems herzustellen. Dabei wird ein Weltausschnitt erstellt, der die Begriffe, den Kontext ihrer Verwendung und die Beziehungen zwischen den Begriffen enthält. Um die Nutzbarkeit dieses Weltausschnittes weiter zu verbessern, werden zusätzliche interne und externe Datenquellen integriert, um weitere Kontexte der Begriffe zu erhalten. Daraus ergeben sich Zusammenhänge verschiedener Typen. Um eine nach Relevanz geordnete Liste von Beziehungen eines Begriffes zu erhalten, wird ein Priorisierungsverfahren vorgestellt, das diese Typen gegeneinander gewichtet. Das Vorgehen wird dabei am Beispiel von Produkttags der E–Commerce–Plattform Spreadshirt demonstriert.

1.1 Zielsetzung der Arbeit

Das Ziel dieser Arbeit besteht in der Herstellung von Beziehungen zwischen Begriffen, der so genannten Link Discovery. Ausgangspunkt dafür sind die Daten eines Tagging–Systems. Dazu wird zuerst ein Framework definiert, welches die theoretischen Grundlagen der Link Discovery beschreibt. Dieses Framework wird zur Durchführung der Link Discovery an konkreten Daten verwendet und die Ergebnisse präsentiert. Die inhaltliche Qualität der Beziehungen hängt stark von den verwendeten Daten und der geplanten Anwendung ab und liegt nicht im Fokus dieser Arbeit. Außerdem werden die Anforderungen an ein System, das die Link Discovery technisch umsetzt, formuliert und deren Realisierung diskutiert.

Im Detail werden die folgenden Fragen beantwortet:

- Was sind Tagging–Systeme, welche Arten von Tagging–System gibt es und welche besonderen Eigenschaften hat das beispielhaft verwendete System des Unternehmens Spreadshirt?

- Was ist Link Discovery und wie kann diese umgesetzt werden, um Beziehungen aus Produkttags zu extrahieren?
- Was ist Kookkurrenz und wie kann diese zur Link Discovery genutzt werden?
- Wie kann eine Graphenrepräsentation genutzt werden, um die Ergebnisse der Link Discovery abzubilden?
- Wie können die erzeugten Beziehungen durch Data Mining oder Integration weiterer Datenquellen angereichert werden und welche Datenquellen sind dazu geeignet?
- Wie können die erzeugten Beziehungen mittels interaktiver evolutionärer Algorithmen priorisiert werden, um für einen Anwendungsfall relevante Nachbarn eines Begriffes zu erhalten?
- Welche technischen Anforderungen stellt die Link Discovery und wie kann die Berechnung von Beziehungen implementiert und beschleunigt werden?

1.2 Aufbau der Arbeit

Die Ergebnisse dieser Arbeit werden in vier Hauptkapiteln vorgestellt. Zunächst werden in Kapitel 2 Tagging–Systeme im Allgemeinen und das beispielhaft verwendete System des Unternehmens Spreadshirt im Speziellen erläutert und die Eigenschaften, Datenqualität und Menge der Daten diskutiert.

In Kapitel 3 werden mit dem Link–Discovery–Framework die theoretischen Grundlagen für die spätere Durchführung der Link Discovery definiert. Dies umfasst die Modellierung des betrachteten Weltausschnittes in Abschnitt 3.1, die Beschreibung des Vorgehens in Abschnitt 3.2, Kookkurrenz als Mittel zur Beziehungserzeugung in Abschnitt 3.3, die Überführung des Weltausschnittes in eine Graphenrepräsentation in Abschnitt 3.4, mögliche Datenquellen zur Anreicherung in Abschnitt 3.5 und die Einführung evolutionärer Algorithmen zur Priorisierung von Beziehungen in Abschnitt 3.6.

Kapitel 4 beschäftigt sich mit der Umsetzung des in Kapitel 3 beschriebenen Frameworks an konkreten Daten. Dazu wird in Abschnitt 4.1 die initiale Erstellung aus den Daten des Tagging–Systems von Spreadshirt beschrieben. Diese Daten werden in Abschnitt 4.2 mit den Daten des Clicktracking–Systems von Spreadshirt angereichert. In Abschnitt 4.3 wird

die weitere Anreicherung durch Zerlegung von Wortgruppen beschrieben. Ein letzter Anreicherungsschritt besteht in der in Abschnitt 4.4 beschriebenen Integration der Daten des Wortschatzes der Universität Leipzig. Die quantitativen Ergebnisse dieser Schritte werden in Abschnitt 4.5 ausgewertet. In Abschnitt 4.6 wird die konkrete Priorisierung der erzeugten Beziehungen erläutert und die Ergebnisse diskutiert.

Aspekte der technischen Umsetzung des Link–Discovery–Systems werden in Kapitel 5 beschrieben. Dazu gehören die Formulierung der Anforderungen an ein solches System in Abschnitt 5.1, die Beschreibung der Systemarchitektur in Abschnitt 5.2 sowie die erfolgte Technologieauswahl in Abschnitt 5.3.

Den Abschluss der Arbeit bildet Kapitel 6 mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf mögliche weitere Arbeiten.

In dieser Arbeit wurden die Diagrammtypen der Methodik FMC [KGT06] zur Modellierung von Architektur, Prozessen und Daten verwendet. Code–Beispiele für Beispieldaten sind in JSON–Notation [Cro06] formuliert.

Der Quelltext dieser Arbeit und der implementierten Link–Discovery–Schritte wurden unter http://github.com/sebastianmarr/thesis veröffentlicht.

2 Tagging–Systeme

Das folgende Kapitel beschäftigt sich mit Tagging–Systemen. Dabei werden die Grundlagen, das Datenmodell und die Arten von Tagging–Systemen erläutert sowie das System von Spreadshirt, das in dieser Arbeit verwendet wurde, genauer erklärt. Dies umfasst die spezifischen Eigenschaften dieses Systems, die Diskussion der Datenqualität sowie das Mengengerüst der vorhandenen Daten.

2.1 Grundlagen

Tags sind eine Form von Metadaten, also "Daten über Daten". Sie erfüllen die Funktion der *Beschreibung* von Dokumenten und werden im Allgemeinen von Benutzern angelegt, im Gegensatz zu professionell kuratierten Metadaten wie beispielsweise Bibliothekskatalogen [Mat04].

Im Allgemeinen sind Tags kurze Schlagworte, die von dem Benutzer, der sie vergibt, frei gewählt werden können. Jedes Dokument kann mit beliebig vielen Tags versehen werden. Dies steht im Gegensatz zu einer festen, vorgegebenen Klassifikation mittels Kategoriebäumen, wie sie beispielsweise in E–Commerce–Systemen üblich sind, um Artikel zu ordnen. In solchen Kategoriebäumen kann ein Dokument üblicherweise nur in einer begrenzten Anzahl von Kategorien, meistens nur in einer, eingeordnet werden.

Die Menge der Tags eines Systems ist nicht hierarchisch geordnet und es bestehen keine explizit formulierten Beziehungen zwischen einzelnen Tags. Somit ergibt sich eine lose Kategorisierung der Dokumente, die, im Gegensatz zu formalen Taxonomien und Ontologien, ständig von den Benutzern erweitert und verändert wird [Shi05]. Jacob [Jac04] beschäftigt sich tiefer gehend mit dem Unterschied zwischen starrer Klassifikation und loser Kategorisierung.

Aus den genannten Eigenschaften ergeben sich bestimmte Schwächen und Stärken von Tagging–Systemen, die von Mathes [Mat04] definiert wurden. Demnach liegen die Schwächen in der Mehrdeutigkeit von Tags und der mangelnden Kontrolle von Synonymen. Mehrdeutigkeit bezeichnet den Umstand, dass gleiche Tags zur Beschreibung von sehr unterschiedlichen Dokumenten genutzt werden können, da keine Systematik vorgegeben ist. Die mangelnde Kontrolle von Synonymen führt dazu, dass verschiedene Tags verwendet werden, um den gleichen Sachverhalt zu beschreiben.

Zu den von Mathes [Mat04] formulierten Stärken gehören die starke Ausrichtung von Tags an den Gedankengängen der Benutzer und dem einfacheren Durchstöbern von Dokumenten. Da die Tags von den Benutzern eines Systems formuliert werden, spiegeln sie deren Vokabular und deren Gedankengänge wieder. Werden Tags statt zum Finden von konkreten Dokumenten zum Durchstöbern genutzt, bieten sie größere Möglichkeiten, interessante Inhalte zu finden, als in eine starre Kategorisierung.

Tagging–Systeme enthalten demnach implizites Wissen, das durch Data Mining extrahiert und genutzt werden kann. Im Rahmen dieser Arbeit wurden Tagging–Systeme als Ausgangspunkt für die Link Discovery mittels Kookkurrenz genutzt (siehe Abschnitte 3.3 und 4.1).

2.1.1 Datenmodell von Tagging–Systemen

Ein Tagging–System ist allgemein durch ein Tripel $S = (D, T, U)$ von Mengen, sowie durch die Relation $R = D \times U \times T$ definiert.

D repräsentiert eine Menge von Dokumenten. Ein Dokument d kann ein beliebiger Datensatz sein, beispielsweise ein Design, Artikel oder Produkt. Die Menge U stellt alle Benutzer des Systems dar. Ein Benutzer u ist eine Entität mit beliebigen weiteren Attributen, die jedoch im Kontext des Tagging–Systems nicht weiter betrachtet werden. T repräsentiert die Menge der Tags. Ein Tag t ist eine Entität, die als benötigtes Attribut eine Zeichenkette besitzt, die zur Beschreibung von Dokumenten genutzt werden kann. T bildet das *Vokabular* des Tagging–Systems.

Die Relation R beschreibt den Vorgang des *Taggings*. Ein Benutzer u des Systems vergibt einen Tag t an ein Dokument d, um den Inhalt von d mit der Zeichenkette von t zu be-

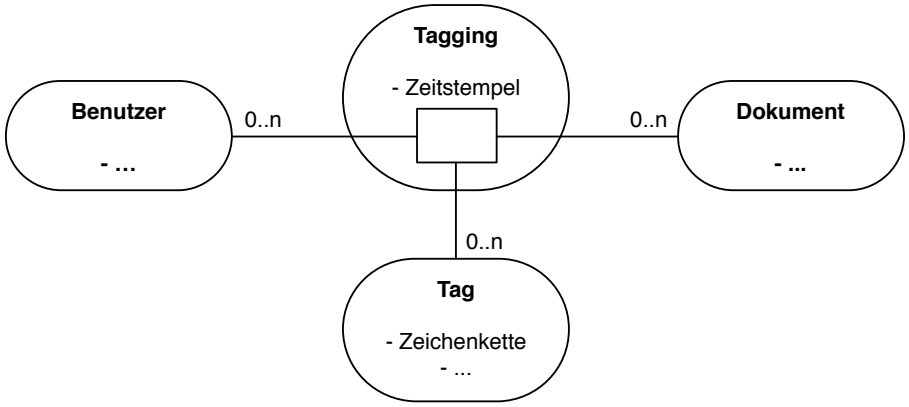


Abbildung 2.1: FMC–Entity–Relationship–Diagramm eines Tagging–Systems

schreiben. Der Zeitpunkt der Vergabe des Tags wird durch einen Zeitstempel ts repräsentiert. R enthält demnach Quadrupel der Form (d, t, u, ts).

Das beschriebene Tagging–System lässt sich somit in ein Datenmodell mit den Entitätstypen *Benutzer*, *Dokument*, *Tag* und der ternären Beziehung *Tagging* überführen. Dieses Modell ist in Abbildung 2.1 als Entity–Relationship–Diagramm dargestellt. Alle genannten Entitätstypen können weitere Attribute besitzen, die von der Anwendungsdomäne des konkret betrachteten Tagging–Systems abhängen.

2.1.2 Arten von Tagging–Systemen

Abhängig vom gewünschten Einsatzzweck des Tagging–Systems kann der Betreiber bestimmte Aspekte des Systems beschränken. Außerdem können die Benutzer einen vorherrschenden Umgang mit dem System entwickeln. Aus diesen Faktoren ergeben sich verschiedene Arten und Nutzungsmuster von Tagging–Systemen. Hauptsächlich kann zwischen offenen Tagging–Systemen, den so genannten *Folksonomies* und den geschlossenen Tagging–Systemen unterschieden werden.

Folksonomies

Eine Folksonomy beschreibt ein offenes Tagging–System [Mat04]. Bei dieser Art von System kann grundsätzlich jeder Benutzer jeden Tag an jedes Dokument vergeben. Außerdem

stammen die Dokumente selbst meist ebenfalls von den Benutzern. Beispiele für Folksonomies sind der Bookmarking–Dienst *Delicious* [Del] und die Foto-Plattform *Flickr* [Flr].

Der Begriff *Folksonomy* steht im Gegensatz zur *Taxonomie* und beschreibt den Umstand, dass die Kategorisierung und Ordnung von Inhalten vom *folk*, also den Benutzern selbst vorgenommen werden [Van07].

Geschlossene Tagging–Systeme

In geschlossenen Tagging–Systemen beschränkt der Betreiber des Systems bestimmte Aspekte. Dies können die Benutzer, die Tags vergeben dürfen, die Dokumente oder auch das Vokabular sein.

Eine häufige Form der Einschränkung, der auch das in dieser Arbeit verwendete System unterliegt (siehe Abschnitt 2.2.2), ist die Einschränkung der Benutzer, die ein Dokument taggen können. Oftmals ist dies nur den Autoren des Dokumentes selbst oder Benutzern mit besonderen Rechten, beispielsweise Moderatoren oder Angestellten des Betreibers, erlaubt.

In den meisten Tagging–Systemen stammen die Dokumente ebenfalls von den Benutzern des Systems, beispielsweise Artikel, Fotos oder Musikstücke. Jedoch kann die Erstellung der Dokumente eingeschränkt werden, wenn dies in der Anwendungsdomäne sinnvoll ist. Beispiele hierfür sind Produkte in Online–Shops. Diese werden nicht von den Benutzern erstellt, jedoch kann die Vergabe von Tags an diese Produkte einen Mehrwert liefern.

Die Einschränkung des Vokabulars kann vorgenommen werden, um Rechtschreibfehler und Fehleingaben der Tags zu vermeiden. Sie bringt jedoch den Nachteil mit sich, dass die Tags dann unter Umständen nicht mehr das Vokabular der Benutzer widerspiegeln und somit Inhalte für diese schwerer auffindbar sind.

2.2 Tagging–System von Spreadshirt

Nachdem im vorherigen Abschnitt die Grundlagen von Tagging–Systemen diskutiert wurden, beschäftigt sich dieser Abschnitt mit dem konkreten Tagging–System der Website

Spreadshirt, welches in dieser Arbeit für den initialen Schritt der Link Discovery genutzt wurde (siehe Abschnitt 4.1).

2.2.1 Spreadshirt

Die vorliegende Masterarbeit wurde im Kontext der sprd.net AG (Spreadshirt) [Sprd] erstellt. Spreadshirt ist eine E–Commerce–Plattform, die es seinen Benutzern erlaubt, personalisierte Textilien und andere Artikel zu gestalten, zu kaufen und zum Verkauf anzubieten. Spreadshirt übernimmt die Produktion und den Versand der Produkte. Ein Produkt bezeichnet hierbei einen Produkttyp, beispielsweise ein T–Shirt, der mit einem oder mehreren Designs bedruckt wurde.

Das Erstellen von Designs und die Konfiguration eines Produktes, also das Positionieren von Designs auf Produkttypen, wird vollständig vom Benutzer durchgeführt. Es agieren grundsätzlich zwei Arten von Benutzern mit der Spreadshirt–Plattform: *Kunden* und *Partner*.

Als Kunden werden Benutzer bezeichnet, die Produkte bestellen. Diese Produkte können entweder von ihnen selbst oder von einem Partner erstellt worden sein.

Partner sind Benutzer, die Designs oder Produkte erstellen und diese zum Verkauf anbieten. Zu diesem Zweck kann der Partner einen eigenen Shop auf der Spreadshirt–Plattform eröffnen. Kunden können in diesem Shop Produkte bestellen und der Partner erhält einen Anteil des Verkaufspreises, während Spreadshirt die Produktion und den Versand an den Kunden übernimmt.

Neben den von Kunden für sich selbst erstellten Produkten und den Partner–Shops existiert mit dem Spreadshirt–Marktplatz ein weiterer Vertriebskanal. Auf dem Marktplatz können Partner nach ihrer Zustimmung ihre Designs vertreiben. Kunden können nach Motiven suchen, die ihrem Geschmack entsprechen und diese bestellen, mit anderen Motiven kombinieren oder mit Texten versehen. Ein Produkt, das entweder in einem Partner–Shop oder auf dem Marktplatz positioniert und mit einem Preis versehen wurde, wird Artikel genannt.

Die grundsätzliche Funktionsweise der Spreadshirt–Plattform ist in Form eines FMC–Blockdiagrammes in Abbildung 2.2 dargestellt.

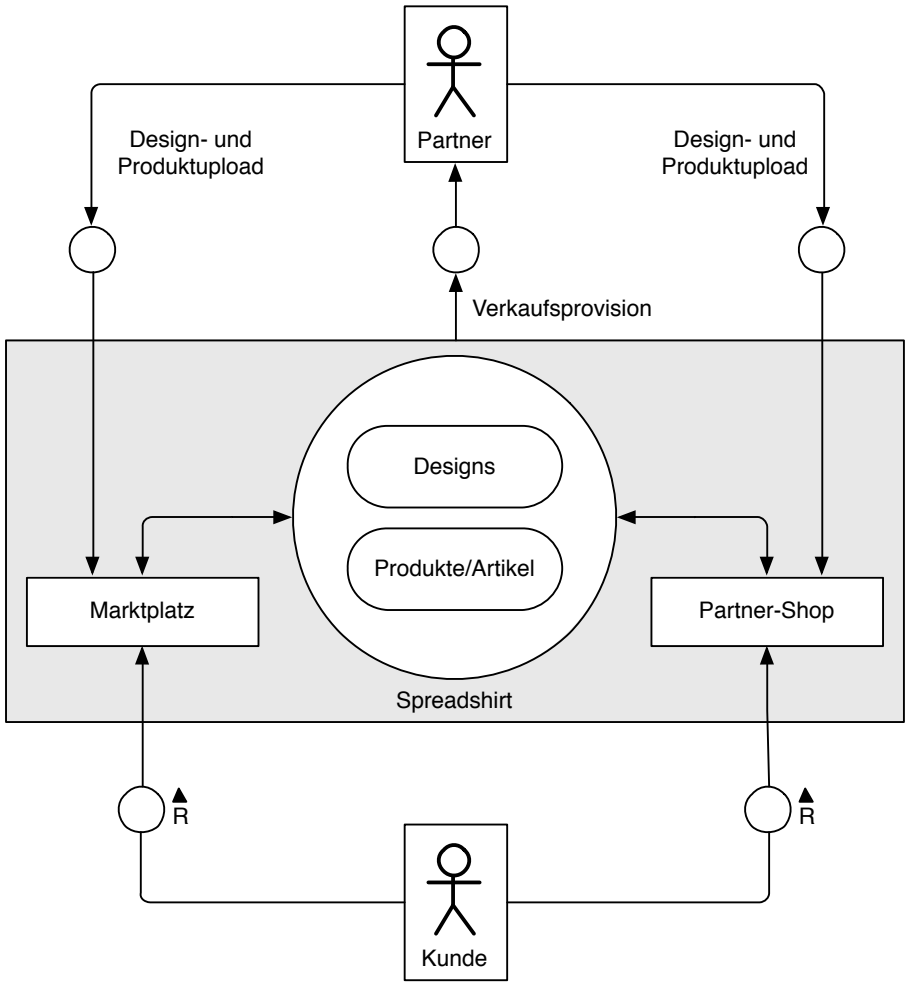


Abbildung 2.2: FMC–Blockdiagramm der Spreadshirt–Bereiche und Benutzer

Das Suchergebnis für Suchen auf dem Marktplatz hängt maßgeblich von den Metadaten ab, die der Partner für seine Designs oder Produkte vergeben hat. Dazu gehören Tags, aber auch Titel und Beschreibung des Designs oder Produktes. In dieser Arbeit wird ausschließlich das Tagging–System betrachtet.

Spreadshirt betreibt aus historischen Gründen zwei Plattformen, deren Datenbestände größtenteils voneinander getrennt sind. Jeweils eine Plattform ist für den nordamerikanischen und den europäischen Markt zuständig. Im Kontext dieser Arbeit wird die europäische Plattform als Ausgangsbasis für alle Betrachtungen gewählt. Der Datenbestand dieser Plattform besteht aus circa 2 Millionen Tags, 6 Millionen Designs, 14 Millionen Produkten, 6 Millionen registrierten Nutzern und 750 000 eröffneten Partner–Shops.

2.2.2 Eigenschaften des Tagging–Systems

Im Fall von Spreadshirt ist die Vergabe von Tags auf die Menge der Partner $P \subseteq U$ begrenzt (siehe auch Abschnitt 2.2.1). Es handelt sich demnach um ein in Abschnitt 2.1.2 beschriebenes geschlossenes Tagging–System.

Die Dokumente, die von den Partnern getaggt werden können, sind auf die Designs und Produkte beschränkt, die der Partner selbst angelegt hat. Eine Beschreibung kann somit ausschließlich durch den Autor des Inhaltes erfolgen. Deshalb fehlt im Vergleich zu anderen Tagging–Systemen auch die Information, welcher Benutzer den Tag vergeben hat, da diese implizit durch den Autor des Dokumentes gegeben ist.

Des Weiteren besitzen Tags in der Spreadshirt–Datenbank ein Attribut *Sprache* aus der Menge L. Die Sprache spielt bei der Eingabe und Anzeige der Tags zu Dokumenten eine Rolle. Je nach eingestellter Sprache auf der Website erstellt und sieht der Benutzer nur Tags, die mit dieser Sprache markiert sind.

Das Vokabular der Tags ist nicht eingeschränkt. Dies bringt zwangsläufig Probleme der Datenqualität mit sich, welche im folgenden Abschnitt erläutert werden.

2.2.3 Datenqualität des Tagging–Systems

Die Qualität von Daten wird im Allgemeinen unter mehreren Gesichtspunkten beurteilt. Dazu gehören unter anderem *Korrektheit*, *Vollständigkeit*, und *Redundanzfreiheit* [HKP12, S. 84 f.]. Nachfolgend werden die bei Spreadshirt vorhandenen Tagging–Daten nach diesen Kriterien betrachtet und die Quellen eventueller Fehler [Ols, S. 43 f.] diskutiert.

Korrektheit

Die Korrektheit der Tagging–Daten kann an vielen Punkten angezweifelt werden. Das hervorstechende Problem hierbei ist das Auftreten von Spam. Viele Partner versehen ihre Artikel und Designs mit Tags, die nicht den Inhalt beschreiben. So werden beispielsweise falsche Tags vergeben, damit die getaggten Dokumente bei populären Suchbegriffen in der Ergebnisliste erscheinen.

Ein weiterer Defekt ist die Inkorrektheit des Attributes *Sprache* der Tags. Die Sprache wird aus der Domain abgeleitet, die der Benutzer, der den Tag eingegeben hat, besucht hat. Viele Partner geben jedoch ihre Tags in mehreren Sprachen ein, um ihre Inhalte besser auffindbar zu machen. Dies führt in der Konsequenz dazu, dass das Attribut Sprache in einem nicht unwesentlichen Teil der Tags als falsch angesehen werden kann.

Die Quelle beider Fehler ist die bewusste Falscheingabe von Informationen, um einen persönlichen Vorteil zu erlangen, da die Partner versuchen, ihre Designs und Produkte möglichst zu vielen Sucheingaben in den Ergebnissen auftauchen zu lassen.

Vollständigkeit

Wie bereits in Abschnitt 2.2.2 beschrieben, fehlt in den Daten des Spreadshirt–Systems die Angabe, welcher Benutzer einen Tag vergeben hat. Außerdem besitzen die Taggings keinen Zeitstempel. Dies führt in der Konsequenz dazu, dass Spam schwerer erkannt werden kann. Zwar ist bekannt, wann ein Tag das erste Mal verwendet wurde, alle weiteren Verwendungen des Tags haben jedoch keinen Zeitstempel. Der Benutzer, der den Tag angelegt und verwendet hat, kann nur daraus abgeleitet werden, von wem das getaggte Dokument angelegt wurde.

Die Unvollständigkeit der Daten rührt in erster Linie daher, dass zum Zeitpunkt der Implementierung des Tagging–Systems noch nicht bedacht wurde, dass die fehlenden Attribute später nützlich sein können.

Redundanzfreiheit

Bedingt durch die Form der Dateneingabe besteht für das Vokabular des Tagging–Systems ein großes Potenzial für redundante Daten. Da eingegebene Tags durch einen Separator getrennt eingegeben werden müssen, besteht hier ein Risiko für Fehleingaben. Wird der falsche Separator verwendet, werden die eigentlich getrennten Tags als eine einzige Entität abgespeichert.

Technisch kann jeder Tag genau ein Mal in der Datenbank vorkommen. Jedoch führen Tipp– und Rechtschreibfehler, unterschiedliche Groß- und Kleinschreibung, verschiedene

Schreibweisen zusammengesetzter Wörter und Leerräume vor, nach und zwischen Wörtern eines Tags dazu, dass das gleiche Wort mehrfach in der Datenbank gespeichert wurde.

Außerdem führten in der Vergangenheit Systemfehler und Implementierungsfehler dazu, dass falsche, nicht druckbare Zeichen in den Tags enthalten waren. Nach Beseitigung der Fehler blieben die fehlerhaften Tags bestehen, so dass bei einer erneuten Eingabe des gleichen Wortes ein neuer Tag in der Datenbank angelegt wurde.

2.2.4 Mengengerüst des Tagging–Systems

Zum Zeitpunkt der Bearbeitung dieser Arbeit befanden sich im Datenbestand der europäischen Spreadshirt–Plattform:

- 2 072 079 Tags in 15 verschiedenen Sprachen
- 6 433 410 Benutzer
- 26 147 860 Dokumente (16 494 430 Artikel und 9 653 430 Designs)
- 71 938 905 Taggings

Diese Datenmengen stellen besondere Anforderungen an die Verarbeitung. Wie diese umgesetzt wurden, wird in Kapitel 5 diskutiert.

2.3 Zusammenfassung

Dieses Kapitel befasste sich mit Tagging–Systemen. Dazu wurden die Grundlagen und Begriffe genannt, die Vor– und Nachteile von Tagging–Systemen erläutert, das Datenmodell formalisiert und die grundsätzlichen Unterschiede zwischen Folksonomies und geschlossenen Tagging–Systeme beschreiben. Außerdem wurde das im weiteren Verlauf dieser Arbeit verwendete Tagging–System von Spreadshirt [Sprd] näher betrachtet. Dazu wurden die speziellen Eigenschaften dieses Systems, die Datenqualität und die Menge der vorhandenen Daten beschrieben.

Das folgende Kapitel beschäftigt sich mit der Definition des Frameworks für die Link Discovery.

3 Link–Discovery–Framework

Link Discovery [NA11; Vol+09], oder *Link Prediction* [Tas+03; LK07], bezeichnet Methoden des Data Minings [HKP12], die zum Ziel haben, Verbindungen zwischen Objekten herzustellen. Diese Verbindungen werden aus vorhandenen Daten abgeleitet.

Das folgende Kapitel spezifiziert das Framework, das für die Link Discovery im Rahmen dieser Arbeit angewendet wurde. Das Framework beschreibt die Kombination aus dem modellierten Weltausschnitt, dem Prozess zur Herstellung der Beziehungen, Kookkurrenz als Maß für Beziehungen, Graphen als Mittel zur Beschreibung des Weltausschnittes, möglichen Datenquellen und evolutionären Algorithmen als Mittel zur Priorisierung der erzeugten Beziehungen.

3.1 Modell des Weltausschnittes

Der folgende Abschnitt beschäftigt sich mit der Modellierung des im Rahmen dieser Arbeit verwendeten Weltausschnittes. Das Entity–Relationship–Diagramm des resultierenden Modells ist in Abbildung 3.1 dargestellt.

Zentrale Entität des Modells ist der *Begriff*. Ein Begriff repräsentiert ein Einzelwort oder eine Wortgruppe in einer bestimmten Sprache. Dies berücksichtigt den Umstand, dass Wörter in mehreren Sprachen vorkommen können, jedoch in den jeweiligen Kulturräumen verschiedene Bedeutungen besitzen. Wortgruppen können aus beliebig vielen Einzelwörtern zusammengesetzt sein. Mit Hilfe der Link Discovery sollen zwischen diesen Begriffen verschiedenartige *Zusammenhänge* gefunden werden.

