Technische Universität Darmstadt

Fachbereich Informatik

Diplomarbeit

zur Erlangung des akad. Grades eines

Diplom-Informatikers (Dipl.-Inform.)

Vergleich von Methoden zur Hypertext-Klassifikation

Prof. Dr. Johannes Fürnkranz

Fachgebiet Knowledge Engineering

vorgelegt von:

Stefan Seiermann

In der Quelle 51

63303 Dreieich

Matrikel Nr: 725314

Kurzfassung/Abstract

Als Hypertext-Klassifikation bezeichnet man die Einteilung von Webdokumenten in bestehende Kategorien. Die Klassifikatoren nutzen hierbei die Attribute verlinkter Dokumente durch unterschiedliche Methoden. In dieser Diplomarbeit wurden Methoden untersucht, welche die Inhalte sowie die Klassen der Nachbarschaft eines Dokumentes nutzen. Für die Evaluierung wurden Sammlungen von Hypertext-Dokumenten unter verschiedenen Gesichtspunkten generiert. Ein Vergleich der Methoden in diesen Sammlungen lieferte ein Ranking hinsichtlich der Vorhersagegenauigkeit. Hierbei wurde festgestellt, dass Methoden, welche die Ko-Zitierung (IO-Bridges) oder relevante Textelemente der Nachbarschaft (Link-Locals) verwenden, die höchste Vorhersagegenauigkeit aufweisen. Eine Kombination der Methoden erbrachte eine nochmalige Erhöhung der Vorhersagegenauigkeit. Als Ergebnis wurde ein Klassifikator implementiert, der (bei einer vollständigen Coverage) eine deutliche Steigerung der Vorhersagegenauigkeit gegenüber einem reinen Text-Klassifikator aufweist.

Schlagwörter: Text-Klassifikation, Hypertext, Web-Mining, Benchmarks

Hypertext classification is defined as the classification of web documents into existing categories. For this process, the classifiers use attributes of linked documents with different methods. This thesis investigates methods using contents and categories of adjacency of documents. For the evaluation compilations of hypertext documents were generated focussing on varying parameters. A comparison of the methods for these compilations delivered a ranking in terms of prediction accuracy. The result is that methods using co-citation (IQ bridges) or relevant text elements of adjacency (link locals) show the highest accuracy. A combination of methods resulted in an accuracy increases. As a result a classifier was implemented (including complete coverage) showing a marked accuracy increase as compared to a mere text classifier.

Keywords: Text-Classification, Hypertext, Web-Mining, Benchmarks

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Das Internet stellt heutzutage eine immer größer werdende Quelle für Informationen aller Art dar. In einer Studie wurde der Umfang des Internets auf ca. 2,1 Mrd. Seiten geschätzt, bei einem täglichen Zuwachs von ca. 7,1 Mio. Seiten (Murrey & More, 2000).

Aufgrund der Vielzahl der potentiellen Informationen ergibt sich die schwierige Aufgabe, dass relevante Informationen aus der Menge von Daten gefunden werden müssen. Für die Suche nach relevanten Informationen im Internet haben sich zwei verschiedene Verfahren etabliert, welche durch entsprechende Programme abgedeckt werden.

Suchmaschinen, wie z. B. Google[1] oder AltaVista[2], liefern bei der Eingabe von Suchbegriffen mögliche Seiten, welche die Informationen enthalten, die ein Anwender sucht. Problematisch hierbei ist, dass das Ergebnis einer Suche sehr stark von der Qualität der eingegebenen Suchbegriffe abhängig ist.

Webverzeichnisse, wie z. B. Yahoo[3] oder das Open-Directory-Project (ODP)[4], basieren auf dem Aufbau von hierarchischen Ontologien; sie stellen eine Art Inhaltsverzeichnis (von Teilen) des Internets dar (Abbildung 1.1):

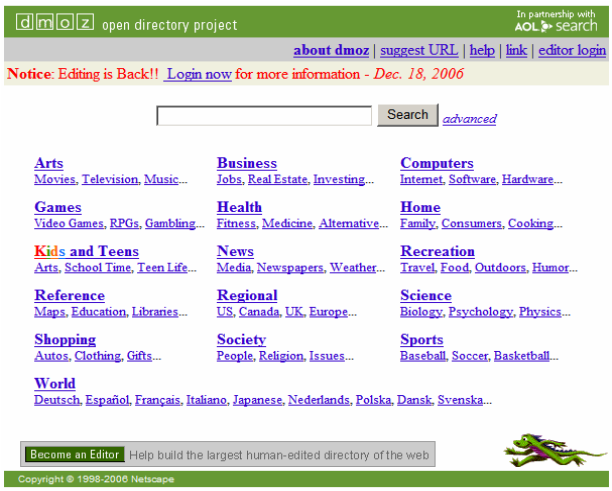

Abbildung 1.1: Web-Verzeichnis: „Open-Directory-Project“ (ODP)

[1] http://www.google.com
[2] http://www.altavista.com
[3] http://www.yahoo.com
[4] http://dmoz.org

Diese Webverzeichnisse werden heutzutage meist manuell gepflegt. Als Probleme hierbei erweisen sich der hohe Aufwand bei der Erstellung sowie die Aktualität der erstellten Verzeichnisse. Auch ist eine Einordnung einer Webseite in eine bestehende Ontologie sehr stark von dem jeweiligen Betreiber dieser Verzeichnisse abhängig.

Eine Lösung dieser Probleme bietet der Einsatz von Zuordnungsvorschriften, welche eine Webseite in eine bestehende Kategorie einordnet. Diesen Vorgang bezeichnet man als Klassifikation.