Ein Zusammenhang (oder *Beziehung*) besteht immer zwischen genau zwei Begriffen und besitzt einen bestimmten *Typ*. Der Typ bezeichnet die Art des Zusammenhangs zwischen diesen beiden Begriffen. Beispiele für Zusammenhangstypen sind inhaltliche Zusammenhänge wie Synonyme, grammatikalische Zusammenhänge wie Wort– und Grundformen

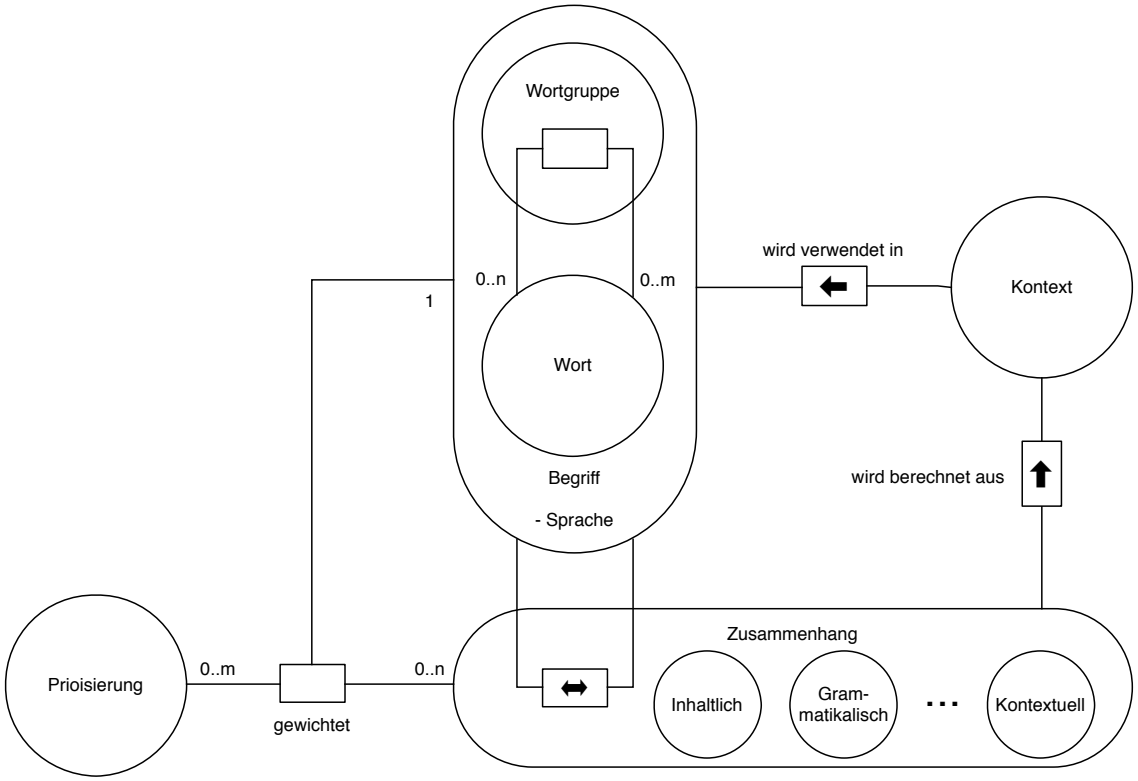


Abbildung 3.1: FMC–Entity–Relationship–Diagramm des Modells des Weltausschnittes

oder kontextuelle Zusammenhänge, die sich aus der Verwendung des Begriffes ergeben. Dabei kann ein Zusammenhang, abhängig vom Typ, Attribute besitzen, die den Zusammenhang genauer spezifizieren. Dies kann beispielsweise ein Gewicht des Zusammenhangs sein, das die Wichtigkeit gegenüber anderen Beziehungen gleichen Typs angibt.

Je nach Nutzungsform der Daten wird unter Umständen eine andere Sicht auf die Beziehungen benötigt. Eine *Priorisierung* stellt eine Gewichtung der Beziehungen eines Begriffes nach Typ dar. Sie teilt jeder Zusammenhangsart ein Gewicht relativ zu den anderen Arten zu. Somit werden durch die Priorisierung bestimmte Zusammenhänge höher gewichtet als andere. Die Priorisierung wird zu einer auf den Anwendungsfall abgestimmten Ordnung der Beziehungen eines Begriffes genutzt.

Die Verwendung eines Begriffes wird durch den *Kontext* beschrieben. Dieser Kontext repräsentiert, wie der Begriff innerhalb einer bestimmten Anwendungsdomäne verwendet wurde. Daher sind die Attribute, die ein Kontext besitzen kann, nicht vorab spezifizierbar. Sie hängen von der jeweiligen Anwendungsdomäne ab. Beispiele für Kontexte sind

die Verwendung eines Begriffes in einem Tagging–System oder in einer Ontologie. Aus diesem Kontext werden im Laufe der Link Discovery Zusammenhänge berechnet.

Dieses Modell bildet die Grundlage für den im folgenden Abschnitt beschriebenen Link–Discovery–Prozess.

3.2 Link–Discovery–Prozess

Der Link–Discovery–Prozess beschreibt die Abfolge von Schritten, die zur Erzeugung und Anreicherung des in Abschnitt 3.1 beschriebenen Weltausschnittes durchgeführt werden. Er dient somit zur Erzeugung von Begriffen und deren Zusammenhängen. Abbildung 3.2 zeigt den Prozess als Petri–Netz.

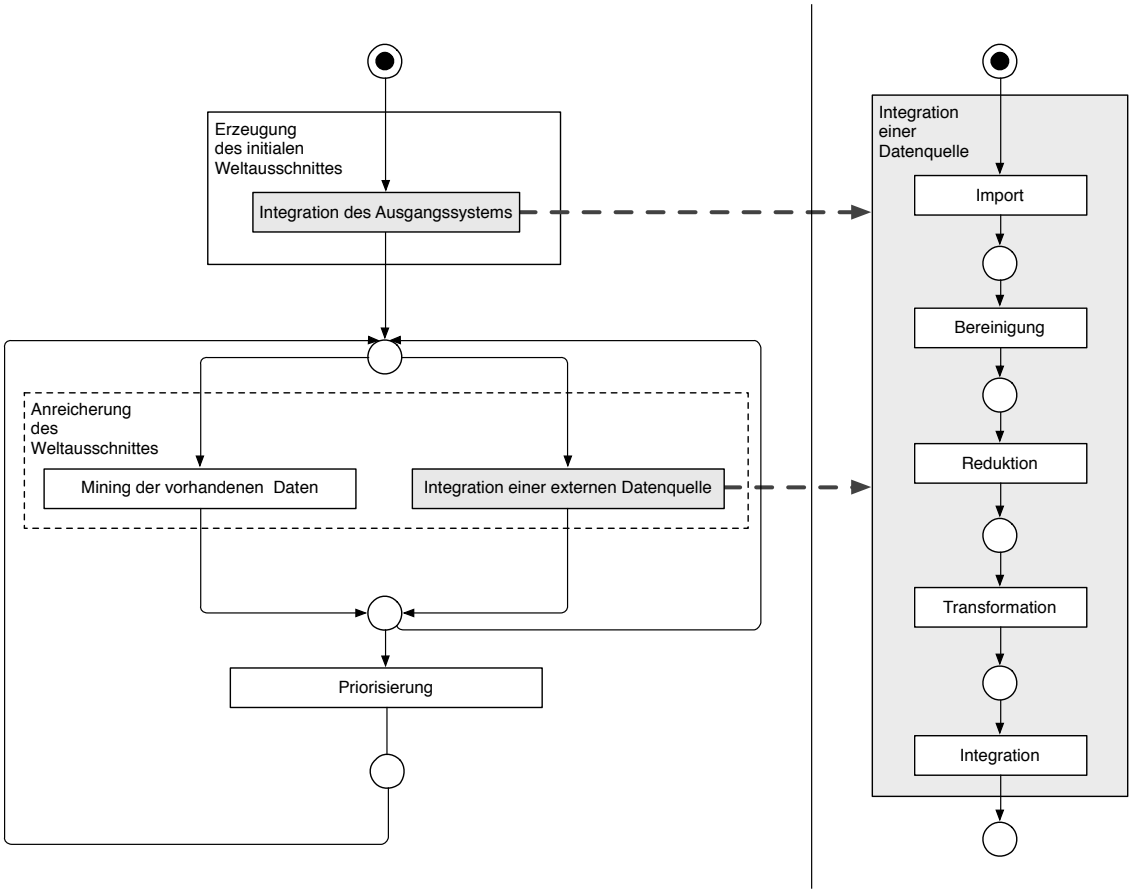


Abbildung 3.2: FMC–Petri–Netz des Link–Discovery–Prozesses

Die grundlegenden Phasen des Prozesses sind die initiale *Erzeugung* des Weltausschnittes, dessen *Anreicherung* und die *Priorisierung* der Beziehungen. Sowohl bei der initialen Erzeugung als auch bei der Anreicherung wird ein Prozess zur Integration von Datenquellen benötigt.

Die Schritte der Anreicherung und Priorisierung können beliebig oft wiederholt werden, um das Ergebnis zu verbessern und auf die gewünschte Anwendung anzupassen. Die Anreicherung kann grundsätzlich durch das Mining der bereits im Weltausschnitt vorhandenen Daten oder durch die Integration neuer Datenquellen erfolgen.

Die genannten Schritte werden in den folgenden Abschnitten erläutert.

3.2.1 Integration von Datenquellen

Für die Link Discovery wird ein einheitliches Vorgehen zur Integration von Datenquellen benötigt. Die Datenquellen stellen grundsätzlich für die Link Discovery nützliche Daten zur Verfügung, die jedoch im Allgemeinen noch nicht direkt dem Modell des Weltausschnittes entsprechen. Demzufolge müssen diese Daten entsprechend verarbeitet werden.

Die zur Integration von externen Datenquellen nötigen Schritte entsprechen im Wesentlichen den von Han, Kamber und Pei [HKP12, S. 48f.] beschriebenen Aufgaben der Datenvorverarbeitung: *Bereinigung*, *Reduktion*, *Transformation* und *Integration*. Diesen wird in dieser Arbeit der Schritt *Import* vorangestellt, da es nicht immer möglich oder nötig ist, den gesamten Datenbestand einer Datenquelle zu nutzen. Somit sollten im Importschritt auch die Anfragen an die Datenquelle spezifiziert werden. Die genannten Schritte werden zur Link Discovery immer in der genannten Reihenfolge ausgeführt und werden im folgenden kurz beschrieben.

Import Im Importschritt werden die Rohdaten aus der Datenquelle extrahiert. Dabei wird die Form der Daten nicht verändert. Ist es nicht möglich oder nötig, den gesamten Datenbestand einer Quelle zu importieren, so muss eine Auswahl der anzufragenden Daten formuliert werden. Diese Auswahl richtet sich nach Möglichkeit nach den bereits im Weltausschnitt vorhandenen Daten. Beispielsweise können die Anfragen an die Datenquelle den bereits im Weltausschnitt gespeicherten Begriffen entsprechen.

Bereinigung Im nachfolgenden Bereinigungsschritt werden die importierten Daten so gut wie möglich von eventuell vorhandenen Defekten bezüglich der Datenqualität (siehe auch Abschnitt 2.2.3) befreit. Dazu zählen beispielsweise die Entfernung von nicht nutzbaren Zeichen oder unvollständigen Datensätzen.

Reduktion Der Reduktionsschritt dient zur Verkleinerung der Datenmenge. Dazu gehören beispielsweise Schritte zur Duplikatentfernung oder zur Auswahl relevanter Datensätze. In dieser Arbeit bestand die Haupteinschränkung der Datenmenge darin, nach Möglichkeit nur deutschsprachige Begriffe auszuwählen.

Transformation Der Schritt der Transformation überführt die Daten schließlich in das Modell des Weltausschnittes. Dies bedeutet, dass die Datensätze in Begriffe und Beziehungen umgewandelt werden. Dabei sollten möglichst viele Informationen über den Kontext der Begriffe erhalten bleiben. Die Methode, nach der diese Transformation vorgenommen wird, hängt von der Datenquelle ab. Generell werden die Beziehungen meist aus dem Kontext der Begriffe, wie er in der Datenquelle vorliegt, gebildet. Ein Mittel für die Beziehungserzeugung ist die Kookkurrenz, welche in Abschnitt 3.3 erläutert wird. Somit stellt der Transformationsschritt die wichtigste Komponente für die Integration einer Datenquelle dar.

Integration Der Integrationsschritt für jede Datenquelle dient letztendlich der Zusammenführung des im Transformationsschritt erzeugten Weltausschnittes mit dem bereits vorhandenen Weltausschnitt. Dabei werden bereits existierende Begriffe vereinigt und die neu erzeugten Beziehungen übernommen. Die Zusammenführung der Begriffe erfolgt über die Verknüpfung des existierenden Begriffes mit dem neu erzeugten Kontext, den die Datenquelle zu einem Begriff liefert.

3.2.2 Initiale Erzeugung des Weltausschnittes

Der erste Schritt der Link Discovery besteht in der Auswahl einer geeigneten Datenquelle für die initiale Erzeugung des Weltausschnittes. In dieser Arbeit ist diese Datenquelle das Tagging–System von Spreadshirt (siehe Abschnitt 2.2.2).

Die Auswahl der Datenquelle richtet sich im wesentlichen danach, ob der Kontext, den die Datenquelle potenziell zu Begriffen liefern kann, für die Link Discovery geeignet ist. Der Kontext sollte außerdem für die geplante Anwendung der Ergebnisse der Link Discovery relevant sein.

Nach der Auswahl einer geeigneten Quelle werden die in Abschnitt 3.2.1 beschriebenen Schritte zur Integration durchgeführt. Der Schritt der Integration ist trivial, da zu diesem Zeitpunkt noch keine Daten im Weltausschnitt vorhanden sind.

3.2.3 Anreicherung des Weltausschnittes

Nach der initialen Erzeugung des Weltausschnittes kann dieser mit beliebig vielen Anreicherungsschritten ergänzt werden. Unter Anreicherung wird die Erzeugung neuer Begriffe, Kontexte oder Zusammenhänge verstanden.

Das Hinzufügen neuer Begriffe erweitert das Vokabular des Weltausschnittes. Somit können bei der Benutzung der Daten zu einer größeren Menge von Begriffen Zusammenhänge gefunden werden. Die Erzeugung von Kontexten zu vorhandenen oder neuen Begriffen erweitert das Wissen über die Benutzung eines Begriffes innerhalb einer bestimmten Anwendungsdomäne. Die Anreicherung des Weltausschnittes mit neuen Zusammenhängen ermöglicht einerseits das Finden von relevanten Nachbarn eines Begriffes, erfordert andererseits jedoch, abhängig von der Anwendung, auch eine andere Priorisierung der Zusammenhangstypen.

Grundsätzlich kann die Anreicherung des Weltausschnittes auf zwei Arten erfolgen. Dies ist zum Einen die Anreicherung durch das Mining der bereits vorhandenen Daten, zum Anderen die Anreicherung durch Integration einer weiteren Datenquelle.

Anreicherung durch Mining vorhandener Daten

Die Erzeugung neuer Begriffe und Zusammenhänge kann aus den vorhandenen Daten mittels Methoden des Data Minings vorgenommen werden. Dazu werden die bereits im Weltausschnitt vorhandenen Begriffe mit ihren Kontexten und Beziehungen analysiert, um bisher unbekannte Zusammenhänge zu finden.

Beispiele für anwendbare Methoden sind hierbei Assoziationsanalyse [Tan05, S. 328f.] oder Clusteranalyse [HKP12, S. 443f.], um bestimmte, bereits in den Daten vorhandene, aber nicht explizit abgebildete Zusammenhänge zu ermitteln. Jedoch können auch einfachere Methoden wie die Zerlegung von Wortgruppen in Einzelwörter (siehe Abschnitt 4.3) oder das Einfügen von bisher nur transitiv, also nur über mehrere Begriffe, vorhandenen Beziehungen erfolgreich sein, um neue Begriffe und Beziehungen in den Datenbestand einzuführen.

Anreicherung durch Integration externer Datenquellen

Sofern weitere Datenquellen verfügbar sind, stellt deren Integration einen weiteren erfolgversprechenden Weg dar, um die Daten anzureichern. Neue Datenquellen beschreiben immer einen neuen Kontext, in dem Begriffe genutzt werden. Ist der Kontext dieser Begriffe für die spätere Anwendung relevant, ist die Integration der jeweiligen Datenquelle zu Zwecken der Link Discovery von großem Interesse.

Um die zusätzliche Datenquelle zur Link Discovery zu nutzen, werden die Schritte, die schon zur initialen Erzeugung des Weltausschnittes zum Einsatz kamen, genutzt (siehe Abschnitt 3.2.1). Dazu werden im Transformationsschritt Daten erzeugt, die dem Modell des Weltausschnittes entsprechen. Im Integrationsschritt werden sie mit den bereits vorhandenen Daten zusammengeführt.

3.2.4 Priorisierung von Beziehungen

Ziel des Priorisierungsschrittes der Link Discovery ist das Finden einer Gewichtung der Zusammenhangstypen, die für einen Anwendungsfall relevante Nachbarn zu einem Begriff liefert. Die Relevanz kann jedoch nur von einem Benutzer bewertet werden. Abbildung 3.3 stellt den Priorisierungsprozess als FMC–Petri–Netz dar.

Grundsätzlich kann nicht davon ausgegangen werden, dass eine Priorisierung für alle im Weltausschnitt gespeicherten Begriffe relevante Nachbarn liefert. Daher sollte die Priorisierung stichprobenhaft für einzelne Begriffe durchgeführt werden, um dann möglicherweise eine global gute Ergebnisse liefernde Priorisierung zu ermitteln.

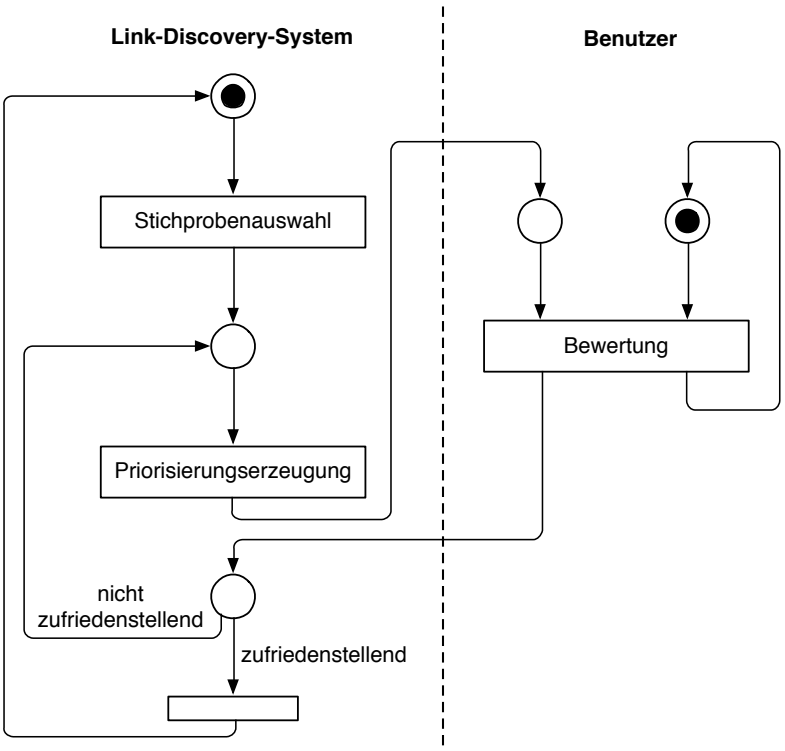


Abbildung 3.3: FMC–Petri–Netz der Priorisierung

Der Priorisierungsprozess beginnt mit der Auswahl einer Stichprobe anhand geeigneter Kriterien für das Anwendungsszenario. Beispielhaft für Kriterien sind externe Faktoren wie die Popularität des Begriffes in der Anwendungsdomäne oder im Weltausschnitt gespeicherte Faktoren wie die Anzahl der Zusammenhänge, die ein Begriff besitzt. Wird der Prozess mit mehreren Stichproben durchgeführt, sollte auf eine möglichst breite Streuung des jeweiligen Kriteriums geachtet werden, um die Güte der Priorisierungen abhängig vom Begriff beurteilen zu können.

Nach der Auswahl der Stichprobe wird vom Link–Discovery–System eine Priorisierung erzeugt. Wie diese Erzeugung konkret implementiert ist, hängt von der Anwendung ab. Im Rahmen dieser Arbeit wurden evolutionäre Algorithmen (siehe Abschnitt 3.6) gewählt. Die erzeugte Priorisierung wird auf den Begriff angewendet und von einem Benutzer bewertet. Der Benutzer sollte Wissen über die Anwendungsdomäne besitzen.

Ist die Bewertung der Priorisierung positiv, ist der Priorisierungsprozess beendet. Bei nicht zufriedenstellendem Ergebnis wird die Erzeugung einer neuen Priorisierung und die anschließende Bewertung wiederholt. Stellt sich auch nach einer im Voraus gewählten Anzahl

von Iterationen dieser Art kein zufriedenstellendes Ergebnis ein, so sollte die Priorisierung abgebrochen und die Gründe für das Fehlschlagen analysiert werden. Diese können beispielsweise in einer schlechten Qualität der Beziehungen des Weltausschnittes, in einer unpassenden Stichprobenauswahl oder fehlendem Wissen des Benutzers gefunden werden.

3.3 Kookkurrenz als Mittel zur Beziehungserzeugung

Im Transformationsschritt der Link Discovery (Abschnitt 3.2.1) wird eine Methode benötigt, um aus dem Kontext von Begriffen Beziehungen zwischen eben jenen zu berechnen. Eine der möglichen Methoden ist die Berechnung von *Kookkurrenz*. Diese wird im folgenden Abschnitt näher erläutert.

3.3.1 Grundlagen von Kookkurrenz

Um gewichtete inhaltliche Beziehungen zwischen Begriffen herstellen zu können, wird eine Definition von *Ähnlichkeit* benötigt. Diese lässt sich auf vielfältige Arten bestimmen.

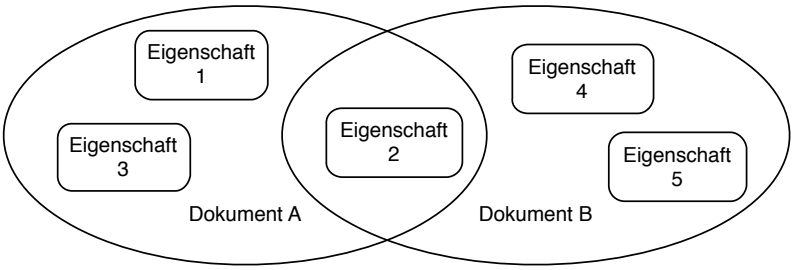


Abbildung 3.4: Repräsentation von Dokumenten als Mengen von Eigenschaften

Die Ähnlichkeit zwischen zwei Dokumenten kann grundsätzlich nach Tversky [Tve77] definiert werden. Dieser Definition liegt zu Grunde, dass sich die Dokumente als Mengen von Eigenschaften beschreiben lassen. Im Gegensatz zu anderen Ähnlichkeitsmodellen hängt die Ähnlichkeit nicht nur von den gemeinsamen Eigenschaften der Dokumente ab, sondern auch von den Eigenschaften, die die Dokumente allein besitzen. Diese Definition von Ähnlichkeit wird in Abbildung 3.4 veranschaulicht.

Somit lässt sich Ähnlichkeit $s(A, B)$ zwischen den Dokumenten, die durch die Mengen A und B dargestellt werden, durch $s(A, B) = F(A \cap B, A - B, B - A)$ definieren.

Die Gestaltung der Funktion s und die Auswahl der für die Ähnlichkeitsberechnung genutzten Eigenschaften der Dokumente hängt stark von der Anwendung ab. Somit beschreibt beispielsweise die Levenshtein–Distanz [Lev66] die Ähnlichkeit zweier Zeichenketten durch die minimale Menge von Einfüge-, Lösch- und Ersetzungsoperationen, die nötig sind, um eine Zeichenkette in die andere umzuwandeln. In Ontologien und Taxonomien kann die Ähnlichkeit von Begriffen mittels der Knoten- oder Kanteneigenschaften berechnet werden. Beispiele hierfür sind die Ähnlichkeit in Ontologien nach Resnik [Res95] und Pekar und Staab [PS02]. In der Bildverarbeitung können ebenfalls merkmalsbasierte Ähnlichkeitsmaße eingesetzt werden. Beispielsweise beschreiben Omer und Werman [OW06] ein Ähnlichkeitsmaß auf Basis von Clusteringalgorithmen, die auf Rastergrafiken angewandt werden.

Im Rahmen dieser Arbeit wird ein Ähnlichkeitsmaß gesucht, das anhand der Eigenschaften von Begriffen eine Distanz zwischen diesen berechnet. Da eine inhaltliche Ähnlichkeit gesucht wird, spielen linguistische Ähnlichkeitsmaße wie die Levenshtein–Distanz eine untergeordnete Rolle. Zu Beginn bestehen keinerlei Verbindungen zwischen den Begriffen, so dass keine Ähnlichkeitsmaße für Ontologien eingesetzt werden können. Somit bietet sich die Wahl eines Ähnlichkeitsmaßes an, das den Kontext, in dem die Begriffe im Quellsystem verwendet werden, berücksichtigt.

In Abschnitt 2.2.2 wurde das für diese Arbeit verfügbare Tagging–System beschrieben. In diesem System besitzen die Tags wenig Kontext. Die einzig verfügbare Information ist, an welche Dokumente die Tags vergeben wurden.

Wenn mehrere Begriffe pro Dokument verwendet werden, wird damit ein Zusammenhang zwischen den Begriffen beschrieben. Dieser Zusammenhang lässt sich mit dem Ähnlichkeitsmaß *Kookkurrenz* messen. Kookkurrenzmaße beschreiben, wie oft Begriffe gemeinsam verwendet werden [ZZ11, S. 21]. Dies wird ins Verhältnis zum einzelnen Auftreten der Begriffe gesetzt und genügt somit der Definition von Ähnlichkeit in Abschnitt 3.3.1.

Hierzu muss angemerkt werden, dass die Ähnlichkeit mittels Kookkurrenz nicht zwingend eine Ähnlichkeit der den Begriffen zu Grunde liegenden Konzepte darstellt. Die Verwendung von Kookkurrenz als Ähnlichkeitsmaß beruht allein auf der Annahme, dass Menschen zur Beschreibung von gleichen Inhalten die gleichen Begriffe benutzen. Diese An-

nahme muss im Laufe der Evaluierung der Qualität der Ergebnisse validiert werden, liegt jedoch nicht im Fokus dieser Arbeit.

3.3.2 Maße für Kookkurrenz

Werden die Objekte, zwischen denen die Ähnlichkeit berechnet werden soll, als Mengen von Eigenschaften definiert, bieten sich die üblichen Kennzahlen für die Ähnlichkeiten von Mengen an. Ein Begriff kann als Menge der Dokumente, für die er als Beschreibung verwendet wurde, definiert werden.

Um die Ähnlichkeit zwischen zwei Begriffen zu ermitteln, lassen sich die Vereinigungsmenge, Schnittmenge und Kreuzprodukte der jeweiligen Mengen bilden, die die Begriffe repräsentieren. Ist A die Menge der Dokumente, die mit einem Begriff a versehen wurden, B die Menge der Dokumente mit einem Begriff b, so ergeben sich die Mengen:

- $A \cap B$, alle Dokumente die mit a und b versehen wurden
- $A \cup B$, alle Dokumente die mit a oder b versehen wurden
- $A \times B$, alle Dokumentenpaare, die sich aus den Mengen A und B bilden lassen

Die Mächtigkeiten dieser Mengen können dann zur Berechnung verschiedener Ähnlichkeitsmaße verwendet werden. Drei der üblichsten Maße wurden im Rahmen dieser Arbeit verwendet und werden im Folgenden genannt.

Sørensen–Dice

Der Sørensen–Dice–Koeffizient [Sør48] [Dic45], oft auch nur Dice–Koeffizient, entstand ursprünglich in der Biologie und wurde verwendet, um die Ähnlichkeit zwischen Proben zu berechnen. Heute findet er allgemeine Anwendung im Data Mining. Er ist definiert durch:

$$\delta_{Dice}(a, b) = \frac{2|A \cap B|}{|A| + |B|}$$

Der Wertebereich W des Koeffizienten wird mit $W = [0, 1] \in \mathbb{R}$ angegeben.

Jaccard

Der Jaccard–Index [Jac12] wurde ursprünglich mit dem gleichen Zweck wie der Dice–Koeffizient verwendet. Sein Wertebereich wird ebenfalls mit $W = [0, 1] \in \mathbb{R}$ angegeben und er ist definiert durch:

$$\delta_{Jaccard}(a, b) = \frac{|A \cap B|}{|A \cup B|}$$

Kosinus

Die Kosinus-Ähnlichkeit [HKP12] ist ursprünglich ein Maß für die Ähnlichkeit zweier Vektoren. Sie ist eine Maßzahl dafür, ob die Vektoren ungefähr in die gleiche Richtung zeigen. Sie kann jedoch genauso auf Mengen angewendet werden, da das Vorhandensein der Elemente in der Menge auch durch einen Vektor in einem n-dimensionalen Raum dargestellt werden kann, wobei n die Anzahl aller möglichen Eigenschaften ist. Der Wertebereich der Kosinus-Ähnlichkeit ist ebenfalls mit $W = [0, 1] \in \mathbb{R}$ angegeben. Sie ist auf den in Abschnitt 3.3.2 definierten Mengen definiert durch:

$$\delta_{Cosine}(a, b) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}$$

Nachdem die Ähnlichkeit mittels Kookkurrenz und die entsprechenden Maße vorgestellt wurden, wird im nächsten Abschnitt die Berechnung und damit verbundene Komplexität diskutiert.

3.3.3 Berechnung von Kookkurrenz

Um die in Abschnitt 3.3.2 genannten Kookkurrenzmaße zu berechnen, wird das Kreuzprodukt aller Begriffe benötigt. Dabei muss für jedes Paar von Begriffen die Häufigkeit gezählt werden, wie oft die Begriffe gemeinsam zur Beschreibung von Dokumenten verwendet wurden. Diese Häufigkeit beschreibt die Mächtigkeit der Mengen $A \cap B$. Außerdem muss gezählt werden, wie oft jeder Begriff insgesamt verwendet wird, um die Mächtigkeit der Mengen $A, B, \ldots$ zu bestimmen. Danach können über die genannten Formeln die

```
var occurences = {};

foreach (term in terms) {
    occurences[term] = countOccurrences(term);
}

foreach (termA in terms) {
    foreach (termB in terms) {
        var ab = countCoOccurences(termA, termB);
    }
    var diceAB = dice(occurences[termA], occurences[termB], ab);
    var jaccardAB = jaccard(occurences[termA], occurences[termB], ab);
    var cosineAB = cosine(occurences[termA], occurences[termB], ab);
}
```

Listing 3.1: Kookkurrenzberechnung

Kookkurrenzmaße berechnet werden. In Listing 3.1 ist die Berechnung als Pseudo–Code dargestellt.

Der Aufwand, um die Kookkurrenzmaße für alle Paare von Begriffen zu berechnen, hängt von der Anzahl der Begriffe, Dokumente und Verwendungen ab. Beträgt die Anzahl der Begriffe n und die Anzahl der Dokumente d, so ergibt sich für den Fall, dass jeder Begriff an jedes Dokument vergeben wurde eine Laufzeit von $O(d * n^2)$. Wurden keine Begriffe mit Dokumenten verknüpft, beträgt die Laufzeit $\Theta(d)$. Die reale Laufzeit der Ähnlichkeitsberechnung liegt daher zwischen diesen Schranken.

Es ist absehbar, dass der Rechenaufwand mit wachsender Datenmenge stark ansteigt. Somit scheint es ratsam, nach Optimierungen zu suchen, um die Rechenzeit zu verringern. Da sich die Anzahl der Berechnungen nicht vermindern lässt, kann eine Verkürzung der Rechenzeit nur durch Parallelisierung erreicht werden. Eine mögliche Umsetzung der parallelen Berechnung der Kookkurrenz mittels des Programmiermodells MapReduce wird in Abschnitt 5.3.3 erläutert.

3.4 Graphen als Beschreibungsmittel des Weltausschnittes

Nachdem in Abschnitt 3.1 der zur Link Discovery verwendete Weltausschnitt modelliert wurde, wird zur Umsetzung eine Datenstruktur benötigt, die diesen Ausschnitt konkret abbildet. Da im Wesentlichen Objekte, zwischen denen Verbindungen bestehen, abgebildet

werden, bietet sich die Verwendung eines Graphen an. Die Grundlagen von Graphen sowie die konkrete Abbildung des Weltausschnittes auf eine Graph–Datenstruktur werden in den folgenden Abschnitten erläutert.

3.4.1 Grundlagen

Ein *ungerichteter Graph* ist definiert durch ein Paar $G = (V, E)$ von Mengen, für die gilt: $E \subseteq [V]^2$. Dies bedeutet, dass alle Elemente aus E 2-elementige Teilmengen von V sind [Die12, S. 2]. Die Menge V repräsentiert die *Knoten* und die Menge E die *Kanten* des Graphen. Eine Kante stellt eine Verbindung von zwei Knoten dar. Zwei Knoten werden als *Nachbarn* bezeichnet, wenn zwischen ihnen eine Kante existiert. Abbildung 3.5 zeigt ein Beispiel für einen ungerichteten Graphen.

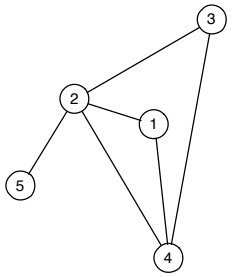


Abbildung 3.5: Ungerichteter Graph

Ein *gerichteter Graph* ist ein Graph, der neben den Mengen V und E die zwei Abbildungen $quelle : E \to V$ und $ziel : E \to V$ enthält [Die12, S. 25]. Diese weisen jeder Kante e einen Quell- und Zielknoten zu. Die Kante ist somit von $quelle(e)$ nach $ziel(e)$ gerichtet. Abbildung 3.6 zeigt einen gerichteten Graphen.

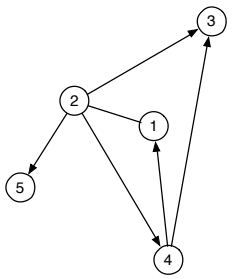


Abbildung 3.6: Gerichteter Graph

Als *Multigraph* wird schließlich ein Graph bezeichnet, bei dem zwischen zwei Knoten mehrere Kanten bestehen [Die12, S. 25]. Sind diese gerichtet, spricht man vom einen *gerichteten Multigraphen*. Abbildung 3.7 illustriert einen solchen gerichteten Multigraphen.

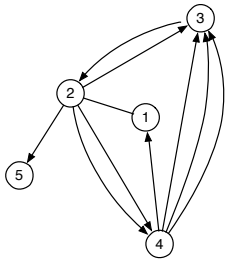


Abbildung 3.7: Gerichteter Multigraph

Die Knoten und Kanten eines Graphen sind Objekte mit beliebigen weiteren Eigenschaften. Kanten besitzen üblicherweise ein Gewicht, das ihre Wichtigkeit oder Kosten im Anwendungskontext des Graphen angibt.

Nachdem Graphen grundsätzlich erläutert wurden, beschäftigt sich der folgende Abschnitt mit der konkreten Umsetzung des Modells des Weltausschnittes auf eine solche Graphen–Struktur.

3.4.2 Graphenrepräsentation des Weltausschnittes

Um den Weltausschnitt in Abschnitt 3.1 in eine Graphenform zur transformieren, muss zunächst modelliert werden, welche Objekte die Knoten und Kanten des Graphen repräsentieren.

Die Knoten repräsentieren die Begriffe, zwischen denen durch die Link Discovery Zusammenhänge hergestellt werden sollen. Sie enthalten als benötigte Attribute eine Zeichenkette und ein Attribut Sprache. Die Kombination dieser beiden Attribute ist innerhalb der Knotenmenge eindeutig, um je verwendetem Begriff in einer Sprache genau einen Knoten zu erhalten.

Der Kontext der Begriffe wird über Eigenschaften von Knoten abgebildet. Dabei kann ein Begriff beliebig viele Eigenschaften besitzen. Da die Kontexte der Begriffe aus jeweils einer Datenquelle stammen, bietet es sich an, pro Datenquelle eine Entität für die kontextuellen Eigenschaften, die diese Datenquelle liefert, zu definieren.

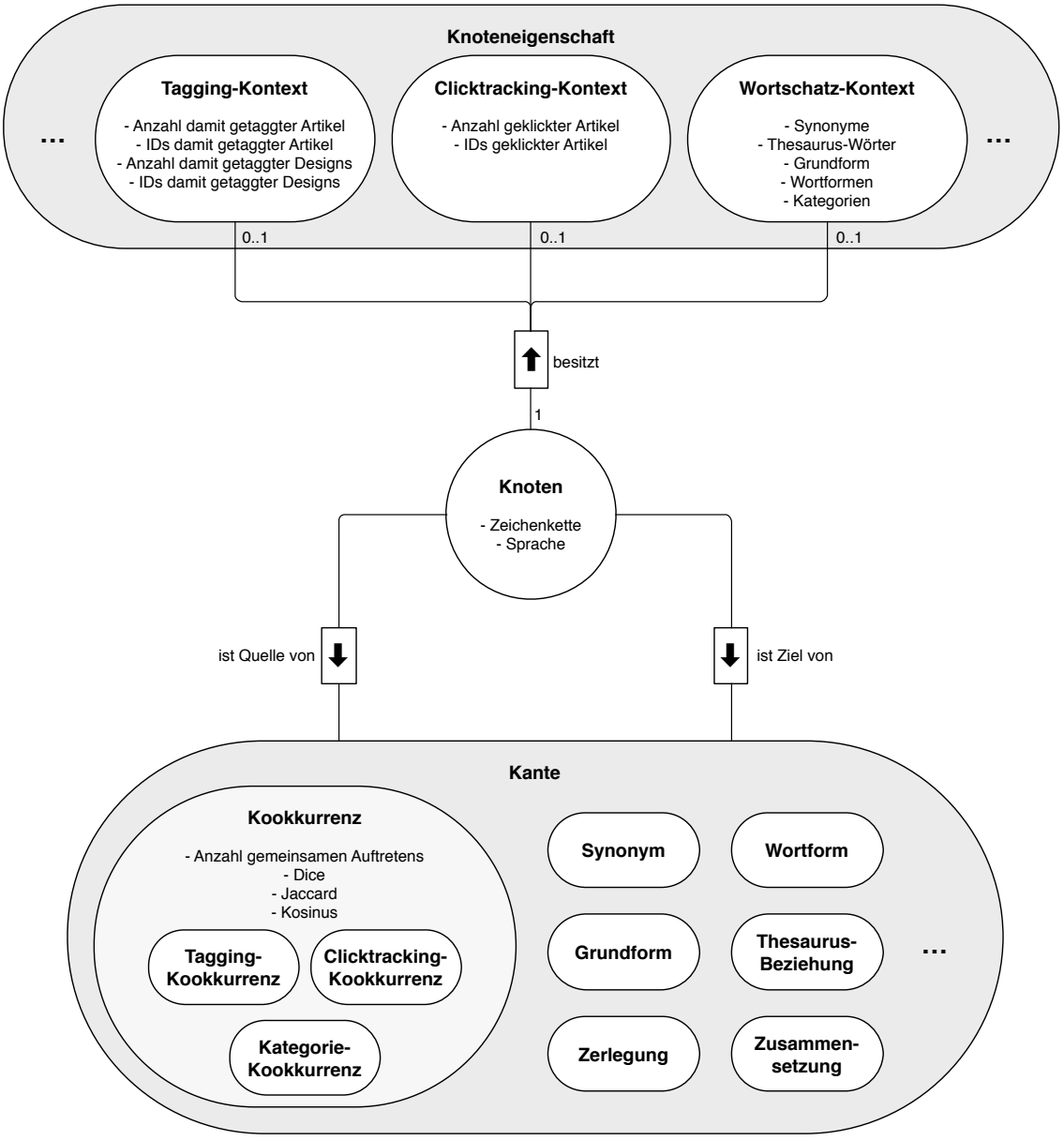


Abbildung 3.8: FMC–Entity–Relationship–Diagramm der Graphenrepräsentation

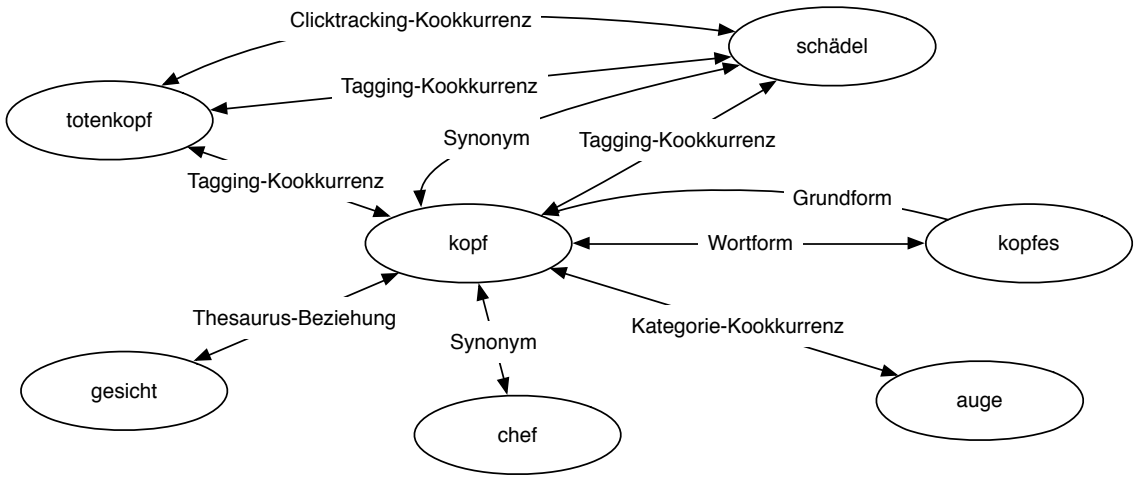


Abbildung 3.9: Beispiel–Graphausschnitt für das Ergebnis der Link Discovery

Eine Kante repräsentiert einen irgendwie gearteten Zusammenhang zwischen zwei Begriffen. Sie besitzt einen *Typ*, der die Art des Zusammenhangs spezifiziert, sowie je einen Quell- und Zielknoten. Zusätzlich kann sie weitere Attribute besitzen. Bei Kanten, die durch Kookkurrenzberechnung (siehe Abschnitt 3.3) erzeugt wurden, sind dies beispielsweise die Anzahl des gemeinsamen Auftretens zweier Begriffe und die berechneten Kookkurrenzmaße. Im Rahmen dieser Arbeit wurden keine weiteren Attribute für Kanten benötigt, diese sind generell jedoch denkbar.

Zwischen zwei Knoten können beliebig viele, je Typ jedoch höchstens eine, Kanten existieren. Daher handelt es sich bei dem Graphen des Weltausschnittes um einen gerichteten Multigraphen (siehe Abschnitt 3.4.1).

Das resultierende komplette Modell des Graphen ist in Abbildung 3.8 dargestellt. Dieses beinhaltet sämtliche zum Zeitpunkt der Bearbeitung bekannten und integrierten Datenquellen und die damit verbundenen Kontexte und Kantentypen. Die Datenquelle des Tagging–Systems und der damit verbundene Kontext wird in Abschnitt 2.2.2 näher beschrieben. Das Clicktracking–System wird in Abschnitt 4.2 und der Wortschatz der Universität Leipzig in Abschnitt 4.4 näher beleuchtet. Details über die Erzeugung der entsprechenden Kantentypen werden in Kapitel 4 erläutert.

Um die Graphenrepräsentation des modellierten Weltausschnittes anschaulicher darzustellen, ist in Abbildung 3.9 ein beispielhafter Ausschnitt des nach Durchführung der Link Discovery entstehenden Graphen abgebildet. Auf Darstellung der Kontexte und Kantenattribute wurde aus Platzgründen verzichtet.

Nach der Definition der verwendeten Graphenstruktur werden im folgenden Abschnitt die möglichen Datenquellen diskutiert.

3.5 Datenquellen zur Anreicherung

Der initiale Weltausschnitt wird im Rahmen dieser Arbeit aus den Daten des Tagging–Systems von Spreadshirt erzeugt. Wie in Abschnitt 3.2.3 bereits einführend erläutert, kann danach zur Anreicherung die Integration weiterer Datenquellen erfolgen. Diese Datenquellen können eine andere Sicht auf die Begriffe des Weltausschnittes liefern und somit zur Erzeugung neuer Beziehungen nützlich sein. In diesem Abschnitt werden mögliche Datenquellen genannt und erläutert sowie die letztendlich verwendeten Datenquellen näher beschrieben.

3.5.1 Lexikalische Quellen

Da die Hauptentität des Weltausschnittes Begriffe einer Sprache darstellt, liegt es nahe, auf lexikalische Quellen zurückzugreifen. Diese existieren in unterschiedlichen Ausprägungen, haben jedoch alle gemeinsam, dass sie Wörter beschreiben und in Beziehung zu anderen Wörtern setzen. Damit eignen sie sich gut zur Integration in den Weltausschnitt. Einige der üblichen Formen von lexikalischen Quellen werden im folgenden genannt und erläutert.

Ein *Wörterbuch* ist ein Nachschlagewerk, das erklärende Informationen über Wörter enthält [HK03]. Dabei kann es sich um Sach- oder Sprachwissen handeln, das einerseits die inhaltliche Bedeutung eines Wortes oder andererseits die linguistischen und grammatikalischen Eigenschaften beschreibt. Beispiele für Wörterbücher sind der *Duden* [SI13] oder das *Oxford English Dictionary* [Ste10].

Als *Thesaurus* wird ein kontrolliertes Vokabular von Wörtern bezeichnet, die nach ihrer inhaltlichen Bedeutung geordnet sind [AC04, S. 2]. Er stellt demnach eine Sammlung von *Synonymen*, also Wörtern mit gleicher Bedeutung, dar und enthält keine Definitionen der Wörter. Oft enthalten Thesauri ebenfalls Ober- und Unterbegriffe und beschreiben somit

Taxonomien. Beispiele für Thesauri sind OpenThesaurus [Nab] für die deutsche und Thesaurus.com [Dic] für die englische Sprache. Thesauri sind eine spezielle Form von Wörterbüchern.

Wortschätze sind eine Mischform aus Sprach- und Sachwörterbüchern. Sie enthalten zu möglichst vielen im Gebrauch befindlichen Wörtern einer Sprache Informationen über Verwendung, Bedeutung, Über- und Unterbegriffe, grammatikalische Eigenschaften und mehr. Für die englische Sprache existiert die *WordNet*–Datenbank [Wor; Fel98], die Synonyme, Wortformen, Bedeutungen, Über- und Unterbegriffe, Beispiele und mehr für englische Wörter katalogisiert. Die Universität Leipzig betreibt mit dem *Deutschen Wortschatz* ein ähnliches Projekt [WSL] für die deutsche Sprache.

Generell sind lexikalische Quellen geeignet, um den Weltausschnitt mit Allgemeinwissen über die enthaltenen Begriffe anzureichern. Sie bieten allgemeine Informationen über Wörter und sind somit in vielen Anwendungsszenarien nutzbar.

3.5.2 Clicktracking–System

Die Aufgabe eines *Clicktracking–Systems* besteht im Wesentlichen darin, die Klicks von Benutzern auf Hyperlinks einer Website aufzuzeichnen. Dabei wird für gewöhnlich auch aufgezeichnet, in welchem Kontext der Klick statt fand. Der Kontext enthält beispielsweise die Suchbegriffe, die der Benutzer auf der Website oder einer externen Suchmaschine eingegeben hat, die Position des geklickten Elementes auf der Website und weitere Metadaten.

Besonders in Verbindung mit eingegebenen Suchbegriffen kann Clicktracking ein hohes Potenzial zur Anreicherung des Weltausschnittes bieten. Mit jedem Klick auf einen Inhalt der Website wird dem Suchbegriff weiterer Kontext innerhalb einer bestimmten Anwendungsdomäne verliehen. Werden gleiche Inhalte zu verschiedenen Suchbegriffen angeklickt, lassen sich Kookkurrenzen zwischen den Suchbegriffen berechnen (siehe 3.3) und somit Zusammenhänge extrahieren.

3.5.3 Verwendete Datenquellen

Im Rahmen dieser Arbeit wurden für die Anreicherung mittels Integration weiterer Datenquellen das Clicktracking–System von Spreadshirt und das Wortschatz–Projekt der Universität Leipzig verwendet.

Das Clicktracking–System von Spreadshirt zeichnet auf, auf welche Suchergebnisse die Benutzer auf Suchergebnisseiten klicken und eignet sich damit zur Kookkurrenzberechnung zwischen den verwendeten Suchbegriffen. Die Struktur und der genaue Ablauf der Link Discovery aus diesen Daten wird in Abschnitt 4.2 detailliert beschrieben.

Der Wortschatz der Universität Leipzig [WSL] wurde ausgewählt, da er frei verfügbar ist und zu deutschen Wörtern viele inhaltliche, linguistische und grammatikalische Informationen bereit stellt. Die Informationen werden zu einem großen Teil aus automatisch analysierten deutschen Texten generiert [Hey11].

Zu jedem Begriff stellt der Wortschatz die folgenden Informationen bereit:

- Grundform des Wortes
- Wortformen
- Kookkurrenzen in den analysierten Texten
- Kategorien, in die das Wort eingeordnet werden kann
- Synonyme
- Thesaurus–Beziehungen
- Häufigkeit des Auftretens
- Sätze, die das Wort enthalten

Die Informationen über Grundform, Wortformen, Thesaurus–Beziehungen, Synonyme und Kategorien wurden im Rahmen dieser Arbeit ausgewertet und entsprechen den in Abbildung 3.8 gezeigten Kantentypen. Die Durchführung der Link Discovery anhand dieser Daten wird ausführlich in Abschnitt 4.4 beschrieben.

3.6 Evolutionäre Algorithmen als Mittel zur Priorisierung

Um die in Abschnitt 3.2.4 beschriebene Priorisierung der Beziehungen durchführen zu können, wird ein Verfahren zur Erzeugung der Priorisierungen benötigt. Im Rahmen dieser Arbeit wurden dazu *evolutionäre Algorithmen* gewählt. Die Grundprinzipien sowie die Anwendung dieser Algorithmen zur Priorisierung werden im folgenden Abschnitt erläutert.

3.6.1 Grundlagen

Als evolutionäre Algorithmen wird eine Klasse von Optimierungsverfahren bezeichnet, deren Funktionsweise an die Evolution natürlicher Lebewesen angelehnt ist. Sie versuchen, Probleme durch die Simulation von Evolution mittels der Auswahl der erfolgreichsten Individuen zu lösen. Dabei kommen ebenfalls aus der Biologie entlehnte Mechanismen wie Mutation und Rekombination zum Einsatz, um iterativ eine Population von Lösungskandidaten zu verbessern [Wei08]. Es handelt sich um heuristische Algorithmen, die das Finden einer optimalen Lösung nicht garantieren können [GKK04, S. 12].

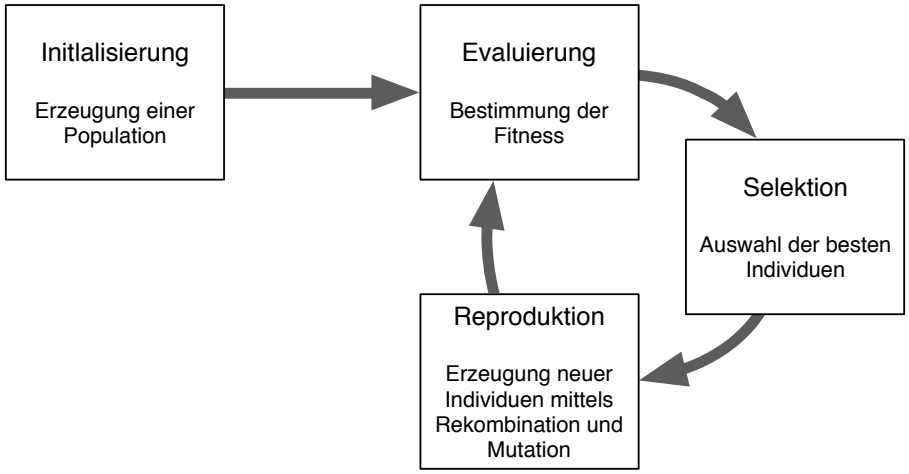


Abbildung 3.10: Ablauf evolutionärer Algorithmen

Grundsätzlich folgt das Vorgehen einem Kreislauf mit den Komponenten *Evaluierung*, *Selektion* und *Reproduktion*. Nach Generierung einer anfänglichen Population (*Initialisierung*) wird dieser Kreislauf so lange durchlaufen, bis ein vorher definiertes Abbruchkriterium eintritt. Ein Durchlauf wird als *Generation* bezeichnet. Das Abbruchkriterium kann

beispielsweise ein bestimmter Schwellwert für die Güte der Lösung oder eine feste Anzahl von Generationen sein. Der beschriebene Ablauf ist in Abbildung 3.10 dargestellt.

Die einzelnen Komponenten eines evolutionären Algorithmus werden im Folgenden beschrieben. Die Definitionen folgen im Wesentlichen denen von Weise [Wei08].

Initialisierung Die Population P stellt eine Menge von Lösungskandidaten dar. Ein Lösungskandidat i wird als *Individuum* bezeichnet und durch seinen *Genotyp* repräsentiert. Der Genotyp ist die kodierte Repräsentation aller Variablen, die den Lösungskandidaten spezifizieren. Die Variablen werden *Gene* genannt. Während der Initialisierung werden Lösungskandidaten erzeugt, die die Startpopulation des evolutionären Algorithmus bilden. Die Gene jedes Individuums werden üblicherweise zufällig gewählt.

Evaluierung Der Evaluierungsschritt dient zur Bestimmung der *Fitness* der Individuen, die noch in der Population enthalten sind. Die Fitness stellt einen Wert dar, der die Güte der durch das Individuum repräsentierten Lösung bezüglich der Problemstellung beschreibt. Die Fitness eines Individuums i kann, je nach Optimierungsproblem, entweder absolut oder bezüglich der anderen Individuen der Population P bestimmt werden. Somit lässt sich die Funktion zur Bestimmung der Fitness auf die Form $fitness(i, P)$ generalisieren.

Selektion Im Selektionsschritt werden die fittesten Individuen der Population P ausgewählt. Alle nicht ausgewählten Lösungskandidaten werden verworfen. Die Selektion kann als Funktion der Form $select(P, fitness, s)$ dargestellt werden, wobei s eine festgelegte Anzahl von Individuen darstellt, die ausgewählt werden sollen.

Reproduktion Die Reproduktion dient dazu, aus den im Selektionsschritt ausgewählten Individuen neue Lösungskandidaten zu erzeugen. Dabei werden üblicherweise die Operationen *Rekombination* und *Mutation* verwendet. Bei der Rekombination wird, analog zur Biologie, aus zwei Elternindividuen ein neues Kindindividuum erzeugt. Sie lässt sich als Funktion der Form $i_n = recombine(i_a, i_b)$ darstellen, wobei i_n das neue Individuum und i_a und i_b die Elternindividuen darstellen. Eine Mutation erzeugt ein neues Individuum

durch die Modifikation eines anderen und ist daher durch die Funktion $i_n = mutate(i_a)$ beschrieben.

In der Literatur [Wei07; Wei08; De 06] finden sich für Selektion, Mutation und Rekombination Standardverfahren, die in Hinblick auf das zu lösende Optimierungsproblem ausgewählt werden können. Die konkret implementierten Verfahren werden in Abschnitt 4.6.1 beschrieben.

Nachdem evolutionäre Algorithmen grundlegend beschrieben wurden, wird im nächsten Abschnitt erläutert, wie die Priorisierung mittels dieser Algorithmenklasse implementiert werden kann.

3.6.2 Anwendung zur Priorisierung

Mit Hilfe von evolutionären Algorithmen kann der Prozess der Priorisierung aus Abschnitt 3.2.4 genauer beschrieben werden. Abbildung 3.11 zeigt den Prozess mit den Komponenten evolutionärer Algorithmen.

Im Gegensatz zu Abbildung 3.3 ist zu erkennen, dass die Schritte zur Priorisierungserzeugung mit den Komponenten evolutionärer Algorithmen Initialisierung, Rekombination und Mutation ausgetauscht wurden. Die Selektion wird vom Benutzer vorgenommen, welcher anschließend bewertet, ob die ausgewählten Lösungskandidaten eine zufriedenstellende Priorisierung der Beziehungstypen darstellen.

Nach der Stichprobenauswahl wird für die Stichprobe eine Population von Lösungskandidaten erzeugt. Der Genotyp eines Individuums sollte pro Zusammenhangstyp ein Gewicht enthalten. Jedes Individuum stellt somit eine Gewichtung der Zusammenhangstypen dar.

Da der Algorithmus den Eingriff eines Benutzers erfordert, handelt es sich um einen *interaktiven evolutionären Algorithmus* [Tak01]. Die interaktive Komponente ist hierbei die Selektion, wodurch der Schritt der Evaluierung übersprungen werden kann. Wird die Selektion direkt vom Benutzer ausgeführt, erübrigt sich die Bestimmung eines Fitnesswertes.

Die Schritte der Rekombination und Mutation dienen zur Erzeugung neuer Lösungskandidaten auf Basis der vom Benutzer selektierten Priorisierungen. Sie sollten so gewählt

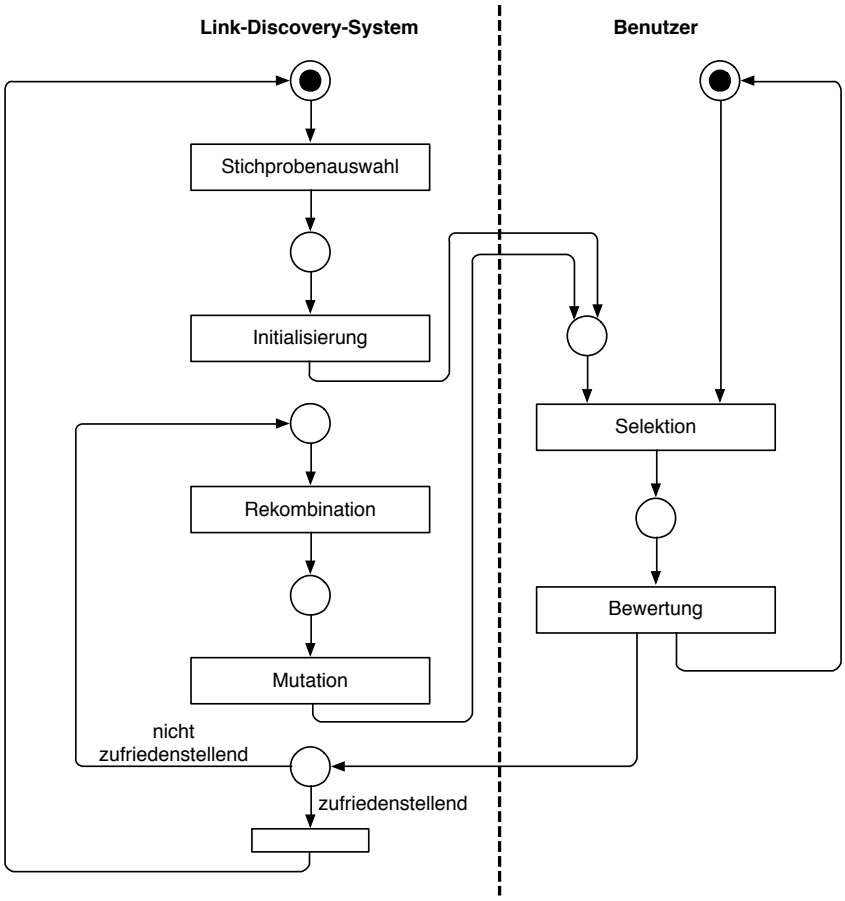


Abbildung 3.11: FMC–Petri–Netz der Priorisierung mittels evolutionärer Algorithmen

werden, dass genügend unterschiedliche Priorisierungen erzeugt werden, aber dennoch eine Verbesserung über die Generationen erkennbar ist.