1.2 Klassifikation von Webseiten

Aufgabe einer Klassifikation von Webseiten ist es, anhand einer vorgegebenen Menge von bekannten Trainigsdokumenten ein Modell aufzubauen (1), mit dessen Hilfe sich unbekannte Dokumente aufgrund ihrer Eigenschaften in Klassen einteilen lassen (2). In Abbildung 1.2 werden diese Schritte veranschaulicht:

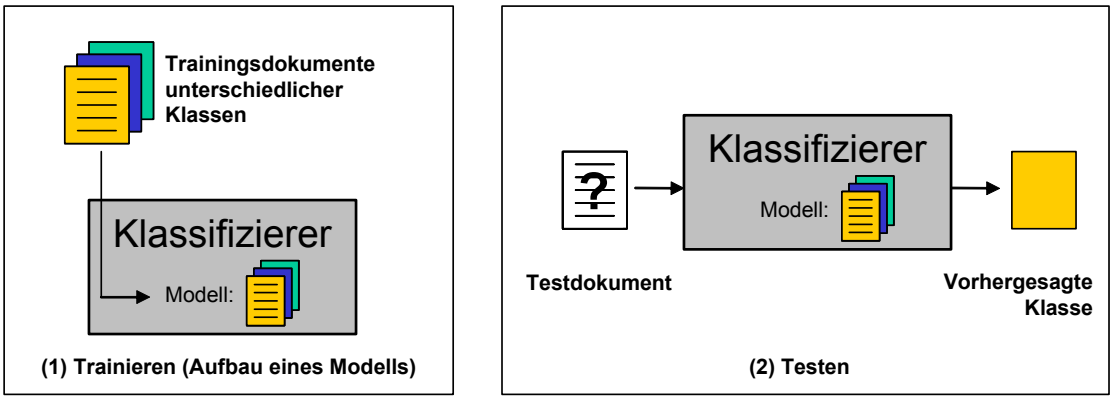

Abbildung 1.2: Klassifikation von Webseiten durch Trainieren (1) und Testen (2)

Hierbei sind sowohl die Klassen als auch die Zugehörigkeit der Trainingsdokumente zu einer Klasse vorher bekannt. Man spricht in diesem Zusammenhang vom überwachten Lernen (*supervised learning*). Falls nur Teile der Klasseninformationen vorliegen, spricht man von *semi supervised learning* (Chakrabarti, 2000).

1.2.1 Text-Klassifikation

Jedes Textdokument besteht aus einer Menge von Wörtern (*Features*). Die bedeutungsmäßigen Inhalte eines Dokumentes können nach Luhn (1958) im Wesentlichen durch eine geeignete Auswahl von Features, durch ihre Häufigkeiten, abgebildet werden. Hierdurch erfolgt eine Transformation der Semantik eines Dokumentes in eine maschinenlesbare Form (*bag of words*).

Bei der Text-Klassifikation werden für jedes Trainingsdokument die Häufigkeiten der enthaltenen Features „trainiert“. Aus diesen Häufigkeiten wird das Modell des

Klassifikators aufgebaut (1). Beim Testen eines Dokumentes wird dessen Häufigkeitsverteilung mit dem trainierten Modell verglichen (2).

Durch die Verwendung von Synonymen (unterschiedliche Wörter mit gleicher Bedeutung) bzw. Polynymen (unterschiedliche Bedeutung gleicher Wörter) kann die Zuordnung eines Dokumentes zu einer Klasse erschwert werden.

Eine Klassifikation des Testdokumentes erfolgt aufgrund der Auswahl der Klasse, welche die höchste Wahrscheinlichkeit aufweist (vgl. Abschnitt 6.1:Der Naive-Bayes-Klassifikator). Diese Methode liefert den *Basiswert* für Vergleiche mit anderen Methoden (vgl. Abschnitt 5.3:Vergleichsmethodik).

1.2.2 Hypertext-Klassifikation

Methoden der Text-Klassifikation beschränken sich auf den Text, der in den Dokumenten selbst zu finden ist. Für Webdokumente ist dieser Ansatz problematisch, da sich oft nur wenig Text auf den Webseiten findet. Für eine erfolgreiche Klassifikation ist man hier auf zusätzliche Informationen angewiesen. Auch bei der Verwendung von Synonymen und Polynymen wäre es hilfreich, den „Kontext“ einer Webseite zu ermitteln, wie z. B. durch die Verwendung von Informationen verlinkter Dokumente.

Durch die Nutzung verlinkter Dokumente kann eine beträchtliche Erhöhung der Vorhersagegenauigkeit einer Klassifikation erreicht werden. Diese Vorgehensweise entspricht der Nutzung der Link-Informationen bei Suchmaschinen. Hier war es die Arbeit von Brin & Page (1998), welche solche Informationen in den Ranking-Prozess einer Suchmaschine integrierten. Dieses Verfahren hat maßgeblich zur Qualität der Suchergebnisse und damit zu dem Erfolg der Suchmaschine „Google“ beigetragen.

1.3 Ziel der Arbeit

Ziel der vorliegenden Arbeit ist es, den von Utard (2005) vorgestellten Ansatz zur Klassifikation von Hypertext-Dokumenten mit anderen Ansätzen aus der Literatur zu vergleichen. Zur Erreichung dieses Ziels wurden die nachstehenden Teilziele verfolgt:

1.3.1 Aufbau von Datenbanken für eine Evaluierung

Der Aufbau von Sammlungen von Hypertext-Dokumenten für einen systematischen Vergleich der Ansätze, da in den vorhergehenden Arbeiten jeweils unterschiedliche Datenbanken genutzt wurden, die einen direkten Vergleich erschweren.

1.3.2 Vergleiche der Methoden

Ein systematischer Vergleich und eine Bewertung der Methoden hinsichtlich ihrer Effektivität (Vorhersagegenauigkeit). Hierbei wurde besonderer Wert darauf gelegt, dass die erhaltenen Ergebnisse *direkt vergleichbar* sind. Zur Erreichung dieses Teilzieles werden alle Experimente unter den gleichen Bedingungen und mit den gleichen Datenbanken durchgeführt.

1.4 Gliederung der Arbeit

Eine schematische Übersicht der Gliederung dieser Arbeit wird in Abbildung 1.3 dargestellt:

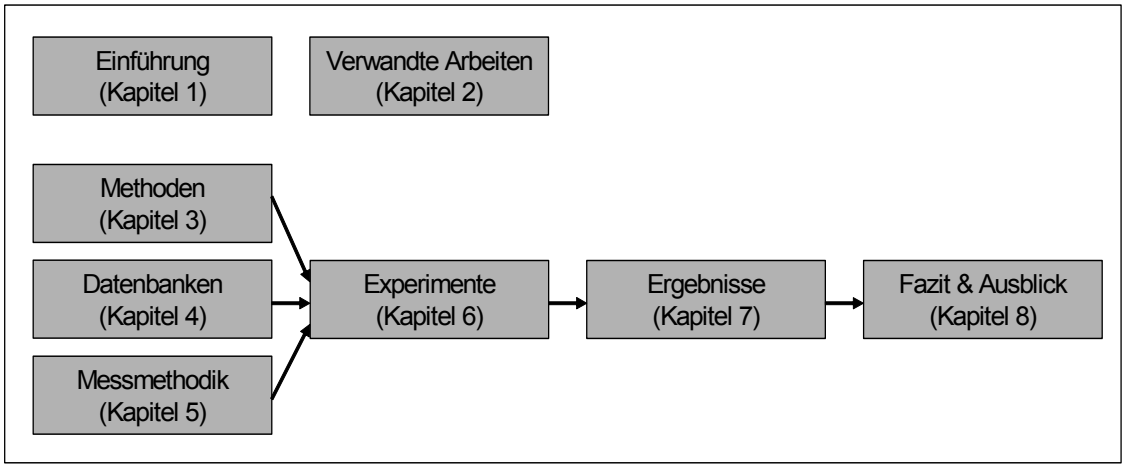

Abbildung 1.3: Die Gliederung der Arbeit

In dem folgenden Kapitel 2 werden *verwandte Arbeiten* aus den Bereichen der Datenbanken sowie Arbeiten, welche sich mit dem Vergleich von Methoden befassen, vorgestellt. Hierbei wird auf die besonderen Anforderungen von Hypertexten eingegangen.

Im Kapitel 3 werden die Eigenschaften von Hypertexten sowie die zu vergleichenden *Methoden der Klassifikation* von Hypertext-Dokumenten anhand einer Gruppierung vorgestellt.

Kapitel 4 beschäftigt sich mit der Auswahl, der Bearbeitung und der Struktur der verwendeten *Datenbanken* für eine Evaluierung. Es liefert auch Aussagen über die Link-Strukturen der verwendeten Daten.

Kapitel 5 gibt eine Einführung in die *Messmethodik* und zeigt eine Übersicht derjenigen Vorhersagegenauigkeits-Werte, die für einen Vergleich der Methoden ausgewählt wurden.

Nachdem die Vorbedingungen eines Vergleiches (Methoden, Datenbanken und Messmethodik) erfüllt wurden, werden in Kapitel 6 der verwendete Klassifikator sowie der Aufbau und die Durchführung der vorgenommenen *Experimente* beschrieben.

Im Kapitel 7 werden die Ergebnisse der Messungen betrachtet und unter verschiedenen Gesichtspunkten (Vorhersagegenauigkeit & Coverage) ein *Vergleich der Methoden* durchgeführt. Als Abschluss wird aus den zuvor erhaltenen Ergebnissen ein neuer Klassifizierer vorgestellt.

Kapitel 8 liefert abschließend eine *Zusammenfassung* der vorhergehenden Ergebnisse und einen *Ausblick* auf mögliche weiterführende Arbeiten.

In den Anhängen befinden sich die detaillierten Messergebnisse der durchgeführten Experimente, eine Beschreibung der implementierten JAVA-Klassen sowie eine Beschreibung der archivierten Testdaten.

Die in den Experimenten verwendeten *Datenbanken* (Sammlungen von Testdaten) befinden sich, in komprimierter Form, auf der beiliegenden DVD.

2 Verwandte Arbeiten

Dieses Kapitel liefert eine Übersicht über verwandte Arbeiten, die sich mit der Erstellung von Datensammlungen für eine Hypertext-Klassifikation sowie mit einem Vergleich von Methoden zur Hypertext-Klassifikation beschäftigen.

2.1 Datensammlungen

Übersichten von Datenbanken aus dem Bereich des Text Mining, welche sich für einen Vergleich eignen, finden sich z. B. bei Chakrabarti (2002), McCallum (2003) und Kan (2005). Hierbei ist aber zu beachten, dass diese Sammlungen unter dem Aspekt einer reinen *Text-Klassifikation* zusammengestellt wurden. Anforderungen der Hypertext-Klassifikation wurden dabei selten berücksichtigt, wie z. B. die Nachbarschaftsseiten (vgl. Abschnitt 3.1:Eigenschaften von Hypertexten).