Zusammenfassend stellen evolutionäre Algorithmen eine geeignete Methode dar, um die Priorisierung durchzuführen, da die Schritte der Priorisierungserzeugung darauf abgebildet werden können. Die Stichprobenauswahl, die konkret implementierten Komponenten, die Durchführung der Selektion und die Ergebnisse werden in Abschnitt 4.6.1 detailliert erläutert.

3.7 Zusammenfassung

In diesem Kapitel wurde das Framework zur Link Discovery erläutert. Dies beinhaltet den modellierten Weltausschnitt und dessen Umsetzung auf eine Graphen–Datenstruktur. Der

Prozess der Erzeugung, Anreicherung und Priorisierung der Beziehungen dieses Modells wurde definiert und beschrieben. Zur Beziehungserzeugung wurde Kookkurrenz vorgestellt, die gängigen Maße genannt und die Berechnung erläutert. Ferner wurden mögliche Datenquellen für die Anreicherung diskutiert sowie der Einsatz evolutionärer Algorithmen zur Beziehungspriorisierung beschrieben. Nachdem in diesem Kapitel das Link–Discovery–Framework umfassend behandelt wurde, beschreibt das nächste Kapitel die konkrete Durchführung der Link Discovery an Beispieldaten.

4 Link–Discovery–Durchführung

Dieses Kapitel beschäftigt sich mit den konkret durchgeführten Schritten zur Link Discovery an Beispieldaten. Das Vorgehen und die verwendete Modellierung des Weltausschnittes wurden umfassend in Kapitel 3 beschrieben. Im Folgenden wird die initiale Erstellung des Weltausschnittes aus den Daten des Tagging–Systems von Spreadshirt, die Anreicherung durch Clicktracking–Daten, Zerlegung von Wortgruppen in Einzelwörter und Integration des Wortschatzes der Universität Leipzig erläutert. Anschließend wird die konkrete Priorisierung der Beziehungen mittels evolutionärer Algorithmen dargestellt und die Ergebnisse der Link Discovery zusammenfassend ausgewertet.

4.1 Initiale Erstellung des Weltausschnittes aus Tagging–Daten

Die initiale Erstellung des Weltausschnittes stellt den ersten Schritt des in Abschnitt 3.2 definierten Link–Discovery–Prozesses dar. Die konkrete Datenquelle, die im Rahmen dieser Arbeit für diesen Schritt verwendet wurde, ist das Tagging–System des Unternehmens Spreadshirt, welches in Abschnitt 2.2.2 beschrieben wurde. Aus diesen Daten wird mittels Kookkurrenz (siehe Abschnitt 3.3) ein Graph erstellt, der der in Abschnitt 3.4.2 erläuterten Repräsentation des Weltausschnittes entspricht. Dieser stellt die Grundlage für alle folgenden Anreicherungsschritte dar. Um diesen initialen Graphen zu berechnen, sind die in Abschnitt 3.2.1 aufgeführten Schritte des Imports, der Bereinigung, der Reduktion, der Transformation und der Integration nötig. Die Umsetzung dieser Schritte wird im folgenden Abschnitt beschrieben.

4.1.1 Import

Die Daten liegen im Quellsystem, einer MySQL–Datenbank, in relationaler Form vor. Somit existieren, dem Datenmodell von Tagging–Systemen in Abschnitt 2.1.1 folgend, Tabellen für Benutzer, Tags, Dokumente und Taggings. Da der Inhalt der Dokumente sowie die Benutzer für die Kookkurrenzberechnung nicht relevant sind, genügt der Import der Tabellen *Tags* und *Taggings*. Das Datenmodell der für den Import benötigten Daten ist in Abbildung 4.1 dargestellt.

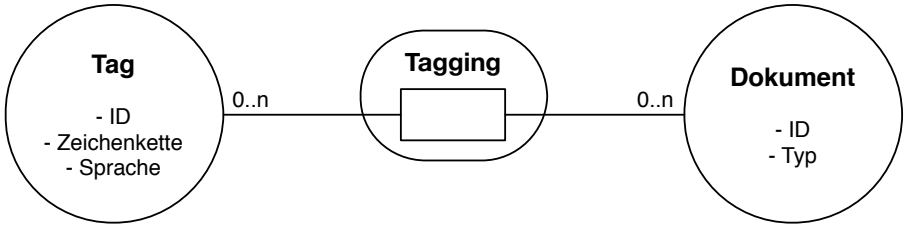


Abbildung 4.1: FMC–Entity–Relationship–Diagramm der Tagging–Quelldaten

Die Tags besitzen, neben einem eindeutigen Bezeichner, die Attribute *tag* für die Zeichenkette und *lang* für die Sprache des Tags. Listing 4.1 zeigt beispielhaft eine solche Tag–Entität.

```
1 {
2     "tag_id": 12345,
3     "tag": "segeln",
4     "lang": "de"
5 }
```

Listing 4.1: JSON–Beispiel für einen importierten Tag

Ein Tagging ist über den eindeutigen Bezeichner des Tags und den eindeutigen Bezeichner des getaggten Dokumentes definiert. Der Schlüssel des Dokumentes setzt sich aus den Attributen *object_type_id* und *object_id* zusammen, da die Dokumente auf der Plattform von Spreadshirt verschiedene Typen wie “Artikel” oder “Design” besitzen können. In Listing 4.2 ist ein Tagging beispielhaft dargestellt.

Gemäß des Mengengerüstes aus Abschnitt 2.2.4 stehen nach dem Import 2 072 079 Tags und 71 938 905 Taggings für die folgenden Integrationsschritte zur Verfügung.

```
{
    "object_id": 45678
    "object_type_id": 3,
    "tag_id": 12345
}
```

Listing 4.2: JSON–Beispiel für ein importiertes Tagging

4.1.2 Bereinigung

An den Tagging–Daten liegen die in Abschnitt 2.2.3 genannten Defekte in Hinblick auf die Datenqualität vor. Diese sollten in einem Bereinigungsschritt reduziert werden. Hierbei liegt das Hauptaugenmerk auf der Erkennung von Duplikaten und später nicht verwertbaren Zeichenketten. Alle durchgeführten Maßnahmen zur Bereinigung beziehen sich hierbei auf das Attribut *tag* eines Tag–Objektes, der Zeichenkette selbst.

In den unbereinigten importierten Daten existieren keine Duplikate in der Art, dass eine Paarung aus Zeichenkette und Sprache immer nur genau einmal in den Daten vorhanden ist. Jedoch enthalten viele der Tags nicht weiter verwertbare Zeichen wie nicht druckbare ASCII–Zeichen, Anführungszeichen, Satzzeichen, Sonderzeichen sowie überflüssige Leerzeichen am Anfang und Ende der Zeichenkette. Außerdem existiert in den importierten Daten eine Unterscheidung zwischen Groß- und Kleinschreibung. Diese Unterscheidung bringt im Kontext der Link Discovery keine Vorteile und kann folglich entfernt werden.

Rohdaten	Bereinigte Daten
\u0003\r\nregenbogen	regenbogen
RegenBogen	regenbogen
"Regenbogen"	regenbogen
regenbogen +einhorn	regenbogen einhorn
regenbogen	regenbogen
regenbogen	regenbogen

Tabelle 4.1: Beispiele für die Tag–Bereinigung

Somit besteht der Bereinigungsschritt darin, nicht verwertbare Zeichen zu entfernen und alle Großbuchstaben in Kleinbuchstaben umzuwandeln. Dadurch entstehen Duplikate, welche im darauf folgenden Reduktionsschritt zusammengeführt werden können. In Tabelle 4.1 sind einige Beispiele für die Bereinigungen aufgeführt. Dabei ist gut zu erkennen, dass durch die Bereinigungen Duplikate erzeugt werden.

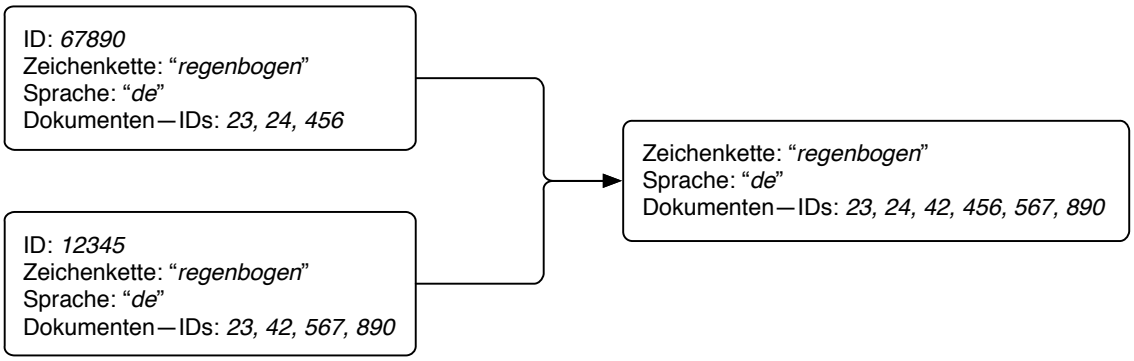


Abbildung 4.2: Beispiel für das Zusammenführen der bereinigten Tags

4.1.3 Reduktion

Der Reduktionsschritt dient zur Einschränkung der Gesamtdaten auf eine nützliche oder handhabbare Menge. Außerdem kann durch Reduktion auch die Datenqualität verbessert werden.

Im Fall der Tagging–Daten liegt das Hauptaugenmerk im Reduktionsschritt auf der Entfernung von Duplikaten, die bei der Bereinigung entstanden sind. Gleichzeitig muss sicher gestellt werden, dass keine Informationen über die Verwendung der Tags verloren gehen. Somit besteht die Duplikatentfernung der Tags im Zusammenführen von Datensätzen mit gleichen Zeichenketten und Sprachen. Gleichzeitig werden auch die Taggings zusammengeführt.

Werden die Verknüpfungen zweier Tags mit Dokumenten zusammengeführt, können wieder Duplikate entstehen. Diese müssen in diesem Fall entfernt werden, da ein Tag nicht mehrmals mit einem Dokument verknüpft werden kann. Das Zusammenführen von Tags ist exemplarisch in Abbildung 4.2 dargestellt.

Dabei findet ebenfalls eine Denormalisierung der Daten statt, da alle Dokumente, für die ein Tag vergeben werden, direkt mit in das Tag–Objekt gespeichert werden. Dies ist beispielhaft in Listing 4.3 abgebildet. Das Attribut *links* enthält alle Verknüpfungen des Tags mit Dokumenten.

Eine weitere im Rahmen dieser Arbeit unternommene Maßnahme zur Datenreduktion bestand darin, sich auf die Menge der Tags zu beschränken, deren Attribut $lang$ den Wert *de* besitzt. Praktisch handelt es sich um alle Tags, die als deutsch gekennzeichnet in der

```
{
    "string": "segeln"
    "language": "de",
    "links": [
        {
            "object_id": 45678,
            "object_type_id": 3
        },
        {
            "object_id": 98764,
            "object_type_id": 4
        },
        ...
    ]
}
```

Listing 4.3: JSON–Beispiel für die denormalisierten Tagging-Daten

Datenbank gespeichert sind. Diese Einschränkung wurde vorgenommen, um die zu verarbeitende Datenmenge überschaubar zu halten. Außerdem wird dadurch der nationale Kontext, in dem die Begriffe verwendet wurden, weitestgehend beibehalten.

Ein letzter Reduktionsschritt besteht in der Entfernung der Tags, deren Zeichenketten eine Länge von 1 besitzen, da in der deutschen Sprache keine einbuchstabigen Wörter existieren.

Nach der beschriebenen Reduktion befinden sich noch 314 351 Tags und 23 255 714 Taggings in der Datenbank. Dies entspricht einer Reduktion von ca. 68 Prozent gegenüber der importierten Menge von Objekten.

4.1.4 Transformation

Der Transformationsschritt stellt die Überführung der Daten in die in Abschnitt 3.4.2 beschriebene Repräsentation des Weltausschnittes dar. Für die Tagging–Daten bedeutet dies eine Umformung in eine Graphenform, wobei die Kanten des Graphen über Kookkurrenz ermittelt werden. Die genaue Umsetzung dieser Transformation mittels des MapReduce–Programmiermodells wird in Abschnitt 5.3.3 detailliert beschrieben.

Je Tag wird während der Transformation ein Knotenobjekt erzeugt. Dieses besitzt als benötigte Attribute die *Zeichenkette* und die *Sprache* des Tags, aus dem es erzeugt wurde und repräsentiert somit einen Begriff des Weltausschnittes. Außerdem wird ein Kontext dieses

Begriffes im Rahmen des Tagging–Systems erzeugt. Dieser enthält die Anzahl der Verwendungen des Begriffes als Tag und die eindeutigen Bezeichner der Dokumente, also der Artikel und Designs, die mit dem Begriff getaggt wurden. Weiterhin wird für jedes Knotenobjekt ein global eindeutiger Bezeichner generiert, um die spätere Referenzierung der Knoten zu ermöglichen. Listing 4.4 zeigt ein Beispiel für ein so erzeugtes Knotenobjekt.

```
{
    "_id" : ObjectId("51efc20147cae77dfc02e0ac"),
    "language" : "de",
    "string" : "mama",
    "tagProperties" : {
        "occurenceCount" : 3,
        "articleCount" : 2,
        "designCount" : 1,
        "articleIDs" : [
            24231101,
            24231105
        ],
        "designIDs" : [
            15514592
        ]
    }
}
```

Listing 4.4: JSON–Beispiel für einen aus den Tagging–Daten erzeugten Knoten

Die Erzeugung der Kanten erfolgt wie in Abschnitt 5.3.3 beschrieben. Für jedes gemeinsame Auftreten von zwei Tags werden zwei Kantenobjekte erzeugt. Diese beschreiben gerichtete Kanten zwischen den Begriffen, die ein gemeinsames Auftreten der Tags repräsentieren. Neben den eindeutigen Bezeichnern der Quell- und Zielknoten enthält das Kantenobjekt den Kantentyp *Tagging–Kookkurrenz* sowie die absolute Anzahl gemeinsamer Vorkommen der Tags und die in Abschnitt 3.3.2 beschriebenen Kookkurrenzmaße. Außerdem erhält die Kante selbst einen global eindeutigen Bezeichner zur Referenzierung. Ein Beispiel für eine aus den Tagging–Daten erzeugte Kante ist in Listing 4.5 dargestellt.

```
{
    "_id" : ObjectId("51efd6f61177ff360605bd99"),
    "source" : ObjectId("51efc1af47cae77dfc00c3f8"),
    "target" : ObjectId("51efc1e047cae77dfc02087c"),
    "type" : "tag-co-occurence",
    "occurences" : 1,
    "dice" : 0.0001317089232795522,
    "jaccard" : 0.00006585879873551106,
    "cosine" : 0.008115343414514944
}
```

Listing 4.5: JSON–Beispiel für eine aus den Tagging–Daten erzeugte Kante

4.1.5 Integration

Da es sich bei der Integration der Tagging–Daten um die initiale Erstellung des Weltausschnittes handelt, ist der letzte Schritt trivial. Die transformierten Daten sind lediglich in die Zieldatenbank zu kopieren, da noch keine weiteren Daten vorhanden sind, mit denen sie integriert werden müssen.

4.1.6 Ergebnisse

Durch auf die Tagging–Daten angewendeten Link–Discovery–Schritte wurden insgesamt 314 351 Knoten und 21 834 868 Kanten erzeugt.

Eine mögliche Metrik für die quantitativen Eigenschaften der Ergebnisse ist die Verteilung der ausgehenden Kanten, die jeder Knoten besitzt. Diese ist als Histogramm in Abbildung 4.3 dargestellt. Hierbei ist anzumerken, dass die Klassen des Histogrammes exponentiell breiter werden, um Knoten, die sehr viele Kanten besitzen, im Histogramm darstellen zu können.

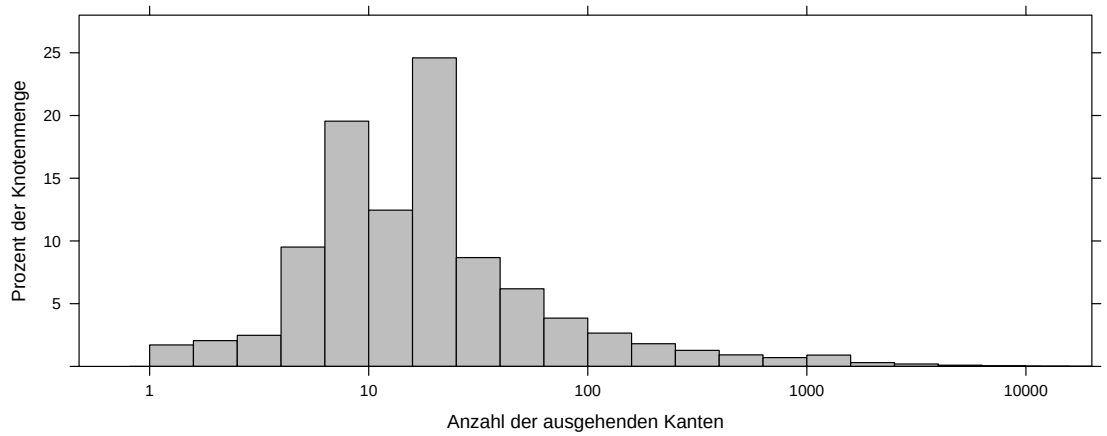


Abbildung 4.3: Histogramm der Verteilung der Tagging–Kookkurrenz–Kanten

Tabelle 4.2 zeigt die statistischen Kennzahlen Minimum, erstes Quartil, Median, drittes Quartil, Maximum und Durchschnitt der Verteilung der Kanten nach Berechnung der Tagging–Kookkurrenz.

Bei der Interpretation dieser Daten fällt auf, dass die Anzahl der ausgehenden Kanten sehr ungleich verteilt ist. Es existieren sehr viele Knoten mit wenigen Kanten und sehr wenige

Kennzahl	Wert
min	0
$Q_{0.25}$	9
$Q_{0.5}$	16
$Q_{0.75}$	29
max	35 169
avg	74.64

Tabelle 4.2: Statistische Kennzahlen für die Verteilung der Tagging–Kookkurrenz–Kanten

Knoten mit vielen Kanten. Der verhältnismäßig hohe Median sagt allerdings ebenfalls aus, dass die Hälfte aller Knoten mehr als 16 ausgehende Kanten besitzen. Dies lässt jedoch keine Aussage über die inhaltliche Qualität der erzeugten Zusammenhänge zu. Diese kann nur auf Basis eines einzelnen Begriffes und dessen Beziehungen beurteilt werden.

Der Begriff, der das Maximum von 35 169 ausgehenden Kanten erreicht, ist der Begriff "liebe". Dieser Begriff zählt zu den meistgesuchten und verwendeten Tags auf der Spreadshirt–Plattform. Dies spiegelt sich offensichtlich auch darin wieder, dass er zusammen mit 35 169 anderen unterschiedlichen Tags vergeben wurde.

Nachdem in diesem Abschnitt die Integration der Tagging–Daten beschrieben wurde, beschäftigt sich der folgende Abschnitt mit der Anreicherung des Weltausschnittes mit den Daten des Clicktracking–Systems von Spreadshirt.

4.2 Anreicherung des Weltausschnittes mit Clicktracking–Daten

Spreadshirt betreibt ein Clicktracking–System (siehe Abschnitt 3.5.2), welches die Klicks der Benutzer auf Artikel und Designs auf Suchergebnisseiten aufzeichnet. Dabei ist unerheblich, ob der Benutzer bei Spreadshirt registriert oder im Moment des Klicks angemeldet ist. Dieses System sammelt Daten von beiden Spreadshirt–Plattformen (siehe Abschnitt 2.2.1). In Abbildung 4.4 ist beispielhaft eine Suchergebnisseite der Spreadshirt–Plattform abgebildet, welche die gefundenen Designs für eine Suchanfrage auflistet.

Die von diesem System erzeugten Daten können für die Link Discovery von großer Bedeutung sein, da sie eine andere Perspektive auf die Begriffe im Graphen liefern. Die Tags

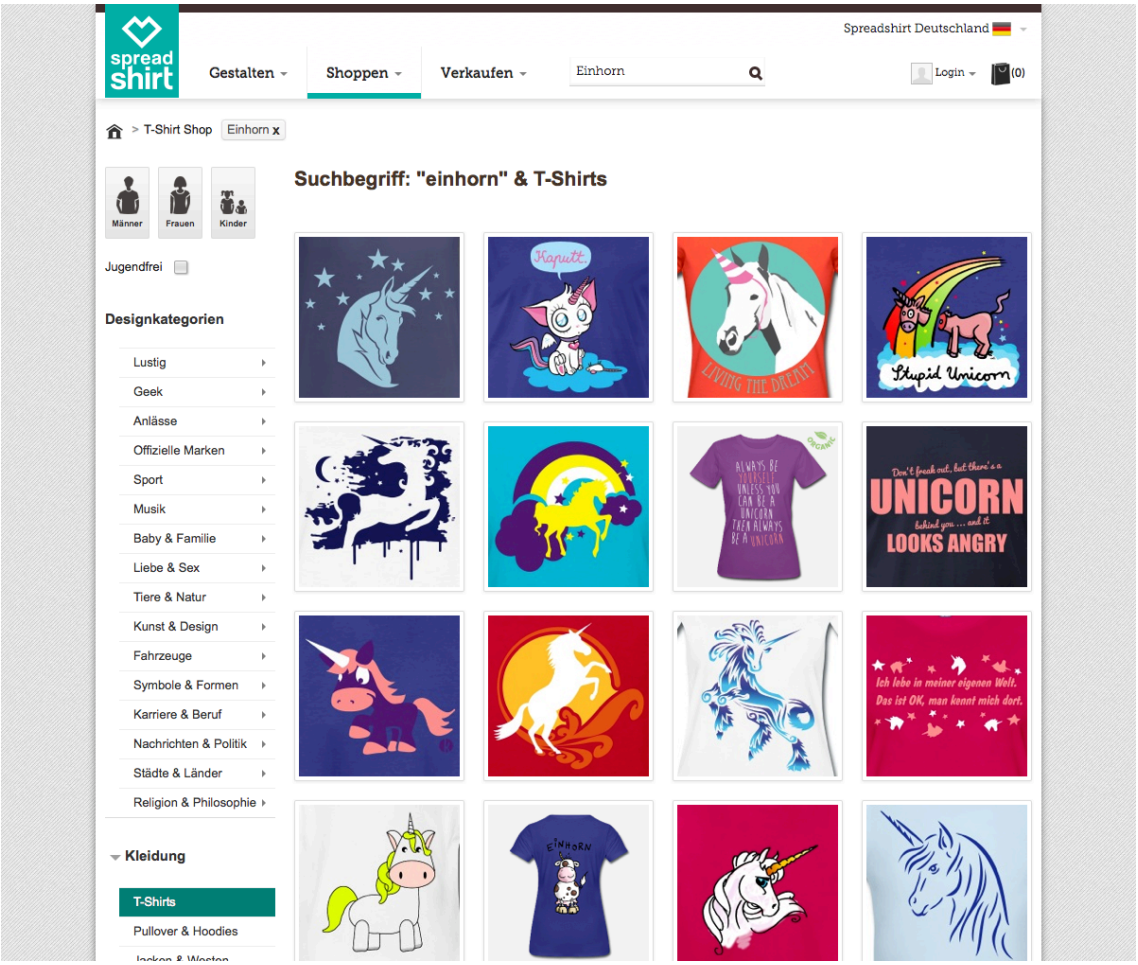


Abbildung 4.4: Beispiel für eine Spreadshirt–Suchergebnisseite

liefern die Sicht der Partner, das heißt der Personen, die Inhalte hochladen und verkaufen möchten. Die Klicks beschreiben die Sicht der Käufer, also der Personen, die nach Inhalten suchen. Durch die Auswertung der Clicktracking–Daten ergibt sich die Möglichkeit, eine Form der Validierung der durch Partner vergebenen Metadaten zu erhalten. Die Annahme hierbei ist, dass Käufer nur auf Suchergebnisse klicken, die eine inhaltliche Relevanz zum eingegebenen Suchbegriff besitzen und somit ihren Erwartungen bezüglich des Suchbegriffes gerecht werden.

Wie bereits für die Tagging–Daten, werden im Folgenden auch für das Clicktracking die in Abschnitt 3.2.1 genannten Schritte Import, Bereinigung, Reduktion, Transformation und Integration näher erläutert. Der Ansatz zur Erzeugung der Verbindungen ist, wie schon bei den Tagging–Daten, kookkurrenzbasiert.

4.2.1 Import

Das Clicktracking–System erzeugt Dateien im JSON–Format, die zu jedem Klick auf einer Ergebnisseite die wesentlichen Informationen enthalten. Je Klick ist ein JSON–Dokument abgespeichert. Ein Beispiel für ein solches Dokument ist in Listing 4.6 dargestellt.

```
{
    "date": "01.07.2013␣00:09:31_633",
    "path": "/track/eu/205909/1E3B6E3E-4496-C51A14A8FA25/2.10.4/list",
    "params": {
        "locale": "[de_DE]",
        "search-query": "[biene]",
        "cl": "[a18869874,␣p25446183,␣i49]"
    }
}
```

Listing 4.6: JSON–Beispiel für ein Clicktracking–Rohdokument

Ein Clicktracking–Dokument enthält die Attribute *Datum*, *Pfad*, *Gebietsschema*, *Suchbegriff* und die Daten des eigentlichen Klicks, welche den geklickten *Artikel*, das geklickte *Produkt* und den *Index*, also die Position des geklickten Inhaltes auf der Suchergebnisseite beschreiben. Die Unterscheidung zwischen Produkt und Artikel ist im Domänenmodell von Spreadshirt begründet (siehe auch Abschnitt 2.2.1) und für die Link Discovery nicht von Interesse. Es genügt, den geklickten Artikel im Weiteren näher zu betrachten.

Auffällig ist, dass die Möglichkeiten des JSON–Formates bei der Speicherung der Klickdaten nicht vollständig ausgenutzt wurden. So sind die Werte, die das geklickte Dokument beschreiben, als Zeichenkette abgelegt und zusätzlich die eindeutigen Bezeichner mit einem Buchstaben versehen, der ihren Typ angibt. Des weiteren enthalten das Gebietsschema und der Suchbegriff zusätzliche eckige Klammern. Diese Defekte sollten im Bereinigungsschritt beseitigt werden, um ein nutzbareres Datenformat zu erhalten.

Da das Clicktracking–System zum Zeitpunkt des Imports erst 3 Monate Daten aufzeichnete, standen 2 249 942 solcher Klickdokumente zur Verfügung.

4.2.2 Bereinigung

Im Bereinigungsschritt sollten zunächst die im vorherigen Abschnitt genannten Defekte an den Daten des Clicktracking–Systems beseitigt werden. Dazu gehört die Entfernung der eckigen Klammern in Suchbegriff und Gebietsschema und die Extraktion des eindeutigen

Bezeichners des geklickten Artikels. Aus dem Gebietsschema ist nur die Sprache von Interesse. Außerdem wurden für den Suchbegriff die gleichen Bereinigungsoperationen wie für die Tagging–Daten vorgenommen, also die Entfernung von überflüssigen Leerzeichen, Groß-/Kleinschreibung, nicht druckbarer Sonderzeichen und Satzzeichen.

Im Bereinigungsschritt werden so auch einfacher verarbeitbare Dokumente erzeugt, da die Möglichkeiten des JSON–Formates besser ausgenutzt werden. Listing 4.7 zeigt das Ergebnis der Bereinigung des in Abschnitt 4.2.1 gezeigten Beispieldokumentes.

```
{
    "_id": ObjectId("51e7b1e0417498f9c6868939"),
    "query" : "biene",
    "date" : "2013-07-01T00:09:31.633Z",
    "articleId" : 18869874,
    "index" : 57,
    "language" : "de"
}
```

Listing 4.7: JSON–Beispiel für die Bereinigung der Clicktracking–Dokumente

4.2.3 Reduktion

Die Reduktion der Clicktracking–Daten besteht zum einen aus einer Duplikatentfernung, zum anderen aus der Einschränkung der Sprache.

Im Sinne der Kookkurrenz ist es nicht von Bedeutung, wenn Paare aus Suchbegriffen und geklickten Artikeln mehrfach auftauchen, da hierfür nur das gemeinsame Auftreten unterschiedlicher Suchbegriffe betrachtet wird. Somit besteht die Duplikatentfernung lediglich darin, aus mehrfach vorkommenden Artikel-/Klickpaaren genau eines auszuwählen.

Außerdem erfolgte, wie schon bei den Tag–Daten, eine Einschränkung auf Klicks, die als *deutsch* gekennzeichnet sind.