Eine Datensammlung, die zum Vergleich von Methoden der Text-Klassifikation herangezogen wird, ist z. B. WebKB (Craven et al., 2000), eine Sammlung von klassifizierten Webseiten amerikanischer Universitäten, auf deren Datenbasis Vergleiche von Methoden der Text-Klassifikation durchgeführt wurden (Chakrabarti & Soundalgekar, 2003). Durch die Einbeziehung ihrer Hypertext-Links kann diese Datenbank auch für eine Verifikation von Methoden der *Hypertext-Klassifikation* genutzt werden, wie sie z. B. von Fürnkranz (2002), Lu & Getoor (2003) und Utard (2005) durchgeführt wurde.

2.2 Vergleich von Methoden

In einer Arbeit von Chakrabarti (2000) werden verschiedene Ansätze zur Extraktion von Informationen aus Hypertext-Seiten (Web Mining) und deren spezifische Methoden (wie sie z. B. für eine Klassifikation verwendet werden) verglichen.

Beim *supervised learning* – in diesem Fall liegen alle Informationen z. B. über eine Klassifizierung vor – werden statistische Verfahren verwendet, die anhand von Modellen (welche aus Trainingsdaten gewonnen werden) vorhersagen, welche Klasse bei einer Klassifikation die wahrscheinlichste ist. Ein Verfahren hierfür wird in Abschnitt 6.1:Der Naive-Bayes-Klassifikator vorgestellt.

Beim *semi supervised learning* sind hingegen nur Teile der (Klassen-) Informationen bekannt. Hier werden Techniken verwendet, welche diese Informationen in iterativen Verfahren erweitern (*relaxation labeling*). Dieser Ansatz wird in Abschnitt 3.3.2.4:Berechnung von Klassen der Nachbarschaftsdokumente behandelt.

Beim *unsupervised learning* liegen keine Informationen vor, wie z. B. beim Clustering. Als Clustering bezeichnet man die Zusammenfassung von Dokumenten in Gruppen, so dass die Dokumente innerhalb einer Gruppe eine höhere Ähnlichkeit aufweisen als Ähnlichkeiten zwischen verschiedenen Gruppen vorliegen (Chakrabarti, 2002., p. 8). Hierbei wird durch die Anwendung von Techniken der linearen Algebra eine „Ähnlichkeit“ von (Hypertext-) Dokumenten berechnet.

Nach Borges & Levene (1999) können die Forschungsrichtungen des Web-Mining in drei verschiedene Klassen unterteilt werden: das Mining nach *Inhalten* in Hypertext-Dokumenten, das Mining der *Link-Strukturen* (in Form von Hyperlinks) sowie das Mining des *Benutzerverhaltens*. Hiefür werden die Begriffe *Web Content Mining*, *Web Structure Mining* und *Web Usage Mining* verwendet (Madria et al., 1999).

Die Hypertext-Klassifikation kann hauptsächlich in die Gruppe des Web Structure Mining eingeordnet werden, d. h. es werden Informationen aus den Hyperlinkstrukturen der Webseiten verwendet, welche für eine Klassifikation hilfreich sind. Bei diesen Informationen handelt es sich aber nicht um strukturierte Daten, wie sie z. B. in einer Datenbank vorkommen. Auf der anderen Seite sind die Daten einer Hypertext-Seite nicht vollkommen unstrukturiert, da diese noch die HTML-Tags (Steuerinformationen für die Darstellung in einem Browser) enthalten. Man spricht in diesem Fall von *semi-strukturierten Daten*.

Obwohl die Erforschung von Techniken zur Hypertext-Klassifikation innerhalb der letzten Jahre eine Vielzahl von Methoden hervorgebracht hat, sind die vorgestellten Vorgehensweisen bisher nicht direkt verglichen worden, da jeder Autor seine Evaluierung mit unterschiedlichen Datenquellen durchführt, verschiedene Vorhersagegenauigkeits-Werte berichtet und die bisherigen Arbeiten noch nicht mit einem einheitlichen Framework verglichen wurden.

Es existieren leider nur wenige Vergleiche dieser Methoden, welche einen systematischen Überblick bieten oder über eine (chronologische) Auflistung hinausgehen.

Gerade die Tatsache, dass bisher nur wenige Arbeiten existieren, die einen systematischen Vergleich von Methoden der Hypertext-Klassifikation liefern, zeigt die Wichtigkeit der Ziele dieser Arbeit.

3 Hypertext-Klassifikation

Um die in dieser Arbeit verwendeten Methoden zur Hypertext-Klassifikation verstehen zu können, liefert das Kapitel 3 eine Beschreibung von Hypertexten sowie eine Gruppierung der Methoden. Im Hauptteil dieses Kapitels werden die einzelnen Methoden anhand dieser Gruppierung beschrieben.

3.1 Eigenschaften von Hypertexten

Hypertexte bestehen aus Textseiten, welche untereinander durch Verweise (Hyperlinks) verbunden werden (Abbildung 3.1). Technische Grundlage von Hyperlinks im Internet bildet die standardisierte Auszeichnungssprache HTML (Conklin, 1987).