Nach dem Reduktionsschritt verblieben zur Transformation noch 411 341 Klicks.

4.2.4 Transformation

Der Transformationsschritt dient zur Umformung der Clicktracking–Daten in die Graphenrepräsentation des Weltausschnittes (siehe Abschnitt 3.4.2). Diese Umformung wird

durch die Ermittlung von Kookkurrenz durchgeführt. Diese bestimmt sich hierbei daraus, welche Suchbegriffe zum Klick auf einen Artikel geführt haben. Wird ein Artikel zu mehreren Suchbegriffen geklickt, liegt die Vermutung nah, dass zwischen den Suchbegriffen ein irgendwie gearteter Zusammenhang besteht.

Ziel der Transformation ist somit die Erzeugung von Knoten und Kanten. Die Knoten werden zusätzlich mit einem Kontext verknüpft, der die Eigenschaften des durch den Knoten repräsentierten Begriff im Kontext des Clicktracking–Systems beschreibt. Konkret sind dies die Artikel, die zu dem Begriff als Suchbegriff geklickt wurden. Listing 4.8 zeigt einen aus den Clicktracking–Daten erzeugten Knoten.

```
{
    "_id": ObjectId("51e7f1e04146498f9c6868945"),
    "string": "biene",
    "language": "de",
    "clickProperties": [
        { "articleId": 4512 },
        { "articleId": 4794 },
        ...
    ]
}
```

Listing 4.8: JSON–Beispiel für ein aus den Clicktracking–Daten erzeugtes Knotenobjekt

Die Kanten besitzen die gleiche Form wie die Kookkurrenzkanten, die bei der Integration der Tagging–Daten erzeugt wurden. Lediglich der Typ der Kanten ist unterschiedlich und lautet *Clicktracking–Kookkurrenz*. Ein Beispiel für eine solche Kante ist in Listing 4.9 dargestellt.

```
{
    "_id": ObjectId("51e91aff3b6a20bfd68c468a")
    "source" : ObjectId("51e91af93b6a20bfd68b0bed"),
    "target" : ObjectId("51e91aff3b6a20bfd68c463e"),
    "type": "click-co-occurence",
    "occs" : 1,
    "dice" : 0.003883495145631068,
    "jaccard" : 0.0019455252918287938,
    "cosine" : 0.04410810913912309
}
```

Listing 4.9: JSON–Beispiel für ein aus den Clicktracking–Daten erzeugtes Kantenobjekt

Die Durchführung des Transformationsschrittes erfolgte mittels MapReduce (siehe Abschnitt 5.3.3). Dadurch wurden 92 727 Knoten und 310 860 Kanten erzeugt.

4.2.5 Integration

Die Integration der erzeugten Daten stellt eine Vereinigung des vorhandenen Weltausschnittes mit dem im Transformationsschritt erzeugten Graphen dar.

Die Knotenmenge wird derart vereinigt, dass sie die Eigenschaft behält, dass Paare aus Sprache und Zeichenkette eindeutig sind. Somit werden bei bereits vorhandenen Knoten die zusätzlichen Informationen bezüglich des Clicktrackings als Attribute hinzugefügt. Existiert eine Kombination aus Sprache und Zeichenkette noch nicht im Zielgraphen, so wird der entsprechende Knoten eingefügt.

Da die erzeugten Kanten einen noch nicht im Graphen vorhandenen Typ besitzen, müssen keine Kanten zusammengeführt werden. Jedoch werden die Bezeichner der Ziel- und Quellknoten entsprechend angepasst, wenn bei der Integration der Knoten eine Zusammenführung stattgefunden hat.

4.2.6 Ergebnisse

Durch die auf die Clicktracking–Daten angewendeten Link–Discovery–Schritte wurden dem Weltausschnitt 78 237 neue Begriffe und 310 860 neue Zusammenhänge vom Typ *Clicktracking–Kookkurrenz* hinzugefügt.

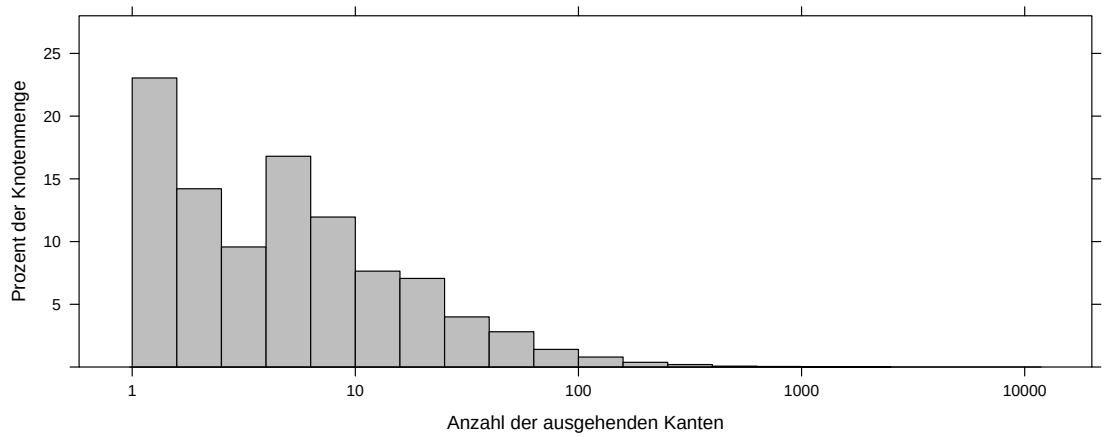


Abbildung 4.5: Histogramm der Verteilung der Clicktracking–Kookkurrenz–Kanten

Wie schon für die Tagging–Daten in Abschnitt 4.1.6, wird im folgenden die Verteilung der Clicktracking–Kanten je Knoten als Metrik für die Ergebnisse der Anreicherung mit den

Clicktracking–Daten verwendet. Diese ist in Abbildung 4.5, analog zu Abbildung 4.3, als Histogramm mit exponentiell wachsenden Klassen dargestellt.

Tabelle 4.3 zeigt die statistischen Kennzahlen Minimum, erstes Quartil, Median, drittes Quartil, Maximum und Durchschnitt der Verteilung der Kanten nach Berechnung der Clicktracking–Kookkurrenz.

Kennzahl	Wert
min	1
$Q_{0.25}$	2
$Q_{0.5}$	4
$Q_{0.75}$	10
max	1688
avg	12

Tabelle 4.3: Statistische Kennzahlen für die Verteilung der Clicktracking–Kookkurrenz–Kanten

Bei der Interpretation dieser Daten fällt auf, dass im Vergleich zu den Tagging–Daten zum einen deutlich weniger Kanten, zum anderen auch durchschnittlich weniger Kanten pro Knoten erzeugt wurden. So muss bei circa einem Viertel der Knoten, die Kanten vom Typ Clicktracking–Kookkurrenz besitzen, mit zwei oder weniger Beziehungen gerechnet werden. Durch die geringe Kantenanzahl wird die Gesamtverteilung der Kanten jedoch nicht wesentlich beeinflusst (siehe Abschnitt 4.5).

Nachdem die Anreicherung der Clicktracking–Daten durchgeführt wurde, beschäftigt sich der folgende Abschnitt mit der Anreicherung durch die Zerlegung von Wortgruppen in Einzelwörter.

4.3 Anreicherung des Weltausschnittes durch Zerlegung von Wortgruppen

Die Zerlegung von Begriffen, die aus mehr als einem Wort bestehen, ist ein Anreicherungsschritt durch das Mining von schon im Weltausschnitt vorhandenen Daten (siehe Abschnitt 3.2.3). Neben der Erzeugung zusätzlicher Verbindungen ist er für weitere Integrationsschritte von Nutzen, da es Vorteile bringt, wenn möglichst viele Einzelwörter im Weltausschnitt gespeichert sind (siehe Abschnitt 4.4).

4.3.1 Vorgehensweise

Zum Zeitpunkt des Importes befanden sich 147 364 Begriffe im Weltausschnitt, die aus mehreren Wörtern bestehen. Dies entspricht 47 Prozent aller bereinigten Tags, die als deutsch gekennzeichnet sind. Dieser Umstand legt die Vermutung nahe, dass in diesen zusammengesetzten Begriffen auch Wörter enthalten sind, die nicht als Einzelwörter im Weltausschnitt existieren.

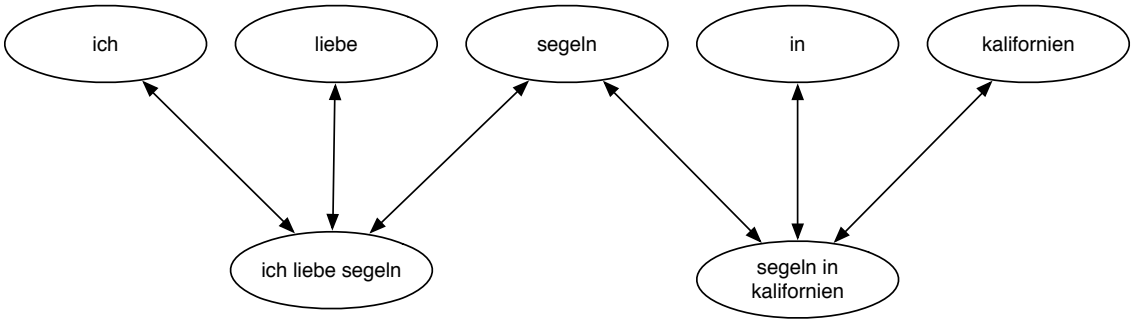


Abbildung 4.6: Beispielhafter Graphausschnitt nach der Zerlegung

Werden diese Begriffe in ihre Einzelwörter zerlegt, entstehen einerseits unter Umständen neue Begriffe, andererseits können in diesem Schritt Zusammenhänge vom Typ *Zerlegung* beziehungsweise *Zusammensetzung* eingefügt werden. Somit sind nach dem Schritt der Zerlegung weitere Informationen über den Kontext, in dem Wörter verwendet werden, verfügbar. Abbildung 4.6 zeigt beispielhaft das Ergebnis einer solchen Zerlegung.

4.3.2 Ergebnisse

Durch die Anwendung des Zerlegungsschrittes auf die vorhandenen Begriffe wurden insgesamt 38 349 neue Knoten und 1 238 900 neue Kanten erzeugt, die für spätere Analyseschritte genutzt werden können.

Die Verteilung der erzeugten Kanten über alle Knoten, die von der Zerlegung betroffen sind, ist in Abbildung 4.7 als Histogramm dargestellt. Statistische Kennzahlen dieser Verteilung sind in Tabelle 4.4 aufgeführt.

Bei der Zerlegung wurden somit deutlich weniger Kanten pro Knoten erzeugt, als noch bei den Tagging– und Clicktracking–Daten. Nur ein Viertel der betreffenden Knoten be-

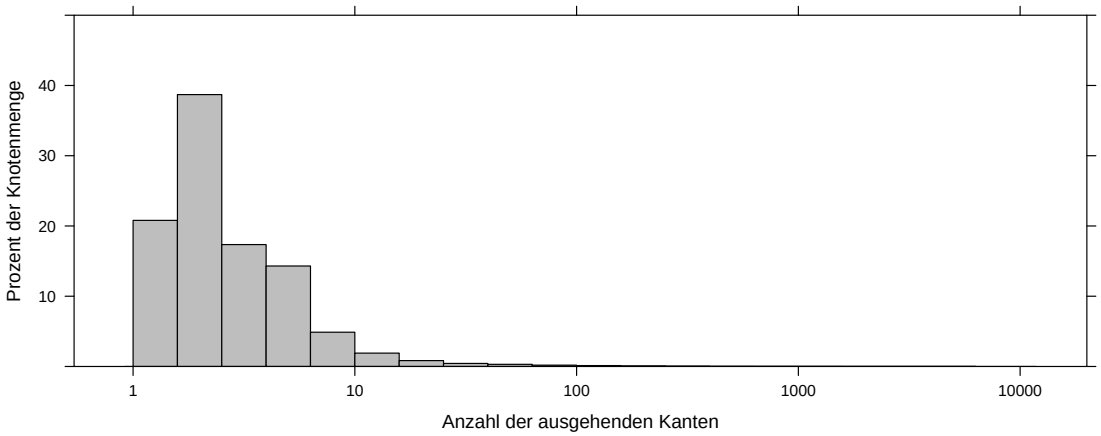


Abbildung 4.7: Histogramm der Verteilung der Zerlegungs– und Zusammensetzungskanten

sitzen mehr als drei Kanten vom Typ *Zerlegung* oder *Zusammensetzung*. Der Einfluss der Zerlegung auf den gesamten Datenbestand wird in Abschnitt 4.5 dargestellt.

Kennzahl	Wert
min	1
$Q_{0.25}$	2
$Q_{0.5}$	2
$Q_{0.75}$	3
max	5215
avg	4.358

Tabelle 4.4: Statistische Kennzahlen für die Verteilung der Zerlegungs– und Zusammensetzungskanten

Nachdem im Zerlegungsschritt Einzelwörter erzeugt wurden, können diese im folgenden Abschnitt für den Import der Wortschatz–Daten genutzt werden.

4.4 Anreicherung des Weltausschnittes mit Wortschatz-Daten

Wie in Abschnitt 3.5.1 bereits einführend beschrieben, betreibt die Universität Leipzig ein Wortschatz–Projekt [WSL]. Im Rahmen dieses Projektes wird durch die Analyse von

großen Textmengen eine Datenbank deutscher Wörter, deren Bedeutungen, grammatikalische Eigenschaften, Häufigkeiten und Kookkurrenzen in Texten und Beziehungen zu anderen Wörtern aufgebaut. Somit stellt dieses Projekt eine sehr gute Möglichkeit dar, weitere Zusammenhänge zwischen den schon im Weltausschnitt vorhandenen Begriffen herzustellen. Neben den Daten, die im Spreadshirt–Kontext entstehen, können somit auch allgemeine lexikalische Daten hinzugefügt und für spätere Analysen genutzt werden.

Neben einem Webportal [WSL] stellt dieses Projekt eine API bereit, über die die Daten des Wortschatzes programmatisch abgefragt werden können. Diese API wurde mittels einer Bibliothek für die Programmiersprache Ruby [RW13] im Rahmen dieser Arbeit für einen weiteren Integrationsschritt zur Link Discovery genutzt. Dazu wurden die Informationen *Grundform*, *Wortformen*, *Kategorien*, *Synonyme* und *Thesaurus–Beziehungen* ausgewertet (siehe Abschnitt 3.5.3).

Wie bereits für die anderen Datenquellen werden im Folgenden auch für den Wortschatz die Schritte des Imports, der Bereinigung, der Reduktion, der Transformation und der Integration beschrieben (siehe Abschnitt 3.2.1).

4.4.1 Import

Da für die Anfrage an die Wortschatz–API nur Einzelwörter und keine Wortgruppen genutzt werden können, muss eine Auswahl der anzufragenden Daten getroffen werden. Im Anreicherungsschritt der Zerlegung von Wortgruppen wurden aus allen zu diesem Zeitpunkt vorhandenen Begriffen Einzelwörter gebildet (siehe Abschnitt 4.3). Diese können nun beim Import der Wortschatz–Daten genutzt werden, um die Anfragen zu formulieren. Zum Zeitpunkt des Importes standen 197 614 Einzelwörter zur Verfügung. Da die im Weltausschnitt gespeicherten Daten keine Groß- und Kleinschreibung enthalten, die Wortschatz–API diese jedoch berücksichtigt, wurde jedes Wort jeweils mit großem und kleinem Anfangsbuchstaben angefragt. Somit wurde die doppelte Menge an Rohdaten, 395 228 Objekte, erzeugt.

Listing 4.10 zeigt beispielhaft ein importiertes Objekt des Wortschatzes nach dem Import. Dieses enthält Listen von *Grundformen* mit der entsprechenden Wortform als Kürzel, den *Kategorien* des Wortes, *Synonymen*, *Thesaurus-Beziehungen* sowie *Wortformen* als Attribute.

```
{
    "_id" : ObjectId("51f7aa06eba16044e900015a"),
    "string" : "Kopf",
    "baseform" : [
        "Kopf",
        "N"
    ],
    "domain" : [
        "Medizin",
        "Anatomie",
        "Literarische/Motive/Stoffe/Gestalten",
        "Körperteile"
    ],
    "synonyms" : [
        "Chef",
        "Figur",
        "Gestalt",
        "Haupt",
        "Jemand",
        "Individuum",
        "Figur"
    ],
    "thesaurus" : [
        "Titel",
        "Hand",
        "Kopf",
        "Mensch",
        "Gesicht",
        "Spitze",
        "Arm",
        "Gestalt"
    ],
    "wordforms" : [
        "Kopf",
        "Köpfe",
        "Köpfen",
        "Kopfes",
        "Kopfs"
    ]
}
```

Listing 4.10: JSON–Beispiel für Rohdaten aus dem Wortschatz

Hierbei ist zu beachten, dass nicht alle Attribute bei allen Wörtern vorhanden sind. Dies hängt davon ab, ob der Wortschatz die Informationen zur Verfügung stellen kann. Somit ist es auch möglich, dass Objekte importiert werden, die keine zusätzlichen Informationen enthalten.

4.4.2 Bereinigung

Zur Bereinigung der importierten Wortschatz–Daten muss in einem ersten Schritt die Groß- und Kleinschreibung entfernt werden, da diese an erster Stelle nur für die Anfragen an die API wieder in den Datenbestand eingeführt wurde.

Weiterhin werden die Kategorien “Vorname” und “Nachname” entfernt, da diese an über 25 000 Wörter vergeben sind und somit für die Verwendung zur Link Discovery ungeeignet sind und zu viele irrelevante Kanten erzeugen würden.

```
{
    _id: ...,
    string: "Kopfs",
    baseform: {
        word: "Kopf",
        type: "N"
    }
}
```

Listing 4.11: JSON–Beispiel für die Umformung der Grundform eines Wortes

Ein letzter Bereinigungsschritt besteht in der Veränderung des Formates der Grundform des Wortes. Die Wortschatz–API liefert lediglich ein Array, in dem das erste Element die Grundform und das zweite Element die Wortart ist. Zur Bereinigung wird diese Eigenschaft in ein geeignetes JSON–Format überführt, welches in Listing 4.11 dargestellt ist.

4.4.3 Reduktion

Der Reduktionsschritt besteht in der Zusammenführung von Objekten, die die gleiche Zeichenkette enthalten. Diese Duplikate sind bei der Entfernung der Groß– und Kleinschreibung im Bereinigungsschritt entstanden. Wie schon im Reduktionsschritt der Daten

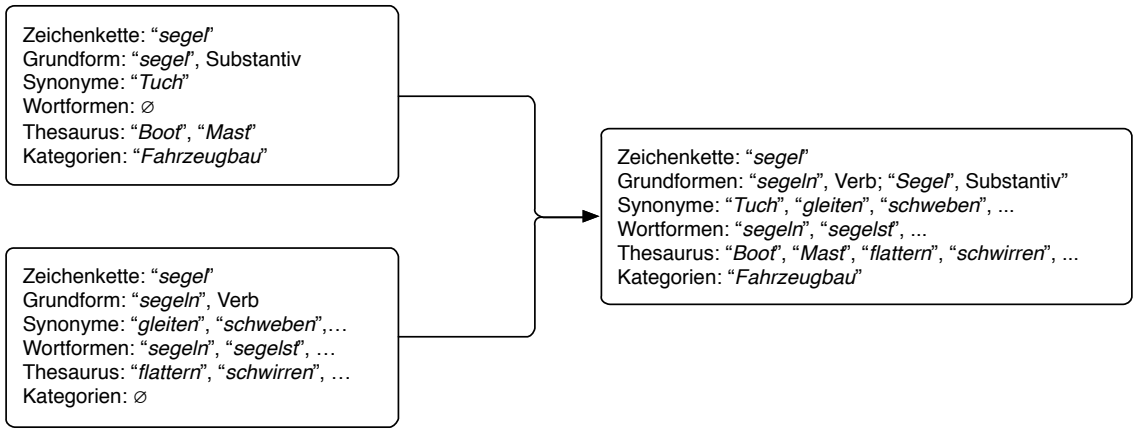


Abbildung 4.8: Reduktion der Wortschatz–Daten

des Tagging–Systems (Abschnitt 4.1.3), muss auch hierbei das Entstehen neuer Duplikate vermieden werden. Dies bedeutet, dass Listen der Wortbeziehungen ebenfalls zusammengeführt werden, wobei jedes Wort nur einmal enthalten sein darf. Die Reduktion ist beispielhaft in Abbildung 4.8 dargestellt. Dabei ist zu beachten, dass bei der Zusammenführung mehrere Grundformen entstehen können, wodurch dieses Attribut in ein Array umgewandelt wird.

Durch die Reduktion wird die Anzahl der Dokumente wieder auf die ursprüngliche Menge von Einzelwörtern reduziert und beträgt demnach 197 614.

4.4.4 Transformation

Die Transformation der Wortschatz–Daten besteht, wie bereits bei den anderen Datenquellen, in einer Überführung in die Graphenrepräsentation des Weltausschnittes aus Abschnitt 3.4.2.

Die Knoten repräsentieren alle Wörter, die durch die Benutzung der Wortschatz–API bekannt sind. Dazu zählen einerseits sowohl die angefragten Wörter, als auch die von der API zurückgegebenen Wortbeziehungen. Der Kontext dieser durch die Knoten repräsentierten Begriffe enthält die Rohdaten des Wortschatzes, um sie bei gegebenenfalls stattfindenden späteren Analysen nutzen zu können.

Die Kanten für *Synonyme*, *Thesaurus*, *Grundform* und *Wortformen* können direkt aus den entsprechenden Attributen erzeugt werden und besitzen die entsprechenden Kantentypen.

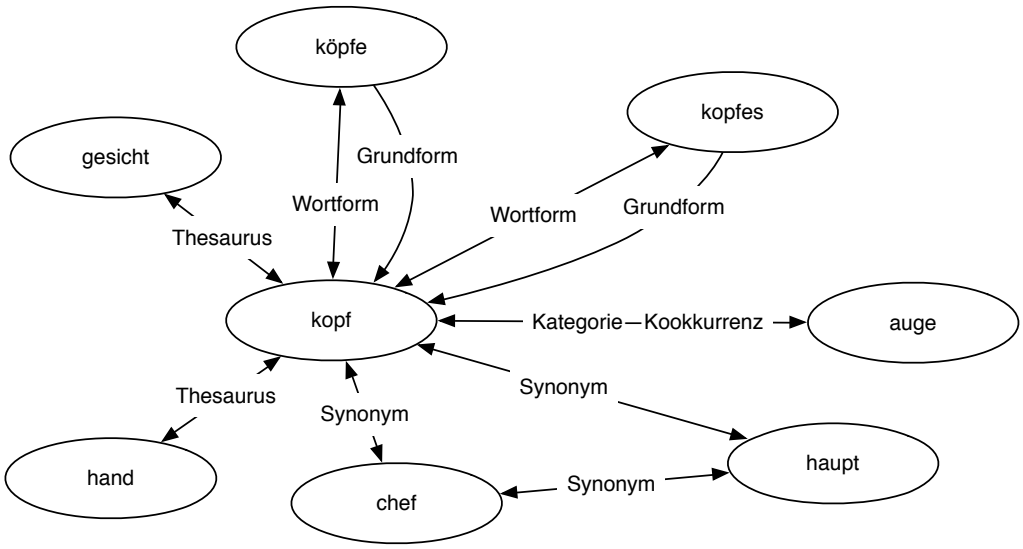


Abbildung 4.9: Beispiel–Graphausschnitt der transformierten Wortschatz–Daten

Sie enthalten keine weiteren Attribute, da über diese Beziehungen keine weiteren Informationen verfügbar sind.

Einen Sonderfall stellen die Kategorien der Wörter dar. Diese können in der vorliegenden Form nicht direkt zur Link Discovery genutzt werden. Daher werden diese Daten mittels der Ermittlung von Kookkurrenzen umgeformt. Dabei ist die Kookkurrenz zweier Begriffe durch die gemeinsame Einordnung in Kategorien definiert. Somit werden bei der Transformation mittels Kookkurrenzberechnung Kanten mit den Kookkurrenzmaßen aus Abschnitt 3.3.2 und dem Kantentyp *Kategorie–Kookkurrenz* erzeugt.

Zusammenfassend werden bei der Transformation demnach Kanten mit den Typen *Synonym*, *Thesaurus*, *Wortform*, *Grundform* und *Kategorie–Kookkurrenz* erzeugt. In Abbildung 4.9 wird beispielhaft ein Ausschnitt des resultierenden Graphen gezeigt. Dabei fällt auf, dass mit Ausnahme der Grundform–Kanten jede Kante in beide Richtungen existiert. Dies ist der Art der Beziehungen geschuldet, da diese in beide Richtungen gültig sind.

4.4.5 Integration

Zur Integration wird der im Transformationsschritt erzeugte Graph mit dem vorhandenen Graphen vereinigt. Diese Vereinigung wird analog zur Integration der Clicktracking–Daten in Abschnitt 4.2.5 durchgeführt.

Kennzahl	Wert
min	1
$Q_{0.25}$	1
$Q_{0.5}$	2
$Q_{0.75}$	10
max	11 020
avg	203.8

Tabelle 4.5: Statistische Kennzahlen für die Verteilung der Wortschatz–Kanten

4.4.6 Ergebnisse

Insgesamt wurden durch die Integration der Wortschatz–Daten 145 023 neue Knoten und 50 227 965 neue Kanten erzeugt. Dabei entfallen 48 399 466 Kanten auf Kookkurrenz von Kategorien, 550 270 Kanten auf Wortformen, 149 381 Kanten auf Grundformen, 279 118 Kanten auf Synonyme und 849 730 Kanten auf Thesaurus–Beziehungen.

In Abbildung 4.10 ist die Verteilung der durch die Integration des Wortschatzes erzeugten Kanten als Histogramm mit exponentiell wachsenden Klassen abgebildet. Tabelle 4.5 zeigt statistische Kennzahlen dieser Verteilung.

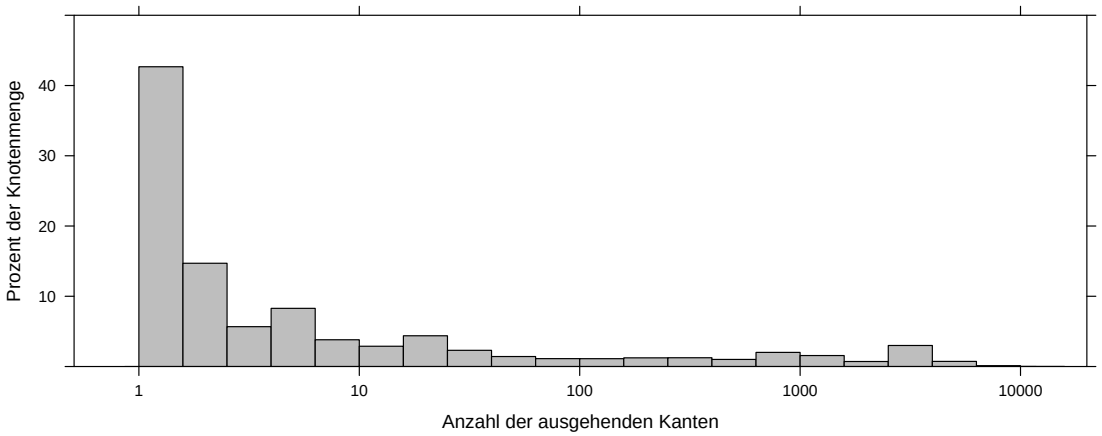


Abbildung 4.10: Histogramm der Verteilung der Wortschatz–Kanten

Bei Inspektion der Daten fällt auf, dass der Großteil der betroffenen Knoten nur eine Wortschatz–Kante besitzt. Jedoch deutet der hohe Durchschnitt, verbunden mit dem Umstand, dass $Q_{0.75} = 10$ ist, darauf hin, dass im Gegensatz zu den anderen Integrationsschritten deutlich mehr Knoten existieren, die viele Kanten besitzen.

Schritt	Knoten	Kanten
Tags	314 351	21 834 868
Clicktracking	78 237	310 860
Zerlegung	38 349	1 238 900
Wortschatz	145 023	50 227 965
Gesamt	575 960	73 612 593

Tabelle 4.6: Entwicklung der Knoten– und Kantenanzahl nach jedem Link–Discovery–Schritt

Nachdem alle Anreicherungsschritte durchgeführt und beschrieben wurde, beschäftigt sich der nächste Abschnitt mit den zusammengefassten Ergebnissen der Integrations– und Anreicherungsschritte.

4.5 Ergebnisse der Integrations– und Anreicherungsschritte

In den bisherigen Auswertungen der Integrations– und Anreicherungsschritte wurden die Eigenschaften der vom jeweiligen Schritt betroffenen Begriffe analysiert. Dieser Abschnitt beschäftigt sich mit der Veränderung des durch den Graphen repräsentierten Weltausschnittes bei jedem durchgeführten Link–Discovery–Schritt.

4.5.1 Anzahl der Knoten und Kanten

Tabelle 4.6 zeigt zusammengefasst die quantitative Veränderung des Graphen nach jedem durchgeführten Schritt. Dabei zeigt sich, dass die Integration der Daten des Wortschatzes mit Abstand die meisten neuen Kanten in den Graphen eingefügt hat.

Bei der Verwendung der Clicktracking–Daten wurden im Verhältnis wenig neue Knoten und Kanten erzeugt. Dies ist im Wesentlichen auf den zum Zeitpunkt des Importes noch geringen Datenbestand zurückzuführen. Daher sollte dieser Schritt zukünftig wiederholt werden, da mit der längeren Laufzeit des Clicktracking–Systems auch ein größeres Potenzial für neue Verknüpfungen vorhanden ist.

4.5.2 Verteilung der Kanten

Neben der absoluten Anzahl der Knoten und Kanten sind bei einer Betrachtung der quantitativen Ergebnisse auch die Anzahl der Kanten, die von einem Knoten ausgehen, von Interesse. Diese Verteilung der Kanten je Knoten nach jedem Schritt ist in den Abbildungen 4.11 bis 4.14 als Histogramm dargestellt.

Schritt	min	$Q_{0.25}$	$Q_{0.5}$	$Q_{0.75}$	max	avg
Tags	0	8	15	26	35 170	69.64
Clicktracking	0	3	10	23	35 170	56.41
Zerlegung	0	4	11	24	37 940	54.26
Wortschatz	0	2	8	23	38 380	127.8

Tabelle 4.7: Statistische Kennzahlen der Kantenverteilung nach jedem Link–Discovery–Schritt

Tabelle 4.7 zeigt einige statistische Kennzahlen der Kantenverteilung nach jedem Schritt. Dabei sind das Minimum, das untere, mittlere (Median) und obere Quartil, das Maximum und der Durchschnitt dargestellt.

Auffällig ist hierbei, dass die Integration von Clicktracking- und Wortschatz–Daten zu einer Herabsetzung des Medians führten. Dies bedeutet, dass nach Durchführung dieser Schritte verhältnismäßig weniger viel verbundene Knoten im Datenbestand existierten als davor. Jedoch deutet die Entwicklung des Durchschnittes nach der Integration der Wortschatz–Daten darauf hin, dass die Knotenanzahl viel verbundener Knoten nach diesem Schritt deutlich größer geworden ist.

Das Absinken des Durchschnittes nach Integration der Clicktracking–Daten kann mit der geringen Menge der Daten begründet werden. Durch die geringe Anzahl von Kookkurrenzen konnten viel verbundene Knoten keinen relevanten Zugewinn an Verbindungen verzeichnen, während für wenig verbundene Knoten verhältnismäßig mehr neue Verbindungen hinzukamen.

Die Entwicklung der Verteilung zeigt, dass der Anteil wenig verbundener Knoten an der Knotenmenge ständig gewachsen ist. Daraus kann geschlussfolgert werden, dass mit jedem Schritt mehr Knoten existierten, die überhaupt Verbindungen besitzen. Speziell die Integration der Wortschatz–Daten hat gleichzeitig den Anteil der wenig verbundenen als auch der viel verbundenen Knoten erhöht.

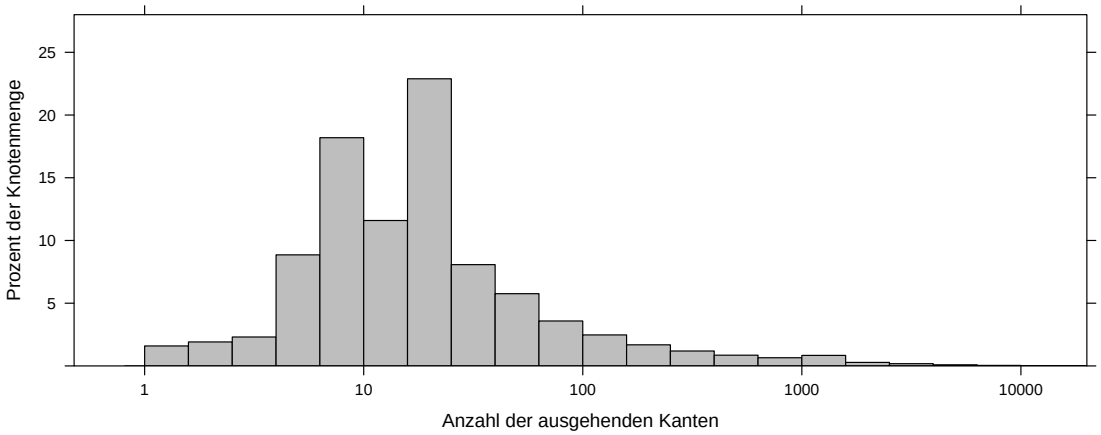


Abbildung 4.11: Histogramm der Kantenverteilung nach Integration der Tagging–Daten

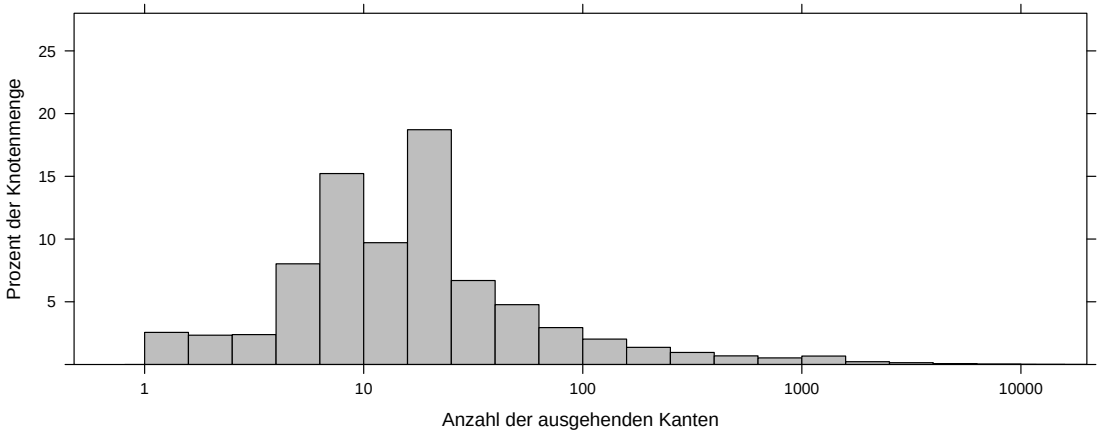


Abbildung 4.12: Histogramm der Kantenverteilung nach Integration der Clicktracking–Daten

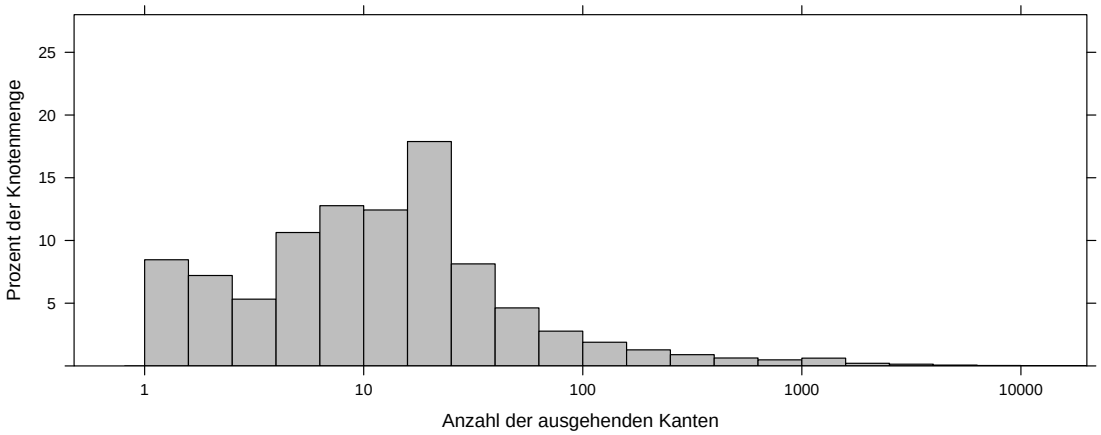


Abbildung 4.13: Histogramm der Kantenverteilung nach der Zerlegung von Wortgruppen

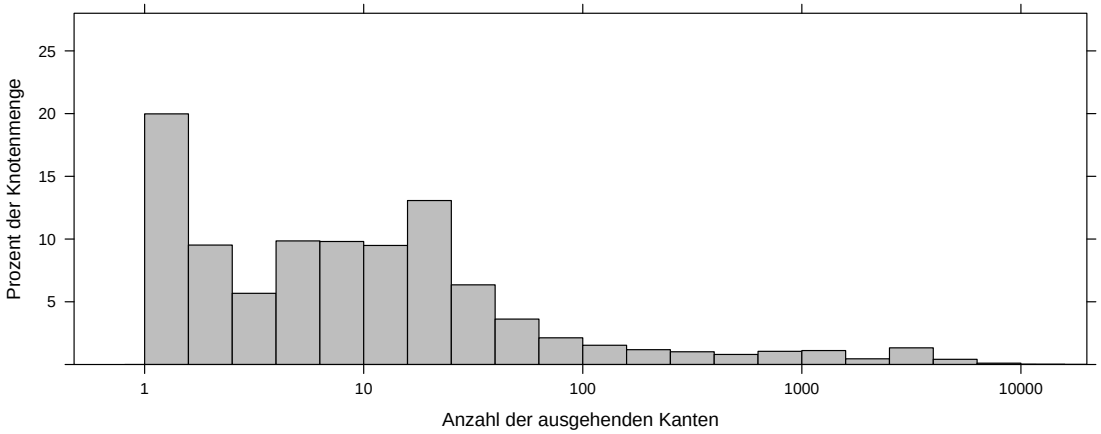


Abbildung 4.14: Histogramm der Kantenverteilung nach Integration der Wortschatz–Daten

Generell lässt sich festhalten, dass die Anzahl viel verbundener Knoten, gemessen an der Gesamtanzahl, relativ klein ist. Die Auswirkungen dessen hängen jedoch stark von der Anwendung der Daten ab und sind im Rahmen dieser Arbeit nicht beurteilbar.

Nach der quantitativen Auswertung der Link–Discovery–Schritte wird im nächsten Abschnitt die konkret durchgeführte Priorisierung der Beziehungen beschrieben.

4.6 Priorisierung der Beziehungen

Im Folgenden wird beschrieben, wie die in den Abschnitten 4.1 bis 4.4 durch Link Discovery erzeugten Zusammenhängen zwischen den Begriffen des Weltausschnittes priorisiert wurden. Der Prozess der Priorisierung folgt grundsätzlich dem in Abschnitt 3.6.2 skizzierten Prozess unter Zuhilfenahme evolutionärer Algorithmen. Dazu wird die Vorgehensweise und die Implementierung der Komponenten des evolutionären Algorithmus erläutert und die Ergebnisse ausgewertet.

Zum Zeitpunkt der Priorisierung befanden sich neun verschiedene Zusammenhangstypen im Weltausschnitt: *Tag–Kookkurrenz*, *Klick–Kookkurrenz*, *Kategorie–Kookkurrenz*, *Zusammensetzung*, *Wortform*, *Grundform*, *Synonym*, *Thesaurus–Beziehung* und *Zerlegung*.

Der Priorisierungsprozess versucht nun, diese Zusammenhangstypen gegeneinander zu gewichten, um für einen Anwendungsfall relevante Nachbarn eines Begriffes zu finden (siehe Abschnitt 3.2.4). Dazu muss jedem Typ ein relatives Gewicht zu den anderen Typen zugeordnet werden. Beziehungen, die durch Kookkurrenz berechnet wurden, besitzen bereits aufgrund der angegebenen Kookkurrenzmaße ein Kantengewicht. Jedoch muss hierbei festgelegt werden, welches Maß für das Kantengewicht herangezogen werden und in welchem Verhältnis zu den Gewichten anderer Kantentypen es stehen soll.

Somit wird zum Zwecke der Priorisierung eine Gewichtung von zwölf verschiedenen Parametern gesucht. Der nächste Abschnitt beschäftigt sich mit dem zur Priorisierung implementierten evolutionären Algorithmus.

4.6.1 Vorgehensweise

Zur Erläuterung der Vorgehensweise wird zum einen die Stichprobenauswahl, zum anderen die Komponenten Genotyp, Initialisierung, Selektion und Reproduktion des evolutionären Algorithmus (siehe Abschnitt 3.6) beschrieben.

Stichprobenauswahl

Zunächst soll die Methode zur Auswahl der Stichproben erläutert werden. Insgesamt wurden fünfzehn Knoten ausgewählt, deren Beziehungen optimiert werden sollen. Die Auswahl der Knoten richtete sich nach der Popularität von Suchbegriffen auf der Website von Spreadshirt. Dazu wurden alle Begriffe mit mehr als eintausend Suchen herangezogen und diese nach Häufigkeit der Suchen geordnet. Daraus wurden zufällig je fünf Begriffe bis zum unteren Quartil, fünf Begriffe zwischen unterem und oberen Quartil und fünf Begriffe über dem oberen Quartil ausgewählt. Die ausgewählten Begriffe, deren Kantengewichtungen lokal optimiert werden sollen, lauten: *Kopfkissenbezug*, *Student*, *Volkswagen*, *Marathon*, *Wow*, *Krankenschwester*, *Mountainbike*, *Hammer*, *Polska*, *Regenbogen*, *Minecraft*, *Kind*, *Dubstep*, *Leipzig* und *Valentinstag*.

Ziel dieser Auswahl war, eine möglichst vielfältige Verteilung der einzelnen Kantentypen zu erreichen, aus welcher sich nach Durchführung der Priorisierung möglicherweise Erkenntnisse ableiten lassen, ob die Priorisierung nur lokal oder auch global durchgeführt werden kann.

Nach Auswahl der Stichproben muss der Genotyp der am evolutionären Algorithmus teilnehmenden Individuen spezifiziert werden. Daraufhin sollten Komponenten des Algorithmus in ihrer Implementierung definiert werden.

Genotyp

Jeder Lösungskandidat wird durch die Werte der zwölf Parameter definiert, die die Gewichtung der Kantentypen untereinander beeinflussen. Daher wird jedes Individuum als Objekt repräsentiert, das zwölf Attribute besitzt. Davon sind neun Attribute reellwertig und drei Attribute vom Aufzählungstyp *Kookkurrenzmaß*.

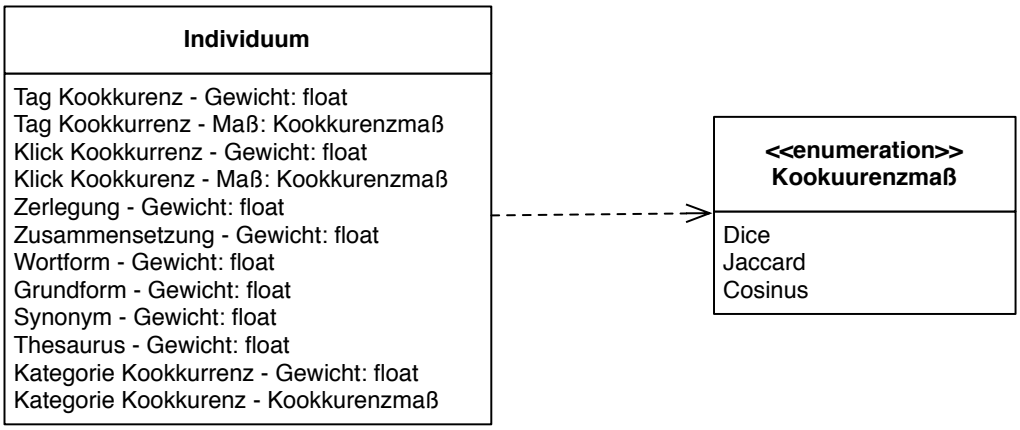


Abbildung 4.15: UML–Klassendiagramm des gewählten Genotyps

Die reellwertigen Attribute stellen ein Gewicht des jeweiligen Kantentyps dar. Sie können nur relativ zu den anderen Kantengewichten betrachtet werden und sind somit nur im Rahmen der Optimierung von Interesse. Der Wertebereich dieser Attribute beschränkt sich auf das Intervall $W = [0, 1] \in \mathbb{R}$. Die Kookkurrenzmaße entsprechen den in Abschnitt 3.3.2 genannten, also *Dice*, *Jaccard* und *Kosinus*. Der Genotyp ist in Abbildung 4.15 als UML–Klassendiagramm dargestellt. Listing 4.12 zeigt beispielhaft ein Individuum in JSON–Notation.

```
{
    "tagCoOccMeasure" : 0,
    "tagCoOccWeight" : 0.6878115427680314,
    "clickCoOccMeasure" : 2,
    "clickCoOccWeight" : 0.1533144270069897,
    "synonymWeight" : 0.2355003503616899,
    "thesaurusWeight" : 0.9918724503368139,
    "baseformWeight" : 0.2843543488997966,
    "wordformWeight" : 0.93317318148911,
    "sharedDomainsMeasure" : 2,
    "sharedDomainsWeight" : 0.462230023695156,
    "compositionWeight" : 0.07963851140812039,
    "decompositionWeight" : 0.4740817183628678
}
```

Listing 4.12: JSON–Beispiel für ein Individuum

Initialisierung

Im Initialisierungsschritt wird die anfängliche Population für jede der Stichproben gebildet. Dazu muss zuerst eine geeignet erscheinende Populationsgröße festgelegt werden.

In jeder Generation müssen im Selektionsschritt alle Individuen bewertet werden. Wie in Abschnitt 3.6.2 bereits erläutert wurde, kann dies nur mit menschlicher Interaktion erfolgen. Somit sollte die Populationsgröße möglichst gering sein, um mit der gleichen Anzahl Bewertungen eine größere Anzahl von Generationen zu durchlaufen. Dies geschieht auf Kosten der Diversität in der Population. Jedoch wurde im Rahmen dieser Arbeit dieser Ansatz gewählt, um möglichst viele Optionen zu haben, die Parameter zu optimieren.

Es wurde eine Populationsgröße von zehn Individuen je Stichprobe gewählt. Somit müssen in jeder Generation einhundertfünfzig Individuen bewertet werden. Bei der Initialisierung wurden die Variablen jedes erzeugten Individuums zufällig gewählt.

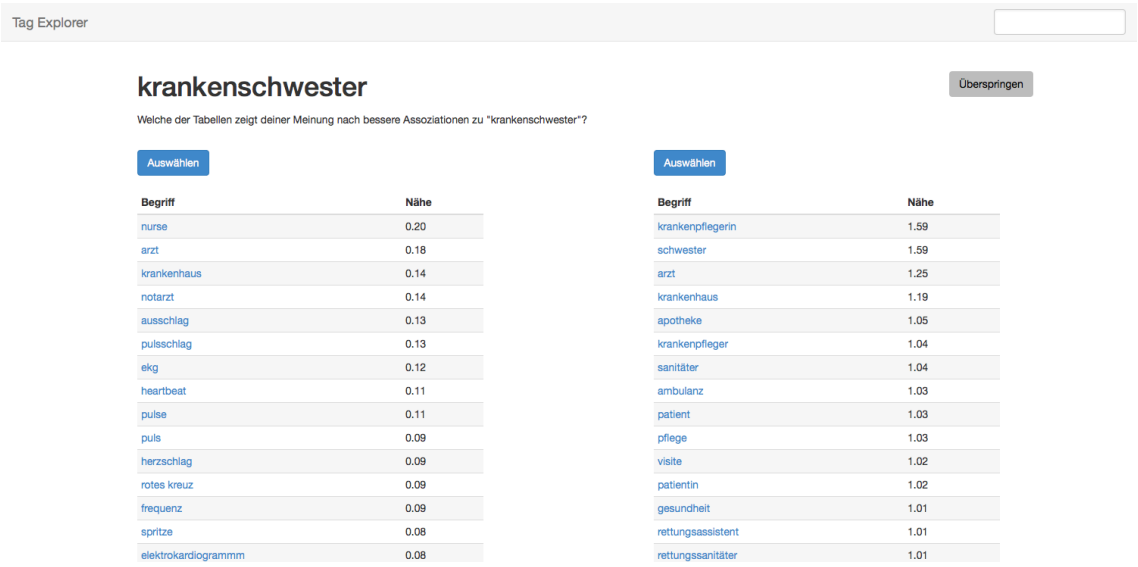
Selektion

Die Bestimmung einer Fitnessfunktion für die Optimierung der Link Discovery gestaltet sich durch die Notwendigkeit menschlicher Beurteilung als schwierig. Eine solche Fitnessbestimmung müsste derart erfolgen, dass ein Benutzer jedem Individuum einen reellwertigen Fitnesswert zuweist. Dies gestaltet sich jedoch auf Grund der hohen Anzahl von Individuen nicht praktikabel.

Aus diesem Grund werden die Individuen, die in die nächste Generation übernommen werden, direkt durch den bewertenden Benutzer umgesetzt. Die umgesetzte Selektion erfolgt durch die Durchführung von Wettkämpfen. Hierzu werden in jeder Generation je fünf Paare von Individuen gebildet. Diese Paarungen stellen die Wettkämpfe dar, bei denen der Benutzer den Gewinner bestimmt. Alle Gewinner werden selektiert und im Reproduktionsschritt weiter verwendet.

Der Vorteil dieses Vorgehens ist, dass der Benutzer nur jeweils zwei Lösungskandidaten vergleichen muss, anstatt direkt fünf der zehn Individuen auszuwählen. Somit verringert sich der zu einem Zeitpunkt zu leistende kognitive Aufwand für die Selektion.

Zur Selektion besucht der Benutzer eine Website, auf der ihm zwei Lösungskandidaten für eine Stichprobe präsentiert werden. Die Lösungskandidaten werden in Form von Listen von je fünfzehn Begriffen dargestellt, die die mit der jeweiligen Kantengewichtung erzeugten nächsten Nachbarn des Begriffes sind. Diese Oberfläche ist in Abbildung 4.16 als Screenshot abgebildet. Nach Auswahl eines Gewinners wird dem Nutzer der nächste Wettkampf präsentiert.



Begriff	Nähe
nurse	0.20
arzt	0.18
krankenhaus	0.14
notarzt	0.14
ausschlag	0.13
pulsschlag	0.13
ekg	0.12
heartbeat	0.11
pulse	0.11
puls	0.09
herzschlag	0.09
rotes kreuz	0.09
frequenz	0.09
spritze	0.08
elektrokardiogrammm	0.08

Begriff	Nähe
krankenpflegerin	1.59
schwester	1.59
arzt	1.25
krankenhaus	1.19
apotheke	1.05
krankenpfleger	1.04
sanitäter	1.04
ambulanz	1.03
patient	1.03
pflege	1.03
visite	1.02
patientin	1.02
gesundheit	1.01
rettungsassistent	1.01
rettungssanitäter	1.01

Abbildung 4.16: Screenshot der Oberfläche zur interaktiven Selektion

Am Selektionsprozess kann sich jeder interessierte Benutzer beteiligen. Dieses Vorgehen wurde gewählt, um ein möglichst breites Spektrum an Meinungen bezüglich der Güte der Verbindungen zu erhalten.

Durch die direkte Auswahl beträgt die Reproduktionswahrscheinlichkeit für die ausgewählten Individuen eins und für die nicht ausgewählten Individuen Null [Dre12]. Die gewählten Reproduktionsverfahren werden im nächsten Abschnitt beschrieben.

Reproduktion

Die fünf selektierten Individuen werden zur Reproduktion herangezogen, um fünf neue Individuen zu erzeugen, damit die Populationsgröße für die nächste Selektion wieder auf zehn Individuen steigt. Dazu werden die selektierten Individuen zuerst rekombiniert, um fünf Kindindividuen zu erzeugen, welche dann durch Mutation verändert werden.

Rekombination Der *Ein–Punkt–Crossover* [Wei07] wurde als Rekombinationsverfahren ausgewählt. Dazu werden zufällig zwei Elternindividuen ausgewählt und ein zufälliger Crossover–Punkt berechnet. Werden die Genotypen der beiden Elternindividuen als Arrays dargestellt, werden bis zum Crossover–Punkt die Variablen des ersten Individuums,

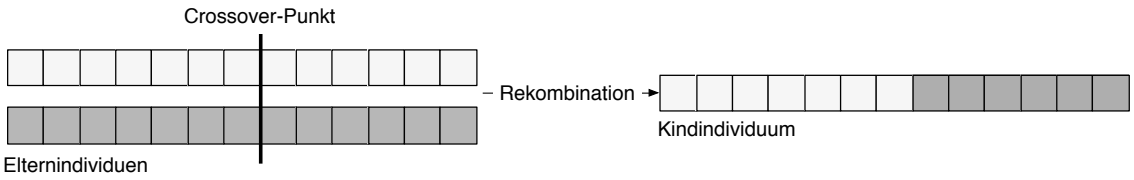


Abbildung 4.17: Ein–Punkt–Crossover

nach dem Crossover–Punkt die Variablen des zweiten Individuums übernommen. Der Crossover ist in Abbildung 4.17 dargestellt.

Mutation Zur Mutation der reellwertigen Kantengewichtungen wurde das Verfahren der *Gauß-Mutation* [Wei07] gewählt. Dabei wird zu jeder reellwertigen Variablen des Genotyps ein normalverteilter Zufallswert addiert. Verlässt der entstehende Wert das zulässige Intervall, so wird er auf die entsprechende Intervallschranke angepasst. Die Gauß-Mutation macht daher große Änderungen der Variable möglich, aber weniger wahrscheinlich als kleine Änderungen. Sie eignet sich somit gut, um Diversität in der Population zu erzeugen. Konkret wurde in der Umsetzung der Priorisierung eine auf das Intervall $(0, 1)$ angepasste Normalverteilung mit Erwartungswert $\mu = 0$ und Varianz $\sigma = 0.5$ gewählt. Die Variablen für die Kookkurrenzmaße werden nicht mutiert.

Nachdem in diesem Abschnitt die konkrete Umsetzung der Priorisierung mittels interaktiver evolutionärer Algorithmen beschrieben wurde, folgt im nächsten Abschnitt die Darstellung und Diskussion der Ergebnisse.

4.6.2 Ergebnisse

Die Priorisierung wurde über insgesamt 13 Generationen durchgeführt. Somit wurden über alle Stichproben hinweg 975 Selektionen von Benutzern vorgenommen. Die Benutzer waren allesamt Mitarbeiter von Spreadshirt. Aufgrund der anonymen Teilnahme lässt sich nicht angeben, wie viele Einzelpersonen an der Priorisierung teilgenommen haben.

Kantengewichte

Da nach jeder Selektion je Stichprobe fünf Individuen in der Population verbleiben, werden zunächst deren Kantengewichtungen untersucht. Tabelle 4.8 zeigt für jede Variable

aus dem Genotyp den Median der in der dreizehnten Generation selektierten Individuen. Das Kantengewicht für den Typ *Zerlegung* ist nicht in der Tabelle enthalten, da es sich bei den Stichproben nur um Einzelwörter handelt.

Stichprobe	k_t	k_{kl}	s	t	g	w	k_{ka}	zu
Dubstep	0.85	0.60	0.53	0.77	1.00	1.00	1.00	0.48
Hammer	0.81	0.56	0.88	0.45	1.00	0.89	0.00	0.59
Kind	0.58	0.22	0.84	1.00	0.66	0.00	0.01	0.00
Kopfkissenbezug	0.34	0.00	0.00	0.60	0.77	0.93	0.26	0.93
Krankenschwester	1.00	0.20	0.28	0.60	0.70	0.88	0.59	0.33
Leipzig	0.69	1.00	0.66	0.01	0.41	0.00	0.17	1.00
Marathon	0.62	0.70	0.23	0.41	0.69	1.00	0.00	0.95
Minecraft	0.71	0.34	0.31	0.38	0.74	1.00	0.00	1.00
Mountainbike	0.34	0.48	0.00	0.00	1.00	1.00	0.85	0.00
Polska	0.63	0.25	0.86	0.97	0.24	1.00	0.00	0.00
Regenbogen	1.00	0.43	0.36	0.47	0.00	0.26	0.61	0.52
Student	0.67	0.15	0.38	0.84	0.78	0.70	0.72	0.33
Valentinstag	1.00	0.44	0.80	0.24	0.53	0.58	0.70	1.00
Volkswagen	0.47	1.00	0.11	0.51	1.00	0.89	0.14	0.30
Wow	0.08	1.00	0.93	0.21	1.00	0.54	0.53	0.68

Tabelle 4.8: Mediane der Kantengewichte nach der finalen Selektion

k_t	Tag–Kookkurrenz
k_{kl}	Klick–Kookkurrenz
s	Synonyme
t	Thesaurus
g	Grundform
w	Wortform
k_{ka}	Kategorie–Kookkurrenz
zu	Zusammensetzung

Bei Betrachtung dieser Ergebnisse fällt auf, dass sich die Gewichtungen der Kanten von Stichprobe zu Stichprobe stark unterscheiden. Somit bestätigt sich die Vermutung, dass eine globale Priorisierung der Kantengewichtungen keine sinnvollen Ergebnisse erzielt. Jedoch ergaben sich für die Typen *Wortform* und *Grundform* in allen Stichproben relativ hohe Gewichte.

Kookkurrenzmaße

In Tabelle 4.9 sind für alle Stichproben die nach der letzten Selektion jeweils am häufigsten in der Population auftretenden Kookkurrenzmaße aufgeführt.