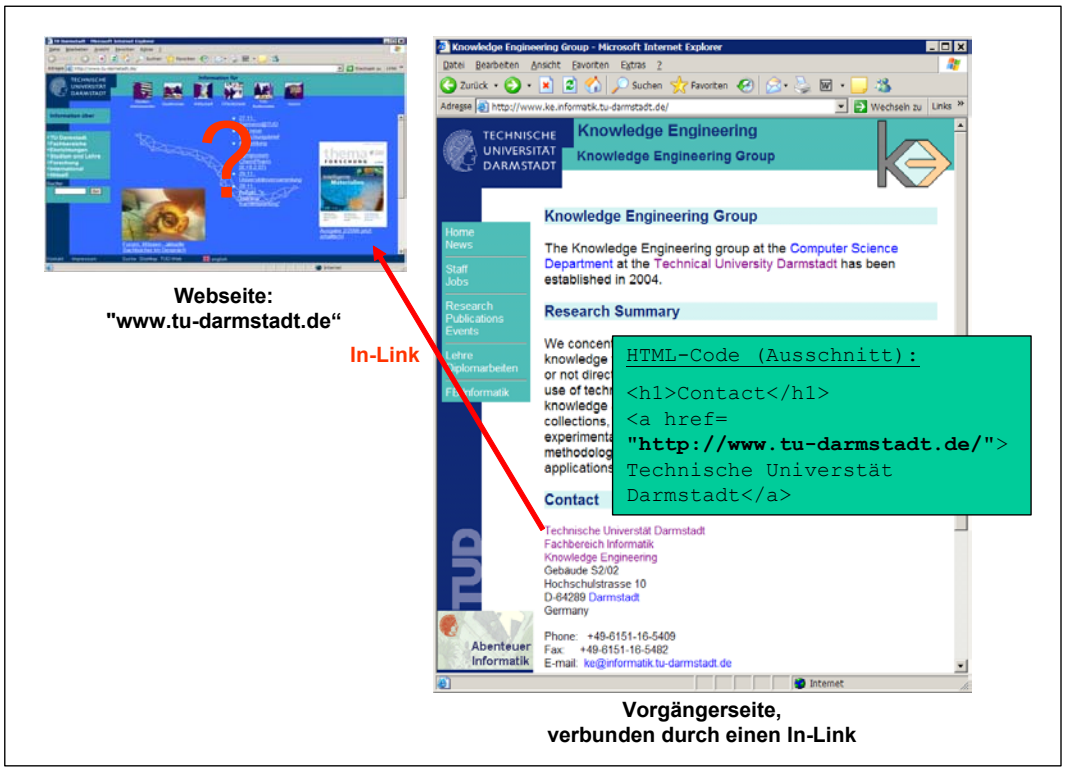

Abbildung 3.1: Verweis (In-Link) zwischen Hypertext-Seiten

Eine Seite kann eine Menge von Vorgängerseiten (verbunden durch In-Links) und eine Menge von Nachfolgerseiten (verbunden durch Out-Links) besitzen (Abbildung 3.2, links). Falls zwei Seiten eine gemeinsame Vorgängerseite besitzen, spricht man von einer Ko-Zitierung (verbunden durch Co-cited-Links).

Eine Ko-Zitierung bildet eine „Brücke“ zwischen zwei Seiten. Aufgrund der Verbindung der Seiten durch einen In-Link und einen Out-Link nennt man diese Form der Verlinkung auch eine *IO-Bridge* (Abbildung 3.2, rechts).

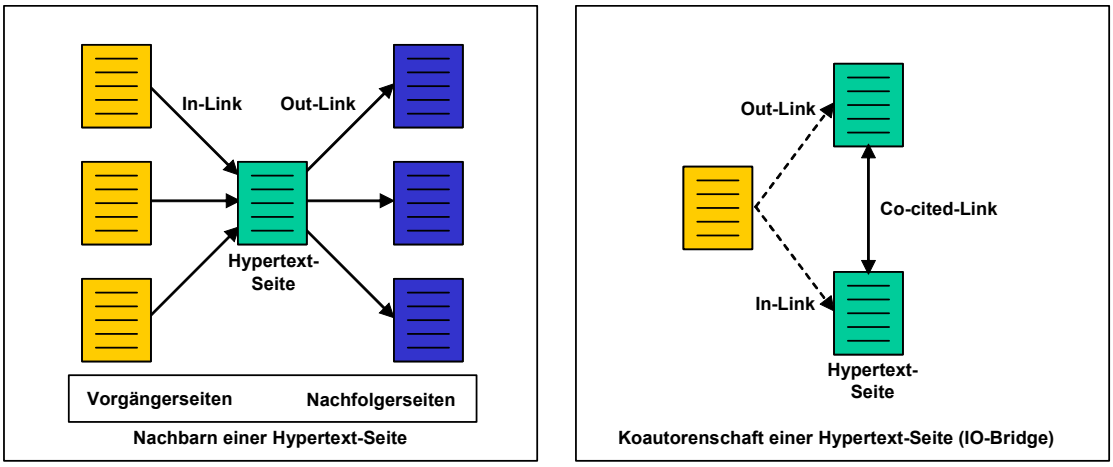

Abbildung 3.2: Nachbarschaften (links) und Ko-Zitierungen (IO-Bridges) von Hypertexten (rechts)

Formal lassen sich die Verweise von Hypertexten durch einen gerichteten Graphen $G=(O,L)$ beschreiben (Lu & Getoor, 2003). Die Menge der n Seiten bildet hierbei die Menge der Knoten $O = \{x_1,\ldots,x_n\}$, die Menge der Links $l_{i\to j}$ zwischen Seiten bildet die Menge der Kanten L des Graphen G.