Stichprobe	Tags	Klicks	Kategorien
Dubstep	Jaccard	Dice	Dice
Hammer	Dice	Dice	Kosinus
Kind	Kosinus	Dice	Kosinus
Kopfkissenbezug	Dice	Dice	Kosinus
Krankenschwester	Jaccard	Dice	Dice
Leipzig	Jaccard	Jaccard	Kosinus
Marathon	Dice	Dice	Jaccard
Minecraft	Dice	Dice	Kosinus
Mountainbike	Kosinus	Jaccard	Jaccard
Polska	Jaccard	Dice	Kosinus
Regenbogen	Dice	Dice	Jaccard
Student	Dice	Dice	Dice
Valentinstag	Dice	Jaccard	Kosinus
Volkswagen	Kosinus	Dice	Dice
Wow	Jaccard	Dice	Dice

Tabelle 4.9: Häufigste Kookkurrenzmaße nach der finalen Selektion

Die auffälligste Beobachtung ist, dass für die Clicktracking–Kookkurrenz das Maß *Dice* bei 80 Prozent der Stichproben am häufigsten vorkommt. Für die beiden Typen der Tag– und Kategorie–Kookkurrenz ergeben sich keine so deutlich häufiger vorkommenden Maße.

Erzeugte Priorisierungen

Nach den quantitativen Ergebnissen der Priorisierung werden nachfolgend die konkret erzeugten Priorisierungen ausgewertet. In Tabellen 4.10 bis 4.12 sind die nach den Ergebnissen der Priorisierung fünf höchstgewichteten Nachbarn einiger Stichproben beispielhaft dargestellt. Dazu wurden die Mediane der Gewichte der nach der letzten Selektion enthaltenen Individuen gebildet. Als Kookkurrenzmaße wurden die jeweils am häufigsten auftretenden Maße für die Stichprobe gewählt. Die Nachbarn der übrigen Stichproben sind in Anhang A dargestellt. Diese Tabellen zeigen gleichzeitig die Ergebnisse der Link Discovery

mittels Integration und Anreicherung sowie die Ergebnisse der Priorisierung. Die inhaltliche Beurteilung der Qualität dieser Nachbarn hängt vom Anwendungszweck der Daten ab und kann somit im Rahmen dieser Arbeit nicht endgültig beurteilt werden. Jedoch lässt sich bei Betrachtung der Daten feststellen, dass für viele der Stichproben durchaus Nachbarn aufgelistet werden, die eine inhaltliche Nähe zum Ausgangsbegriff besitzen.

Begriff	Gewicht
kleinkind	2.08
säugling	2.07
junge	2.02
nachwuchs	1.9
dreikäsehoch	1.87

Tabelle 4.10: Nachbarn des Begriffes “Kind” nach der Priorisierung

Begriff	Gewicht
i heart leipzig	1.18
tshirts leipzig	1.17
t-shirts leipzig	1.17
leipzig stadt	1.09
deutschland leipzig	1.09

Tabelle 4.11: Nachbarn des Begriffes “Leipzig” nach der Priorisierung

Begriff	Gewicht
mountainbikes	1.02
fahrrad	0.98
rennrad	0.96
rad	0.93
gangschaltung	0.89

Tabelle 4.12: Nachbarn des Begriffes “Mountainbike” nach der Priorisierung

4.7 Zusammenfassung

In diesem Kapitel wurde am praktischen Beispiel die Anwendung des in Kapitel 3 beschriebenen Link–Discovery–Frameworks dargestellt. Dazu wurden die initiale Erstellung des

Weltausschnittes aus den Tagging–Daten von Spreadshirt, die Anreicherung mit den Daten des Clicktracking–Systems, durch Zerlegung von Wortgruppen und Integration des Wortschatzes der Universität Leipzig ausführlich erläutert. Für jeden dieser Schritte wurden die quantitativen Ergebnisse dargestellt und diskutiert. Außerdem wurde die quantitative Entwicklung des gesamten Weltausschnittes dargelegt. Anschließend wurden die erzeugten Beziehungen mittels eines interaktiven evolutionären Algorithmus priorisiert und die Ergebnisse präsentiert.

Nachdem die praktische Durchführung der Link Discovery beschrieben wurde, werden im folgenden Kapitel ausgewählte Aspekte der technischen Umsetzung erläutert.

5 Link–Discovery–System

Das folgende Kapitel beschäftigt sich mit ausgewählten Aspekten der technischen Umsetzung der in Kapitel 4 beschriebenen Link–Discovery–Durchführung. Dazu gehören die Formulierung der Anforderungen an das System, die Systemarchitektur sowie die getroffene Technologieauswahl und einige Implementierungsaspekte.

5.1 Anforderungen an das System

Dieser Abschnitt spezifiziert die Anforderungen an ein System, das zur Link Discovery eingesetzt werden kann. Die Anforderungen unterteilen sich hierbei in einen funktionalen und einen nichtfunktionalen Anteil.

5.1.1 Funktionale Anforderungen

Nachfolgend werden die funktionalen Anforderungen an das System aufgeführt. Diese ergeben sich direkt aus dem in Kapitel 3 beschriebenen Link–Discovery–Framework.

Datenimport Das System muss in der Lage sein, Rohdaten aus verschiedenen Datenquellen zu importieren und zu speichern. Dazu zählen die MySQL–Datenbank des Unternehmens Spreadshirt, die JSON–Dokumente des Clicktracking–Systems und die API des Wortschatzes der Universität Leipzig. Das System sollte außerdem erweiterbar sein, um in Zukunft andere Datenquellen einbinden zu können.

Datenspeicherung Das System muss die importierten Rohdaten, Zwischenergebnisse der Link Discovery und die Graphenrepräsentation des Weltausschnittes permanent speichern können.

Datenverarbeitung Das System muss die importierten Daten entsprechend des in Abschnitt 3.2 beschriebenen Link–Discovery–Prozesses verarbeiten können. Dazu zählen die Schritte Bereinigung, Reduktion, Transformation und Integration. Die Implementierung dieser Schritte sollte änderbar und erweiterbar sein. Im Rahmen des Transformationsschrittes muss das System in der Lage sein, Kookkurrenzmaße zu berechnen.

Visualisierung Das System soll eine Oberfläche für die Visualisierung der in der Graphenrepräsentation des Weltausschnittes gespeicherten Daten bieten. Dazu zählt einerseits die Darstellung der Begriffe, deren Kontext und Beziehungen und andererseits eine Möglichkeit der manuellen Priorisierung zum interaktiven Erkunden des Datenbestandes.

Priorisierung Das System muss den in Abschnitt 3.6 beschriebenen Priorisierungsprozess mittels evolutionärer Algorithmen implementieren. Dazu gehören Komponenten des Algorithmus aus Abschnitt 4.6.1 sowie eine Benutzeroberfläche zur interaktiven Selektion.

Programmierschnittstelle Das System muss eine Programmierschnittstelle (nachfolgend *API*) zur programmatischen Abfrage der Daten bereit stellen. Die angebotenen Daten enthalten die Begriffe, deren Kontexte und priorisierte Beziehungen. Die Priorisierung wird vom Benutzer der API spezifiziert.

5.1.2 Nichtfunktionale Anforderungen

Neben den funktionalen Anforderungen werden einige nichtfunktionale Anforderungen an das System gestellt. Diese werden im folgenden genannt.

Datenmenge Das System muss in der Lage sein, bis zu einer Milliarde Objekte zu speichern und zu verarbeiten. Diese Objekte können Rohdaten der Datenquellen, Zwischenergebnisse der Link Discovery oder Knoten und Kanten des Weltausschnittes sein.

Parallelisierbarkeit Um die Datenmenge verarbeiten zu können, sollten rechenintensive Berechnungsschritte auf mehrere Rechner verteilt werden können, um die Berechnung zu beschleunigen. Zu den anspruchsvolleren Berechnungsschritten zählt beispielsweise die Berechnung von Kookkurrenz aus den Daten des Tagging–Systems.

Antwortzeiten Das System sollte Anfragen, die die Graphenrepräsentation des Weltausschnittes betreffen, in unter einer Sekunde beantworten. Hierzu zählt vor allem die Anfrage der Nachbarn eines Begriffes, geordnet nach einer spezifizierten Priorisierung.

Nachdem die Anforderungen an das System formuliert wurden, werden in den folgenden Abschnitten die sich daraus ergebende Architektur, das Datenmodell und die Technologieauswahl erläutert.

5.2 Architektur des Systems

Aus den im vorherigen Abschnitt formulierten Anforderungen wurde die im Folgenden beschriebene Systemarchitektur entwickelt. Abbildung 5.1 zeigt die komplette Architektur des implementierten Link–Discovery–Systems. Darin sind, neben der Architektur des Systems selbst, alle genutzten Datenquellen und die Daten, die sie bereit stellen, aufgeführt.

Zentraler Bestandteil der Architektur ist das Datenbanksystem, das alle benötigten Daten speichert. Die Wahl des Datenbanksystems hat große Bedeutung für die Realisierung der funktionalen und nichtfunktionalen Anforderungen und wird in Abschnitt 5.3.1 diskutiert. Im Datenbanksystem werden die Rohdaten, Zwischenergebnisse und die Graphenrepräsentation des Weltausschnittes abgelegt. Das Datenmodell des Graphen folgt der Definition aus Abschnitt 3.4.2.

Für jeden Schritt der Integration von Datenquellen aus Abschnitt 3.2.1 existiert in der Architektur eine Komponente, die den jeweiligen Schritt für die entsprechende Datenquelle ausführt. Diese Komponenten kommunizieren direkt mit dem Datenbanksystem, um die Rohdaten oder Zwischenergebnisse zu lesen, führen die Berechnungen durch und speichern die Ergebnisse wiederum in die Datenbank.

Für die Abfrage der im Graphen gespeicherten Informationen existiert eine API, welche Informationen zu Knoten und deren Nachbarn per HTTP als JSON–Dokumente [Cro06]

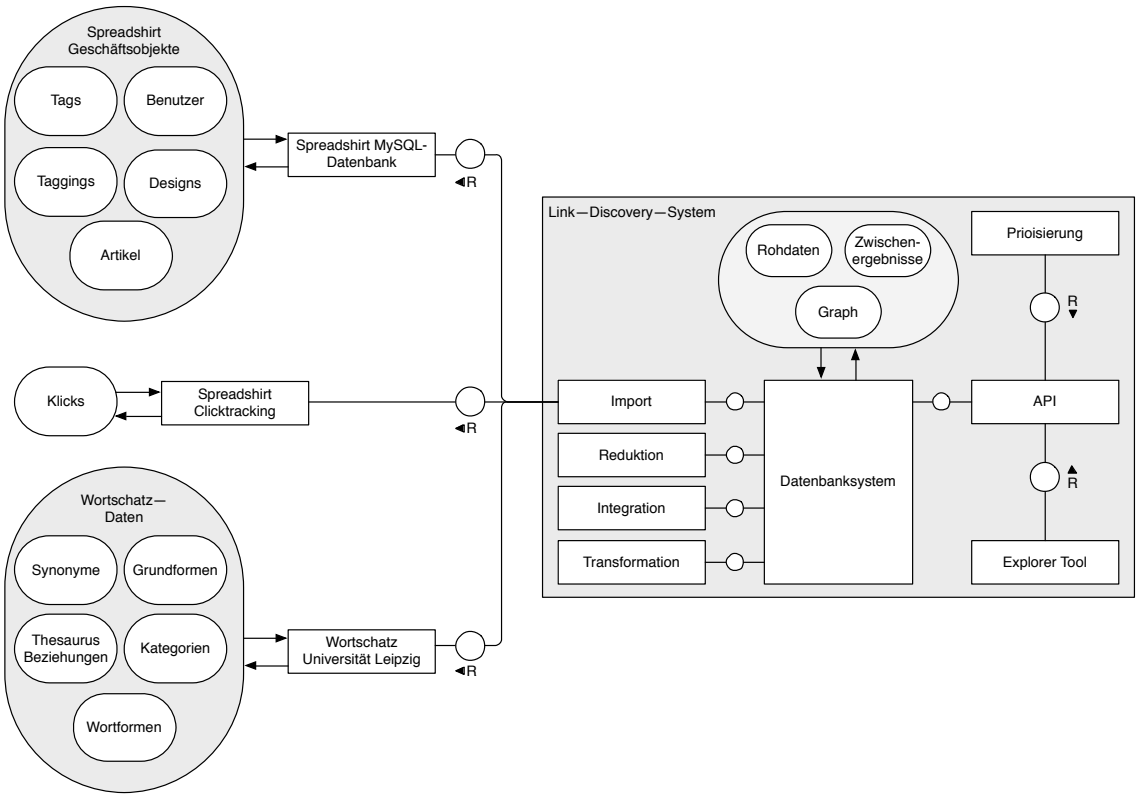


Abbildung 5.1: FMC–Blockdiagramm der gewählten Systemarchitektur

zur Verfügung stellt. Diese API kann für die Einbindung der erzeugten Informationen in andere Applikationen genutzt werden. Über Anfrageparameter kann die Gewichtung der einzelnen Kantentypen beeinflusst werden.

Zum Zeitpunkt der Bearbeitung dieser Arbeit existierten zwei Anwendungen, die die API des Link–Discovery–Systems nutzten. Dies sind der *Tag Explorer* und die Komponente zur Priorisierung der Beziehungen.

Beim Tag Explorer handelt es sich um eine Browseranwendung, die die im Graphen gespeicherten Beziehungen visualisiert und interaktiv erkundbar macht. Der Benutzer dieser Anwendung kann mit selbst gewählten Gewichtungen der Beziehungen den Graphen durchsuchen. Außerdem werden, wenn vorhanden, die Designs auf der Spreadshirt–Plattform angezeigt, die mit dem gewählten Begriff getaggt sind. Somit wurde die funktionale Anforderung, den Kontext zu einem Begriff zu visualisieren, nur zum Teil umgesetzt, da der Tag Explorer lediglich den Tagging–Kontext eines Begriffes darstellt. In Abbildung 5.2 ist ein Screenshot der Anwendung abgebildet.

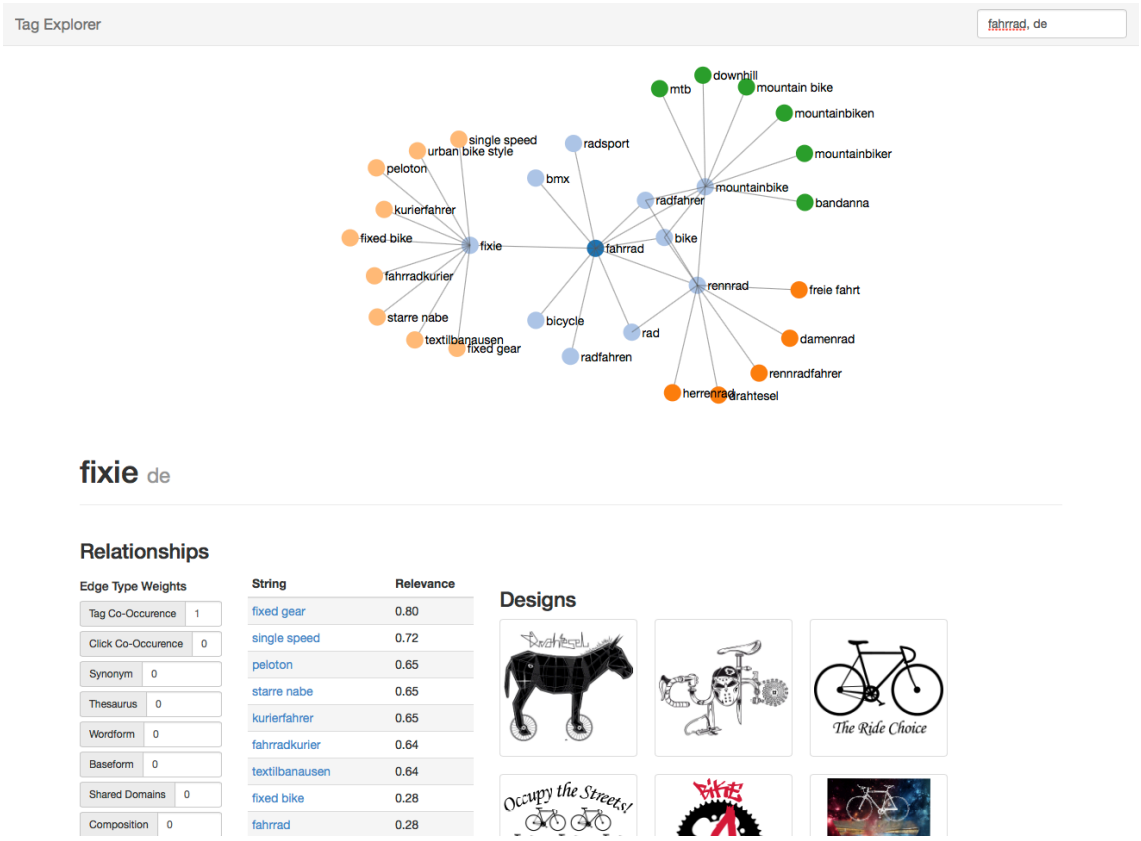


Abbildung 5.2: Screenshot der Anwendung "Tag Explorer"

Die zweite Anwendung, die die API des Systems nutzt, ist die in Abschnitt 4.6.1 beschriebene Umsetzung der Priorisierung. Da diese die Daten des Weltausschnittes nicht verändern muss, ist eine entkoppelte Anbindung über die API sinnvoll.

Nachdem die Architektur vorgestellt wurde, spezifiziert der nächste Abschnitt die konkreten Technologien, die zur Implementierung dieser Architektur ausgewählt wurden.

5.3 Technologieauswahl

Der folgende Abschnitt beschäftigt sich mit einer Auswahl der zur Umsetzung der Link Discovery eingesetzten Technologien. Die Kernelemente sind das Datenbanksystem, die konkrete Implementierung der Architekturkomponenten sowie die parallele Datenverarbeitung mittels des Programmiermodells MapReduce.

5.3.1 Datenbanksystem

Die Wahl des Datenbanksystems ist von zentraler Bedeutung für die Umsetzung der in Abschnitt 5.1 formulierten Anforderungen. Speziell die nichtfunktionalen Anforderungen der großen Datenmenge, der Parallelisierbarkeit und der kurzen Antwortzeiten stellen für traditionelle relationale Datenbanksysteme große Herausforderungen dar. Durch die starre relationale Form der Daten sind Schemaänderungen mit großem Aufwand verbunden. Die Verteilung von Daten über mehrere Server ist üblicherweise nicht grundsätzlich vorgesehen und kann nur mit größerem Implementierungsaufwand erreicht werden. Aufgrund dieser häufigen Limitierungen relationaler Datenbanksysteme wurde für die Implementierung des Link–Discovery–Systems das Datenbanksystem *MongoDB* [Mo] gewählt.

MongoDB

Bei MongoDB handelt es sich um eine quelloffene dokumentenorientierte Datenbank. Im Gegensatz zu traditionellen relationalen Datenbanksystemen verzichtet MongoDB auf eine tabellenförmige Struktur der Daten und speichert Datensätze in Form von so genannten *Dokumenten*. Dabei handelt es sich um hierarchische Schlüssel-/Wertpaare, die schemalos in so genannten *Collections* gespeichert werden. Schemalos bedeutet, dass die Dokumente innerhalb einer Collection nicht alle dieselbe Struktur besitzen müssen.

Zur Repräsentation der Dokumente verwendet MongoDB ein Format, das sich sehr an JSON [Cro06] anlehnt. JSON ist ein menschenlesbares Datenaustauschformat, das aus der Objektnotation der Programmiersprache JavaScript abgeleitet wurde. Das Datenformat von MongoDB ist BSON [BS13], eine binäre Repräsentation von JSON, die einige zusätzliche Datentypen unterstützt.

Listing 5.1 zeigt ein Beispiel für ein Dokument in MongoDB. Das Feld *_id* ist hierbei ein Bezeichner vom Typ *ObjectID*. Dieser stellt einen global eindeutigen Bezeichner dar, der benutzt werden kann, um Dokumente zu referenzieren. Innerhalb einer Collection muss *_id* grundsätzlich eindeutig sein. Das Feld *address* zeigt, dass Dokumente weitere Dokumente enthalten können. Am Feld *friends* wird deutlich, dass Werte für Schlüssel auch Arrays von Werten sein können. Diese sind nicht auf primitive Typen wie Zeichenketten oder Zahlen beschränkt, sondern können auch weitere Dokumente oder Arrays sein.

```
{
    "_id" : ObjectId("51efc20147cae77dfc02e0ac"),
    "name" : "Bob",
    "age": 25,
    "address": {
        "city": "Leipzig",
        "street": "Karl--Liebknecht--Str.␣132"
        "zip": "04277"
    },
    "friends" : [
        "alice",
        "fred",
        "jason"
    ]
}
```

Listing 5.1: JSON–Beispiel für ein Dokument in MongoDB

MongoDB unterstützt Anfragen über ein Binärprotokoll, welches über so genannte *Treiber* in vielen Programmiersprachen abstrahiert zur Verfügung steht. Dieses Protokoll unterstützt vielfältige Lese- und Schreiboperationen, die komplexe Abfragen und Operationen auf den gespeicherten Daten zulassen. Außerdem bietet MongoDB eine Implementierung des MapReduce–Programmiermodells (siehe Abschnitt 5.3.3) sowie die Möglichkeit, Indizes auf allen Hierarchieebenen der Dokumente zu nutzen. Für interaktive Operationen steht die *Mongo Shell* zur Verfügung, welche Abfragen mittels der Programmiersprache JavaScript erlaubt und somit einen Treiber für diese Sprache darstellt.

Aufgrund der genannten Eigenschaften stellt MongoDB einen exzellenten Ausgangspunkt für die Link Discovery im Rahmen dieser Arbeit dar. Durch die vorhandene Schemaflexibilität können die Daten in der gerade benötigten Form gespeichert und abgefragt werden. Durch die Unterstützung von MapReduce mit mehreren Rechnern lassen sich Berechnungen wie die der Kookkurrenz (siehe Abschnitt 5.3.3) parallelisieren und somit beschleunigen. Die Unterstützung von Indizes auf allen Hierarchieebenen der Dokumente bietet Vorteile zur effektiven Verkürzung von Antwortzeiten auf Anfragen.

MongoDB stellt das zentrale technische Element für die Link Discovery im Rahmen dieser Arbeit dar. Sobald die Daten aus den externen und internen Quellen in MongoDB importiert wurden, können die folgenden Schritte direkt mit Datenbankabfragen realisiert werden.

Umsetzung der Graphenrepräsentation

Um das in Abschnitt 3.4.2 beschriebene Datenmodell in MongoDB umzusetzen, muss es in eine Dokumentenform überführt werden. Dazu bietet es sich an, Knoten in Kanten in unterschiedlichen Collections zu speichern, um sie voneinander zu trennen.

Somit stellt sich anschließend die Frage, wie die Knoten und Kanten als Dokumente repräsentiert werden. Durch die durch MongoDB gegebene Schemaflexibilität lassen sich die Kontexte der durch die Knoten repräsentierten Begriffe direkt als Unterdokumente im Knotendokument ablegen. Der Schlüssel für diese Unterdokumente ist der Name des Kontextes. Dadurch lassen sich die Knoten leicht filtern, da die Abfrage auf das Vorhandensein des jeweiligen Schlüssels angepasst und durch Indizes auf diesen Schlüsseln unterstützt werden kann. Listing 5.2 zeigt ein Beispiel für einen Knoten in JSON–Notation. Arrays mit vielen Elementen sind aus Platzgründen verkürzt dargestellt.

```
{
    "_id" : ObjectId("51efc22447cae77dfc03e16b"),
    "language" : "de",
    "string" : "segeln",
    "tagProperties" : {
        "occurenceCount" : 4678,
        "articleCount" : 2347,
        "designCount" : 2331,
        "articleIDs" : [
            4961057,
            4977725,
            ...
        ],
        "designIDs" : [
            1645572,
            2216059,
            ...
        ]
    },
    "wortschatzProperties" : {
        "synonyms" : [
            "flattern",
            "fliegen",
            "gaukeln",
            ...
        ]
    }
}
```

Listing 5.2: JSON–Beispiel für ein Knotendokument in MongoDB

Die Kanten können direkt als Dokumente abgebildet werden. Über den Typ ergeben sich zusätzliche Eigenschaften. Ein Kantendokument für eine Tagging–Kookkurrenz ist beispielhaft in Listing 5.3 dargestellt.

```
{
        "_id" : ObjectId("51efd6f61177ff360605bd99"),
        "source" : ObjectId("51efc1af47cae77dfc00c3f8"),
        "target" : ObjectId("51efc1e047cae77dfc02087c"),
        "type" : "tag-co-occurence",
        "occurences" : 1,
        "dice" : 0.0001317089232795522,
        "jaccard" : 0.00006585879873551106,
        "cosine" : 0.008115343414514944
}
```

Listing 5.3: JSON–Beispiel für ein Kantendokument

5.3.2 Implementierung der Komponenten der Architektur

Die Komponenten zum Import, zur Bereinigung, Reduktion, Transformation und Integration wurden in der Programmiersprache JavaScript als Skripte für die Mongo Shell (siehe Abschnitt 5.3.1) umgesetzt. Diese Skripte implementieren die funktionalen Anforderungen an das System.

JavaScript wurde ausgewählt, da die Sprache eine natürliche Interaktion mit dem JSON–Format ermöglicht. Die Mongo Shell stellt alle Operationen für MongoDB für diese Programmiersprache zur Verfügung und eignet sich damit gut zur Kommunikation mit dem Datenbanksystem.

Die API wurde ebenfalls in JavaScript programmiert, unter Zuhilfenahme der Laufzeitumgebung *node.js* [NJS]. Diese ermöglicht die serverseitige Benutzung von JavaScript und ist demnach gut geeignet, um mit den Daten aus MongoDB zu arbeiten und diese als JSON–Dokumente an Clients auszuliefern. Der Tag Explorer sowie die Oberfläche zur interaktiven Selektion sind in JavaScript als Browseranwendungen implementiert.

Einzig die Importskripte für die MySQL–Datenbank von Spreadshirt und den Wortschatz der Universität Leipzig wurden in der Programmiersprache Ruby umgesetzt, da diese zum Zeitpunkt des Imports bessere Unterstützung für diese Datenquellen bot.

Zusammenfassend lässt sich festhalten, dass die Architekturkomponenten größtenteils als Skripte implementiert wurden, die direkt mit MongoDB kommunizieren. Dadurch kann das System bei Benutzung von weiteren Datenquellen einfach erweitert werden.

5.3.3 Datenverarbeitung

Um die nichtfunktionalen Anforderungen an das Link–Discovery–System umzusetzen, wurde zur Verarbeitung der Daten, speziell zur Kookkurrenzberechnung, das Programmiermodell MapReduce gewählt, da dieses die parallele Verarbeitung großer Datenmengen ermöglicht. Die Grundlagen dieses Modells sowie die Umsetzung der Kookkurrenzberechnung werden in den folgenden Abschnitten beschrieben.

Grundlagen von MapReduce

MapReduce [DG04] ist ein Programmiermodell für nebenläufige Verarbeitung und Erzeugung großer Datenmengen. Der Grundgedanke dieses Modells besteht in der Zerlegung der Berechnung in zwei Funktionen: *Map* und *Reduce*. Die Ein- und Ausgabedaten sind Schlüssel-/Wertpaare. Beide Funktionen werden vom Benutzer spezifiziert.

Die Map–Funktion dient zur Erzeugung von Zwischenergebnissen, welche ebenfalls in der Form von Schlüssel-/Wertpaaren vorliegen. Die Funktion wird einzeln auf jedes Paar der Eingabedaten angewandt und kann eine beliebige Anzahl von Zwischenergebnissen *emittieren*. Die MapReduce–Bibliothek gruppiert daraufhin alle Paare mit dem gleichen Schlüssel und übergibt diese an die Reduce–Funktion.

Die Reduce–Funktion wird somit jeweils auf einen Schlüssel und eine Liste von Werten angewandt. Ziel dieser Funktion ist, für jeden Schlüssel kein oder ein Ergebnis zurückzugeben. Die zu reduzierenden Werte werden für gewöhnlich als Iterator übergeben, um auch Datenmengen verarbeiten zu können, die nicht in den Arbeitsspeicher des Rechenknotens passen. Die Reduce–Funktion wird nur angewandt, wenn nach dem Map–Schritt mehr als ein Wert für einen Schlüssel emittiert wurde. Somit sollten Map- und Reduce–Funktion das gleiche Ausgabeformat besitzen. Das grundsätzliche Vorgehen von MapReduce ist in Abbildung 5.3 abgebildet.

Die MapReduce–Bibliothek übernimmt die Kommunikation zwischen den Knoten des Rechnerclusters. Dies hat den Vorteil, dass sich der Programmierer nur über die Umwandlung des zu lösenden Problems auf das Programmiermodell, nicht aber um dessen Implementierung über mehrere Rechner hinweg kümmern muss. Somit kann die verwendete Hardware vergleichsweise einfach an die zu verarbeitende Datenmenge oder die Bedürfnisse an die Rechengeschwindigkeit angepasst werden.