Hierbei wird ein Link $l_{i\to j}$ als ein Verweis der Seite i auf die Seite j bezeichnet. Nachbarschaften und Ko-Zitierungen zwischen Seiten lassen sich nun wie folgt beschreiben:

a) Menge I der Vorgängerseiten einer Seite x_i:

$$I(x_i) = \{ x_j \mid l_{j\to i} \in L \} \qquad (3.1)$$

b) Menge O der Nachfolgerseiten einer Seite x_i:

$$O(x_i) = \{ x_j \mid l_{i\to j} \in L \} \qquad (3.2)$$

c) Menge Co der Ko-Zitierungen (IO-Bridges) einer Seite x_i:

$$Co(x_i) = \{ x_j \mid x_j \neq x_i \wedge (\exists\, x_k : x_k \in I(x_i) \wedge x_k \in I(x_j)) \} \qquad (3.3)$$

Die Menge von Vorgänger- bzw. Nachfolgerseiten bezeichnet man gemeinsam als die *Nachbarschaft* einer Seite. Die Nachbarschaften bzw. Ko-Zitierungen bieten eine zusätzliche Informationsquelle, die für eine Klassifikation genutzt werden kann.

3.2 Gruppierung von Methoden zur Hypertext-Klassifikation

Methoden zur Hypertext-Klassifikation lassen sich nach Utard & Fürnkranz (2006), je nach den verwendeten Informationen aus den zu klassifizierenden Dokumenten, in folgende Gruppen einteilen (Abbildung 3.3):

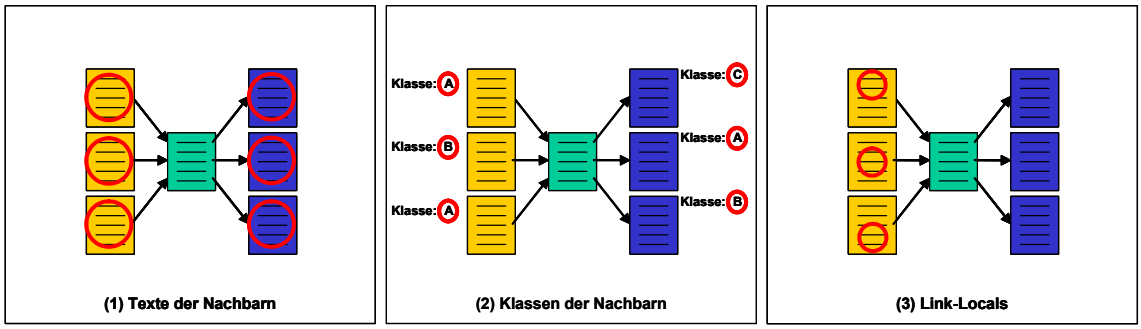

Abbildung 3.3: Gruppierung von Methoden zur Hypertext-Klassifikation

(1) Texte der Nachbarn: Für eine Klassifikation werden die Texte aller Nachbarn verwendet (Chakrabarti et al., 1998). Dieses Vorgehen kann die Menge der verwendeten Informationen für eine Klassifikation (Features) beträchtlich erhöhen.

(2) Klassen der Nachbarn: Hierbei wird versucht, den Inhalt einer Nachbarseite in einem einzigen Attribut (der Klasse) zu abstrahieren. Anstelle des kompletten Textes werden bei dieser Gruppe nur die Klassen der Nachbarn für eine Klassifikation verwendet (Chakrabarti et al., 1998 bzw. Lu & Getoor, 2003).

(3) Link-Locals: Nach Utard & Fürnkranz (2006) besteht bei der Verwendung von (2) der Nachteil, dass zu viele irrelevante Informationen (Features) mit in die Klassifikation eingehen. Bei der Verwendung von (3) gehen jedoch zu viele relevante Informationen verloren. Dies ist die Motivation für die Link-Local-Methoden (Fürnkranz, 2002). Durch die Verwendung von relevanten Teilinformationen der Vorgängerseiten versuchen diese Methoden, einen Kompromiss aus (2) und (3) zu finden, der die Vorhersagegenauigkeit (vgl. Abschnitt 5.1:Evaluierung der Ergebnisse) einer Klassifikation erhöht.

3.3 Methoden zur Hypertext-Klassifikation

In dem folgenden Abschnitt werden die Methoden zur Hypertext-Klassifikation anhand der in Abschnitt 3.2 erfolgten Gruppierung vorgestellt:

3.3.1 Texte der Nachbarn

Chakrabarti et al. haben in ihrer Arbeit von 1998 untersucht, ob die Verwendung von Texten aller Nachbarseiten eine effektive Methode zur Klassifikation von Hypertexten darstellt. Die Experimente wurden mit den Daten durchgeführt, die aus der IBM-Patent-Sammlung[5] gewonnen wurden. Die 11.160 ausgewählten Dokumente dieser Datenbank

[5] http://www.ibm.com/patents/

enthalten im Schnitt 5-10 Nachbarschaftsseiten und sind über 12 Klassen gleichmäßig verteilt.

Für eine Klassifikation der Seite x_i wurden die Wörter der Seite x_i sowie alle Wörter der Vorgängerseiten $I(x_i)$ bzw. der Nachfolgerseiten $O(x_i)$ als Features verwendet. Um die Ergebnisse zu evaluieren, wurden die gewonnenen Messwerte mit denen einer Text-Klassifikation verglichen. Als Ergebnis wurde festgestellt, dass die Einbeziehung der Nachbarschaft einer Seite den Error einer Klassifikation (vgl. Abschnitt 5.1.1:Kennzahlen zur Evaluierung) von 36% (Text Klassifikation) auf 38,3% vergrößert.