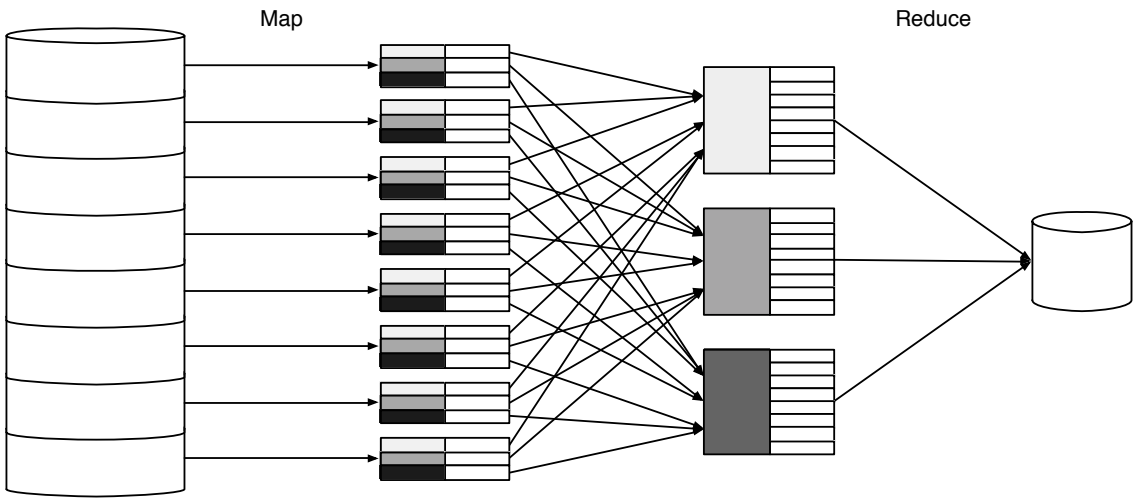


Abbildung 5.3: MapReduce–Prozess

In dieser Arbeit wurde die MapReduce Implementierung von MongoDB eingesetzt (siehe Abschnitt 5.3.1).

Kookkurrenzberechnung mit MapReduce

MapReduce kann für die Berechnung der Knoten und Kanten mittels Kookkurrenz genutzt werden. Dazu müssen für beide Operationen das Ein- und Ausgabeformat sowie die Funktionen Map und Reduce definiert werden.

Um die Berechnung zu vereinfachen, werden zuerst die Knoten erzeugt und mit allen Vorkommen der Begriffe annotiert. Somit kann daraufhin direkt aus der Knotenmenge die Kantenmenge erzeugt werden. Außerdem werden bei der Nutzung der Daten weniger Anfragen benötigt, um Informationen über einen Begriff selbst zu bekommen.

Berechnung der Knoten Als Eingabedaten für die Berechnung der Knotenmenge dienen Tupel der Form (d, t), wobei d ein Dokument und t einen Begriff darstellt. Die Map–Funktion wird nun auf jeden dieser Tupel angewandt und emittiert Schlüssel-/Wertpaare mit dem Begriff als Schlüssel und einer einelementigen Liste, die das Dokument des Tupels enthält sowie der Zahl 1 als Anzahl Vorkommen dieses Begriffs. Dieses Vorgehen ist notwendig, da die Ausgabe der Map- und Reduce–Funktionen das gleiche Datenformat haben sollten.

```
function map(document, term) {
    emit(term, {documents: [document], count: 1});
}

function reduce(term, values) {
    result = {documents: [], count: 0};
    foreach value in values do
        result.documents = concat(result.documents, value.documents);
        result.count = result.count + value.count;
    end
    return result;
}
```

Listing 5.4: Knotenerzeugung mit MapReduce

Die Reduce–Funktion fasst die einelementigen Listen zusammen, addiert die Vorkommen und erzeugt somit den Knoten, der für einen Begriff alle Dokumente, die mit diesem Begriff versehen wurden, sowie die Anzahl der Vorkommen insgesamt enthält.

Die Map- und Reduce–Funktionen für die Knotenberechnung sind als Pseudo–Code in Listing 5.4 dargestellt.

Berechnung der Kanten Die Berechnung der Kantenmenge kann mit den vorher berechneten Knoten als Eingabedaten erfolgen und wird in 2 Verarbeitungsschritte aufgeteilt. Zuerst werden die annotierten Knoten so umgeformt, dass zu einem Dokument alle vergebenen Begriffe bekannt sind. Im zweiten Schritt werden alle Paare von miteinander auftretenden Begriffen gebildet und die Ähnlichkeitsmaße berechnet.

Listing 5.5 zeigt die Umformung der Knoten mittels MapReduce. Als Eingabe für die Map–Funktion dienen die Knoten. Diese werden so umgeformt, dass für jedes Dokument, das am Knoten annotiert ist, ein neues Schlüssel-/Wertpaar emittiert wird. Die emittierten Ergebnisse werden in der Reduce–Funktion zusammengefasst, so dass als Ergebnis alle Begriffe, die an ein Dokument vergeben wurden, gesammelt als Liste vorliegen.

In Listing 5.6 wird die Erzeugung der Kookkurrenzkanten dargestellt. Im Map–Schritt werden dazu alle möglichen Paare der mit einem Dokument verknüpften Begriffe gebildet und emittiert. Der Schlüssel ist eine Kombination aus Ziel- und Quellbegriff. Der Wert zählt die Anzahl der Kookkurrenzen zwischen beiden Begriffen. Während des Reduce–Schrittes werden alle Kanten zwischen zwei Begriffen zusammengefasst, die Summe der Kookkurrenzen gebildet und die Ähnlichkeitsmaße berechnet. Die Funktionen zur Berechnung der Maße sind in Abschnitt 3.3.2 beschrieben.

```
function map(node) {
    foreach document in node.documents do
        emit(document, {terms: [node]});
    end
}

function reduce(term, values) {
    result = {terms: []};
    foreach value in values do
        result.terms = concat(result.terms, value.terms);
    end
    return result;
}
```

Listing 5.5: Umformung der Knoten mit MapReduce

```
function map(document) {
    foreach term1 in document.terms do
        foreach term2 in document.terms do
            emit({source: term1, target: term2}, {count: 1});
        end
    end
}

function reduce(edge, values) {
    result = {count: 0, dice: 0, jaccard: 0, cosine: 0};
    foreach value in values do
        result.count = result.count + value.count;
    end
    result.dice = dice(edge.source, edge.target, result.count);
    result.jaccard = jaccard(edge.source, edge.target, result.count);
    result.cosine = cosine(edge.source, edge.target, result.count);
    return result;
}
```

Listing 5.6: Kantenerzeugung mit MapReduce

MapReduce eignet sich somit gut zur Beschleunigung der Kookkurrenzberechnung, da diese in drei Schritten ohne sequenzielle Anteile auf die Funktionen Map und Reduce abgebildet werden kann, wie in diesem Abschnitt gezeigt wurde.

5.4 Zusammenfassung

Dieses Kapitel beschäftigte sich mit den Implementierungsaspekten des Systems, welches für die in Kapitel 4 beschriebene Link–Discovery–Durchführung im Rahmen dieser Arbeit zum Einsatz kam. Dazu wurden die funktionalen und nichtfunktionalen Anforderungen an das System formuliert. Aus diesen Anforderungen und dem Prozess zur Link Dis-

covery aus Abschnitt 3.2 wurde eine Architektur für das System abgeleitet und erläutert. Abschließend wurden einige Technologien, die zur Implementierung des Systems zum Einsatz kamen, diskutiert. Hierzu gehören MongoDB als Datenbanksystem, JavaScript als Sprache für die einzelnen Komponenten der Systemarchitektur und MapReduce als übergeordnetes Programmiermodell sowie dessen konkreter Einsatz zur Kookkurrenzberechnung.

6 Schlussbetrachtung

In dieser Masterarbeit wurde ein Verfahren und dessen praktische Durchführung zum Finden von Zusammenhängen zwischen Begriffen, die *Link Discovery*, beschrieben. Die Basis dafür stellten die Daten eines Tagging–Systems dar. Dazu wurden Tagging–Systeme grundlegend erläutert, deren Datenmodell definiert und die grundlegenden Unterschiede zwischen Folksonomies und geschlossenen Tagging–Systemen herausgearbeitet. Die Eigenschaften des im späteren Verlauf verwendeten Tagging–Systems von Spreadshirt wurden beschrieben und dessen Datenqualität in Hinblick auf Korrektheit, Vollständigkeit und Redundanzfreiheit diskutiert. Außerdem wurden die zu verarbeitenden Datenmengen definiert.

Für die Durchführung der Link Discovery wurde ein Framework definiert, welches die konzeptionelle Grundlage für die spätere Umsetzung darstellt. Dieses Framework modelliert den betrachteten Weltausschnitt, welcher Begriffe, den Kontext von Begriffen und deren Beziehungen untereinander enthält. Dieser Weltausschnitt wurde in eine Graphenrepräsentation überführt. Weiterhin wurde der Prozess der Link Discovery definiert und die einzelnen Schritte erläutert. Diese bestehen in der initialen Erstellung des Weltausschnittes, der Anreicherung durch Mining oder Integration weiterer Datenquellen und der Priorisierung der erzeugten Beziehungen. Die theoretischen Grundlagen von Kookkurrenz zur Beziehungserzeugung wurden definiert und die Berechnung veranschaulicht.

Weiterhin wurden mögliche Datenquellen diskutiert und die für die praktische Durchführung der Link Discovery in dieser Arbeit verwendeten Quellen ausgewählt. Diese bestehen aus dem Tagging– und Clicktracking–System von Spreadshirt sowie dem Wortschatz der Universität Leipzig. Zur Priorisierung der erzeugten Beziehungen wurden evolutionäre Algorithmen erläutert und deren Einsatz im Rahmen der Priorisierung definiert.

Diese theoretischen Grundlagen wurden anschließend an konkreten Daten praktisch umgesetzt. Für jede integrierte Datenquelle wurden die Schritte Import, Bereinigung, Reduktion, Transformation in die Graphenrepräsentation und Integration in den Weltausschnitt

ausführlich dargestellt und die quantitativen Ergebnisse präsentiert. Zur Anreicherung durch Mining wurde die Zerlegung von Wortgruppen in Einzelwörter erläutert. Im Anschluss wurden die quantitativen Veränderungen der Graphenrepräsentation nach jedem Link–Discovery–Schritt dargestellt und diskutiert. Nachdem alle Datenquellen integriert waren, wurde die praktische Umsetzung der Priorisierung der Beziehung mittels evolutionärer Algorithmen dargestellt und die Ergebnisse präsentiert.

Weiterhin wurden die Anforderungen an ein technisches System, welches die Link Discovery implementiert, formuliert und deren Umsetzung im Rahmen dieser Arbeit beschrieben. Dazu gehören die Wahl des Datenbanksystems MongoDB, die Implementierung der Komponenten in der Programmiersprache JavaScript und die Beschreibung von Kookkurrenzberechnung mittels des Programmiermodells MapReduce.

Die Ergebnisse dieser Arbeit stellen die Grundlage für weitere mögliche Arbeiten dar. Eine zukünftige Arbeit könnte sich mit der Auswertung der inhaltlichen Qualität der erzeugten Beziehungen beschäftigen. Dazu wird eine gründliche Analyse der integrierten Datenquellen und des Ergebnisses der Link Discovery mit Hilfe menschlicher Beurteilung benötigt.

Weiterhin sind Arbeiten denkbar, die die erzeugten Beziehungen für weitere Analyseverfahren nutzen. So könnten beispielsweise die Beziehungen zum Clustering der Begriffe zu Themen genutzt werden. Werden hierarchische Clusteringverfahren genutzt, können daraus Themenbäume und Topic Maps abgeleitet werden. Diese Themenbäume können sich durch ständige Durchführung der Link Discovery an aktuelle Trends in den Inhalten der betrachteten Website anpassen.

Der Link–Discovery–Prozess könnte durch einen interaktiven Trainingsschritt erweitert werden. In diesem Schritt wird die Beurteilung der Beziehungen durch Benutzer nicht nur zur Priorisierung, sondern zur direkten Veränderung des Weltausschnittes genutzt. Dabei werden Kanten eingefügt, die explizite statt nachträglich hergestellte Zusammenhänge beschreiben. Außerdem könnten durch manuellen Eingriff fehlerhafte Beziehungen gelöscht werden.

Insgesamt stellt das in dieser Arbeit beschriebene Framework eine gute Basis für Erweiterungen und neue Implementierungen dar. Das Verfahren ist flexibel und leistungsfähig genug, um auch auf anderen Datenbeständen gute Ergebnisse erzielen zu können. Durch

Integration anderer Datenquellen und neuer Analyseverfahren kann die Qualität der gefundenen Zusammenhänge stetig verbessert werden. Die konkret erzeugten Daten bieten eine gute Grundlage für weitere Auswertungen und praktische Anwendungen.

A Ergebnisse der Priorisierung

Begriff	Gewicht
sex drugs dubstep	0.58
dubstep beat	0.58
dubstep music	0.56
dubstep musik	0.56
dubstep london	0.56

Tabelle A.1: Nachbarn des Begriffes “Dubstep” nach der Priorisierung

Begriff	Gewicht
hämmern	1.53
schlägel	1.42
fäustel	1.38
hammers	1.34
klopfer	1.33

Tabelle A.2: Nachbarn des Begriffes “Hammer” nach der Priorisierung

Begriff	Gewicht
beautifulflower kopfkissenbezug	0.93
ellipses kopfkissenbezug	0.93
aurorae kopfkissenbezug	0.93
chessboard kopfkissenbezug	0.93
butterfly kopfkissenbezug	0.93

Tabelle A.3: Nachbarn des Begriffes “Kopfkissenbezug” nach der Priorisierung

Begriff	Gewicht
nurse	1.06
arzt	1.02
krankenpflegerin	0.99
schwester	0.92
krankenhaus	0.91

Tabelle A.4: Nachbarn des Begriffes “Krankenschwester” nach der Priorisierung

Begriff	Gewicht
gutenberg marathon	1.05
marathons	1
laufe marathon	0.98
berlin marathon	0.98
vienna city marathon	0.98

Tabelle A.5: Nachbarn des Begriffes “Marathon” nach der Priorisierung

Begriff	Gewicht
creeper minecraft	1.06
thegermanylp homies super minecraft pixel	1
creeper girl minecraft sexy	1
minecraft minetime mine craft	1
geek gamer lol minecraft portal diablo spiel game fun tasse g4me	1

Tabelle A.6: Nachbarn des Begriffes “Minecraft” nach der Priorisierung

Begriff	Gewicht
poland	1.24
polen	0.27
polish	0.17
revolução	0.15
rivoluzione	0.15

Tabelle A.7: Nachbarn des Begriffes “Polska” nach der Priorisierung

Begriff	Gewicht
blitz	3.07
regen	2.49
blau	2.48
wolke	2.48
rot	2.47

Tabelle A.8: Nachbarn des Begriffes “Regenbogen” nach der Priorisierung

Begriff	Gewicht
schüler	3.16
universität	2.35
hochschule	2.26
burschenschaft	2.18
college	2.17

Tabelle A.9: Nachbarn des Begriffes “Student” nach der Priorisierung

Begriff	Gewicht
valentinstag t shirt	1.06
alles gute zum valentinstag	1.03
valentinstag geschenk	1.02
ich hasse valentinstag	1.01
ich liebe valentinstag	1.01

Tabelle A.10: Nachbarn des Begriffes “Valentinstag” nach der Priorisierung

Begriff	Gewicht
volkswagen bus	1.1
volkswagen type 14	1.06
opel	1.05
bus volkswagen	1.04
volkswagen g60	1.04

Tabelle A.11: Nachbarn des Begriffes “Volkswagen” nach der Priorisierung

Begriff	Gewicht
limitededition cool edition limitierte edition limited limitiert	0.68
wow world of warcraft draenei	0.68
wow gilde world of warcraft respawn kelthuzad	0.68
chicken wow heiss chickeria	0.68
wow mom	0.68

Tabelle A.12: Nachbarn des Begriffes “Wow” nach der Priorisierung

Abbildungsverzeichnis

2.1 FMC–Entity–Relationship–Diagramm eines Tagging–Systems 7
2.2 FMC–Blockdiagramm der Spreadshirt–Bereiche und Benutzer 10

3.1 FMC–Entity–Relationship–Diagramm des Modells des Weltausschnittes . 16
3.2 FMC–Petri–Netz des Link–Discovery–Prozesses 17
3.3 FMC–Petri–Netz der Priorisierung 22
3.4 Repräsentation von Dokumenten als Mengen von Eigenschaften 23
3.5 Ungerichteter Graph 28
3.6 Gerichteter Graph 28
3.7 Gerichteter Multigraph 29
3.8 FMC–Entity–Relationship–Diagramm der Graphenrepräsentation 30
3.9 Beispiel–Graphausschnitt für das Ergebnis der Link Discovery 31
3.10 Ablauf evolutionärer Algorithmen 35
3.11 FMC–Petri–Netz der Priorisierung mittels evolutionärer Algorithmen . . 38

4.1 FMC–Entity–Relationship–Diagramm der Tagging–Quelldaten 42
4.2 Beispiel für das Zusammenführen der bereinigten Tags 44
4.3 Histogramm der Verteilung der Tagging–Kookkurrenz–Kanten 47
4.4 Beispiel für eine Spreadshirt–Suchergebnisseite 49
4.5 Histogramm der Verteilung der Clicktracking–Kookkurrenz–Kanten . . . 53
4.6 Beispielhafter Graphausschnitt nach der Zerlegung 55
4.7 Histogramm der Verteilung der Zerlegungs– und Zusammensetzungskanten 56
4.8 Reduktion der Wortschatz–Daten 60
4.9 Beispiel–Graphausschnitt der transformierten Wortschatz–Daten 61
4.10 Histogramm der Verteilung der Wortschatz–Kanten 62
4.11 Histogramm der Kantenverteilung nach Integration der Tagging–Daten . 65
4.12 Histogramm der Kantenverteilung nach Integration der Clicktracking–Daten 65

4.13 Histogramm der Kantenverteilung nach der Zerlegung von Wortgruppen 66
4.14 Histogramm der Kantenverteilung nach Integration der Wortschatz–Daten 66
4.15 UML–Klassendiagramm des gewählten Genotyps 69
4.16 Screenshot der Oberfläche zur interaktiven Selektion 71
4.17 Ein–Punkt–Crossover 72

5.1 FMC–Blockdiagramm der gewählten Systemarchitektur 80
5.2 Screenshot der Anwendung “Tag Explorer” 81
5.3 MapReduce–Prozess 87

Tabellenverzeichnis

4.1 Beispiele für die Tag–Bereinigung 43
4.2 Statistische Kennzahlen für die Verteilung der Tagging–Kookkurrenz–Kanten 48
4.3 Statistische Kennzahlen für die Verteilung der Clicktracking–Kookkurrenz–Kanten 54
4.4 Statistische Kennzahlen für die Verteilung der Zerlegungs– und Zusammensetzungskanten 56
4.5 Statistische Kennzahlen für die Verteilung der Wortschatz–Kanten 62
4.6 Entwicklung der Knoten– und Kantenanzahl nach jedem Link–Discovery–Schritt 63
4.7 Statistische Kennzahlen der Kantenverteilung nach jedem Link–Discovery–Schritt 64
4.8 Mediane der Kantengewichte nach der finalen Selektion 73
4.9 Häufigste Kookkurrenzmaße nach der finalen Selektion 74
4.10 Nachbarn des Begriffes “Kind” nach der Priorisierung 75
4.11 Nachbarn des Begriffes “Leipzig” nach der Priorisierung 75
4.12 Nachbarn des Begriffes “Mountainbike” nach der Priorisierung 75

A.1 Nachbarn des Begriffes “Dubstep” nach der Priorisierung 95
A.2 Nachbarn des Begriffes “Hammer” nach der Priorisierung 95
A.3 Nachbarn des Begriffes “Kopfkissenbezug” nach der Priorisierung 95
A.4 Nachbarn des Begriffes “Krankenschwester” nach der Priorisierung . . . 96
A.5 Nachbarn des Begriffes “Marathon” nach der Priorisierung 96
A.6 Nachbarn des Begriffes “Minecraft” nach der Priorisierung 96
A.7 Nachbarn des Begriffes “Polska” nach der Priorisierung 96
A.8 Nachbarn des Begriffes “Regenbogen” nach der Priorisierung 97
A.9 Nachbarn des Begriffes “Student” nach der Priorisierung 97
A.10 Nachbarn des Begriffes “Valentinstag” nach der Priorisierung 97

A.11 Nachbarn des Begriffes “Volkswagen” nach der Priorisierung 97
A.12 Nachbarn des Begriffes “Wow” nach der Priorisierung 98

Listings

3.1 Kookkurrenzberechnung 27

4.1 JSON–Beispiel für einen importierten Tag 42
4.2 JSON–Beispiel für ein importiertes Tagging 43
4.3 JSON–Beispiel für die denormalisierten Tagging-Daten 45
4.4 JSON–Beispiel für einen aus den Tagging–Daten erzeugten Knoten 46
4.5 JSON–Beispiel für eine aus den Tagging–Daten erzeugte Kante 46
4.6 JSON–Beispiel für ein Clicktracking–Rohdokument 50
4.7 JSON–Beispiel für die Bereinigung der Clicktracking–Dokumente 51
4.8 JSON–Beispiel für ein aus den Clicktracking–Daten erzeugtes Knotenobjekt 52
4.9 JSON–Beispiel für ein aus den Clicktracking–Daten erzeugtes Kantenobjekt 52
4.10 JSON–Beispiel für Rohdaten aus dem Wortschatz 58
4.11 JSON–Beispiel für die Umformung der Grundform eines Wortes 59
4.12 JSON–Beispiel für ein Individuum 69

5.1 JSON–Beispiel für ein Dokument in MongoDB 83
5.2 JSON–Beispiel für ein Knotendokument in MongoDB 84
5.3 JSON–Beispiel für ein Kantendokument 85
5.4 Knotenerzeugung mit MapReduce 88
5.5 Umformung der Knoten mit MapReduce 89
5.6 Kantenerzeugung mit MapReduce 89

Literatur

[AC04] Jean Aitchison und Stella Clarke Dextre. „The Thesaurus: A Historical Viewpoint, with a Look to the Future“. In: *Cataloging & Classification Quarterly* 37.3 (2004), S. 5–21.

[BS13] *BSON*. URL: http://bsonspec.org (besucht am 03. 11. 2013).

[Cro06] Douglas Crockford. *The application/json Media Type for JavaScript Object Notation (JSON)*. 2006. URL: http://tools.ietf.org/html/rfc4627 (besucht am 03. 11. 2013).

[De 06] Kenneth De Jong. *Evolutionary Computation: A Unified Approach*. Cambrigde: MIT Press, 2006.

[Del] Avos Systems. *Delicious*. URL: http://www.delicious.com (besucht am 24. 11. 2013).

[DG04] Jeffrey Dean und Sanjay Ghemawat. „MapReduce: Simplified Data Processing on Large Clusters“. In: *OSDI'04: Proceedings the the 6th Conference on Symposium on Operating Systems Design and Implementation*. USENIX Association, 2004.

[Dic] Dictionary.com. *Thesaurus.com*. URL: http://www.thesaurus.com (besucht am 20. 11. 2013).

[Dic45] Lee R. Dice. „Measures of the Amount of Ecologic Association Between Species“. In: *Ecology* 26.3 (Juli 1945), S. 297–302.

[Die12] Reinhard Diestel. *Graph Theory*. 4. Aufl. Bd. 173. Graduate Texts in Mathematics. Springer, 2012.

[Dre12] Stephan Dreyer. „Interaktive Evolution zur Assistenz bei der Einrichtungsplanung“. Masterarbeit. Fachhochschule Brandenburg, 2012.

[Fel98] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. Cambridge, Massachusetts: MIT Press, 1998.

[Flr] Yahoo. *Flickr*. URL: http://www.flickr.com (besucht am 24. 11. 2013).

[GKK04] Ingrid Gerdes, Frank Klawonn und Rudolf Kruse. *Evolutionäre Algorithmen*. Computational Intelligence. Vieweg+Teubner Verlag, 2004.

[Hey11] Gerhard Heyer. „Learning Semantic Relations from Text". In: *Modeling, Learning and Processing of Text Technological Data Structures*. Berlin: Springer, 2011.

[HK03] Thomas Herbst und Michael Klotz. *Lexikografie*. Paderborn: Schöningh, 2003.

[HKP12] Jiawei Han, Micheline Kamber und Jian Pei. *Data Mining. Concepts and Techniques*. 3. Aufl. Waltham, MA: Morgan Kaufmann, 2012. ISBN: 978-0-123-81479-1.

[Jac04] Elin K. Jacob. „Classification and Categorization: A Difference that Makes a Difference". In: *Library Trends* 52 (2004), S. 2004.

[Jac12] Paul Jaccard. „The Distribution of Flora in the Alpine Zone". In: *New Phytologist* 11.2 (1912), S. 37–50.

[KGT06] Andreas Knoepfel, Bernhard Groene und Peter Tabeling. *Fundamental Modeling Concepts. Effective Communication of IT Systems*. Hoboken, New Jersey: Wiley, 2006.

[KSS10] Kathrin Knautz, Simone Soubusta und Wolfgang G. Stock. „Tag Clusters as Information Retrieval Interfaces". In: *Proceedings of the 43rd Hawaii International Conference on System Sciences*. Hawaii, 2010.

[Lev66] Vladimir I. Levenshtein. „Binary Codes Capable of Correcting Deletions, Insertions and Reversals". In: *Soviet Physics Doklady* 10 (1966), S. 707.

[LK07] David Liben-Nowell und Jon Kleinberg. „The Link-Prediction Problem for Social Networks". In: *Journal of the American Society for Information Science and Technology* 58.7 (2007), S. 1019–1031.

[Mat04] Adam Mathes. „Folksonomies—Cooperative Classification and Communication through Shared Metadata". In: *Computer Mediated Communication* 47.10 (2004). URL: http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html (besucht am 22. 11. 2013).

[Mo] MongoDB Inc. *MongoDB*. URL: http : / / www . mongodb . org (besucht am 03. 11. 2013).

[NA11] Axel-Cyrille Ngonga Ngomo und Sören Auer. „LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data“. In: *Proceedings of IJCAI*. 2011.

[Nab] Daniel Naber. *OpenThesaurus*. URL: http://www.openthesaurus.de (besucht am 20. 11. 2013).

[NJS] Joyent Inc. *node.js*. URL: nodejs.org (besucht am 30. 11. 2013).

[Ols] Jack E. Olsen. *Data Quality. The Accuracy Dimension*. 1. Aufl. Burlington, Massachusetts: Morgan Kaufmann. ISBN: 978-1558608917.

[OW06] Ido Omer und Michael Werman. „Image Specific Feature Similarities“. In: *Computer Vision – ECCV 2006*. Bd. 3952. Lecture Notes in Computer Science. Heidelberg: Springer, 2006, S. 321–333.

[PS02] Viktor Pekar und Steffen Staab. „Taxonomy Learning: Factoring the Structure of a Taxonomy into a Semantic Classification Decision“. In: *Proceedings of the 19th International Conference on Computational Cinguistics*. COLING '02. Taipei: Association for Computational Linguistics, 2002, S. 1–7.

[Res95] Philip Resnik. „Using Information Content to Evaluate Semantic Similarity in a Taxonomy“. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. IJCAI'95. Montreal: Morgan Kaufmann, 1995, S. 448–453.

[RW13] Andrei Beliankou. *wlapi*. URL: http://rubygems.org/gems/wlapi (besucht am 07. 11. 2013).

[Sch06] Patrick Schmitz. „Inducing Ontology from Flickr Tags“. In: *Proceedings of the Collaborative Web Tagging Workshop at WWW2006*. Edinburgh, 2006.

[Shi05] Clay Shirky. *Ontology is Overrated. Categories, Links, and Tags*. Apr. 2005. URL: http://shirky.com/writings/ontology_overrated.html (besucht am 13. 08. 2013).

[SI13] W. Scholze-Stubenrecht und Dudenredaktion (Bibliographisches Institut). *Duden: Die deutsche Rechtschreibung*. 26. Aufl. Berlin: Bibliographisches Institut, 2013.

[Sør48] Thorvald Sørensen. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons*. Biologiske Skrifter. I kommission hos E. Munksgaard, 1948.

[Sprd] sprd.net AG. *Spreadshirt*. URL: http : / / www . spreadshirt . de (besucht am 04. 11. 2013).

[Ste10] Angus Stevenson. *Oxford Dictionary of English*. 3. Aufl. Oxford: Oxford University Press, 2010.

[Tak01] Hideyuki Takagi. „Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation". In: *Proceedings of the IEEE* 89.9 (2001), S. 1275–1296.

[Tan05] Pang-Ning Tan. *Introduction to Data Mining*. Boston: Addison-Wesley, 2005.

[Tas+03] Ben Taskar u. a. „Link Prediction in Relational Data". In: *Neural Information Processing Systems*. 2003.

[Tve77] Amos Tversky. „Features of similarity". In: *Psychological Review* 84 (1977), S. 327–352.

[Van07] Thomas Vander Wal. *Folksonomy Coinage and Definition*. 2. Feb. 2007. URL: http://vanderwal.net/folksonomy.html (besucht am 24. 11. 2013).

[Vol+09] Julius Volz u. a. „Discovering and Maintaining Links on the Web of data". In: *The Semantic Web-ISWC 2009*. Springer, 2009, S. 650–665.

[Wei07] Karsten Weicker. *Evolutionäre Algorithmen*. Leitfäden der Informatik. Vieweg+Teubner Verlag, 2007.

[Wei08] Thomas Weise. *Global Optimization Algorithms. Theory and Application*. 2008. URL: http://www.it-weise.de/projects/book.pdf.

[Wor] WordNet. *Princeton University*. URL: http://wordnet.princeton.edu (besucht am 20. 11. 2013).

[WSL] Universität Leipzig, Institut für Informatik, Abteilung Sprachverarbeitung. *Deutscher Wortschatz*. URL: http://wortschatz.informatik.uni-leipzig.de (besucht am 21. 10. 2013).

[ZZ11] Norman Zänker und Christian Zietzsch. *Text Mining und dessen Implementierung*. Diplomica Verlag, 2011.