Hierbei brachte es auch keinen Vorteil, die Features der Seite x_I von den Features der Nachbarseiten $I(x_i)$ und $O(x_I)$ durch *Tagging* zu trennen. Diese Methode bietet bei der Klassifikation eine Möglichkeit, Feature-Gruppen separat zu behandeln (vgl. Abschnitt 6.2.1.2:Tagging). Eine Verwendung des Tagging erbrachte bei diesem Experiment einen Error-Wert von 38,2%.

Chakrabarti et al. erklären diese Verschlechterung mit der hohen Anzahl der Nachbarschaftsseiten und der Tatsache, dass die Seiten der verwendeten Datenbank über eine hohe Anzahl von Querverweisen verfügen, die eine Klassifikation beeinflussen.

3.3.2 Klassen der Nachbarn

Im vorherigen Abschnitt wurden die Texte der Nachbarschaften einer Seite für eine Klassifikation verwendet. Bei der Verwendung dieser Methode besteht das Problem, dass dadurch eine hohe Anzahl von (irrelevanten) Informationen, den Features, verwendet wird.

Eine Lösung dieses Problems könnte darin bestehen, sich bei einer Klassifikation alleine auf die Klassen der Nachbarschaftsseiten zu beschränken. Um eine Motivation für dieses Vorgehen zu liefern, wurden die Nachbarschaftsstrukturen von Hypertext-Seiten in dem folgenden Experiment untersucht.

3.3.2.1 Experiment: Nachbarschaftsstrukturen

Für eine Untersuchung der Nachbarschaftsstrukturen wurden 5.257 Hypertext-Dokumente analysiert, die fünf verschiedenen, nahezu gleich verteilten Klassen angehören. Die Auswahl der Dokumente erfolgte aus dem Webverzeichnis „ODP“, einem aus insgesamt 16 Klassen bestehenden „Katalog“ des Internets (vgl. Abschnitt 4.1:Auswahl der Datenquellen).

Durch ihre Einträge in dem Web-Verzeichnis ist die Zugehörigkeit der Hypertext-Dokumente zu einer der fünf Klassen $c \in C$ bekannt: C={„Arts“, „Business“, „Computers“, „Science“, „Sports“}.

Mit einem im Abschnitt 4.1.3:ODP beschriebenen Verfahren wurden 136.492 Dokumente der Nachbarschaft $I(x_i)$ und $O(x_i)$ beschafft (68.928 verschiedene Vorgängerseiten und 67.564 verschiedene Nachfolgerseiten). Hierbei kann eine Seite x_i sowohl in $I(x_i)$ als auch in $O(x_i)$ enthalten sein.

Die Klasse einer Nachbarschaftsseite kann durch einen evtl. vorhandenen Eintrag in dem gleichen Web-Verzeichnis festgestellt werden. Falls die Klasse nicht festgestellt werden kann, wird diese Nachbarschaftsseite nicht verwendet.

Tatsächlich konnten nur 4.703 (ca. 7%) der Nachbarschaftsseiten einer bekannten Klasse zugeordnet werden (2.173 verschiedene Vorgängerseiten und 2.530 verschiedene Nachfolgerseiten), da nur diese Nachbarschaftsseiten im verwendeten Webkatalog aufgeführt wurden. Obwohl diese Zahl niedrig erscheint, reicht die Anzahl aus, um in dem folgenden Experiment verwertbare Ergebnisse zu erhalten:

Die Menge X_c beschreibt die Menge der Hypertext-Dokumente, die der gleichen Klasse c angehören. Für jede Klasse $c \in C$ wurde untersucht, welchen (bekannten) Klassen die Nachbarschaftsseiten von X_c angehören. Hierbei erfolgte eine Differenzierung nach

- Vorgängerseiten $I(X_c)$ in Abbildung 3.4,
- Nachfolgerseiten $O(X_c)$ in Abbildung 3.5,
- Seiten mit Ko-Zitierungen $Co(X_c)$, d. h. Seiten, die mindestens eine gemeinsame Vorgängerseite besitzen in Abbildung 3.6.

Aus Gründen der Übersichtlichkeit wurden Dokumente, die zwar in dem (aus insgesamt 16 Klassen bestehenden) Webkatalog gefunden wurden, sich aber nicht in den fünf gegebenen Klassen C befinden, unter der Klasse: „*Andere“ zusammengefasst.

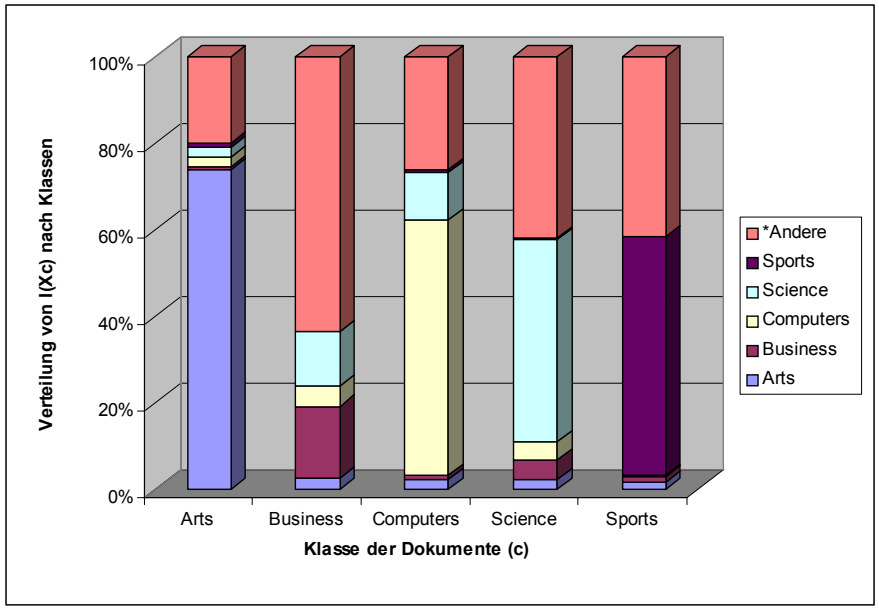

Abbildung 3.4: Verteilung der Vorgängerseiten $I(X_c)$ nach Klassen

Diese Analyse zeigt, dass ein Zusammenhang zwischen der Klasse einer Hypertext-Seite und den Klassen ihrer Vorgängerseiten existiert. Beispielsweise gehören bei Dokumenten der Klasse „Arts“ über 70% der Vorgängerseiten auch der Klasse „Arts“ an.

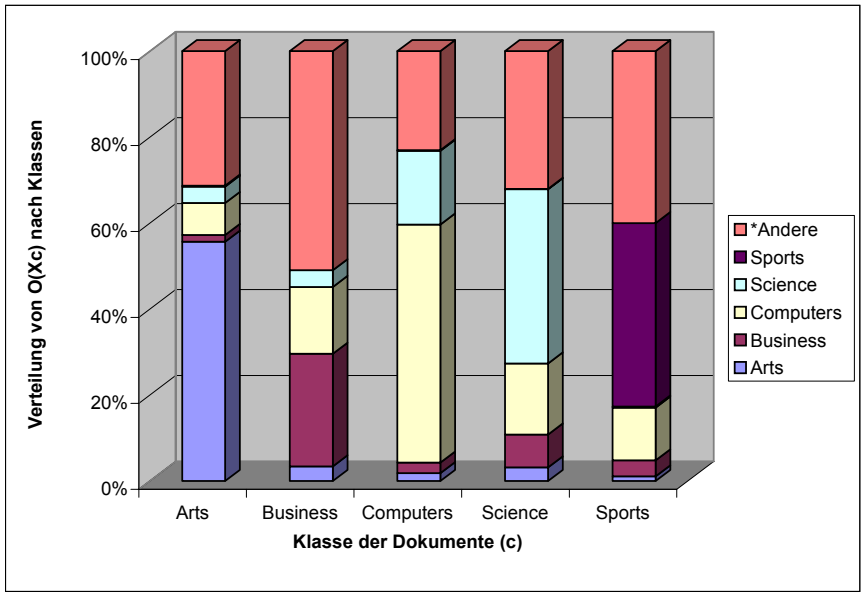

Abbildung 3.5: Verteilung der Nachfolgerseiten $O(X_c)$ nach Klassen

Der Zusammenhang zwischen der Klasse einer Seite und den Klassen ihrer Nachfolgerseiten ist nicht mehr so stark ausgeprägt. So gehören z. B. bei Hypertext-Seiten der Klasse „Arts“ nur noch ca. 50% der Nachfolgerseiten ebenfalls der Klasse „Arts“ an.

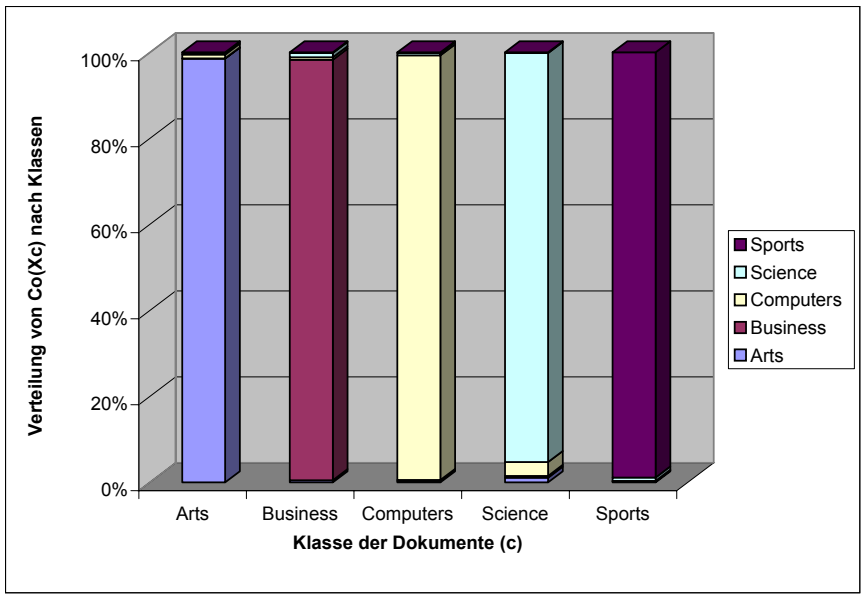

Abbildung 3.6: Verteilung der Ko-Zitierungen $Co(X_c)$ nach Klassen

Besonders auffällig wird der Zusammenhang bei der Analyse von Ko-Zitierungen. Hier befinden sich nahezu alle Seiten, die mindestens eine gemeinsame Vorgängerseite besitzen, innerhalb einer Klasse.

Methoden zur Hypertext-Klassifikation, welche die Klasseninformationen von Nachbarschaften verwenden, versuchen diese Zusammenhänge durch den Einsatz verschiedener Verfahren zu nutzen:

3.3.2.2 Methoden von Chakrabarti et al. (1998)

Bei den Methoden von Chakrabarti et al. wird zwischen Klasseninformationen aus Nachbarschaften und solchen aus Ko-Zitierungen (IO-Bridges bzw. Local IO-Bridges) unterschieden. Aufgrund der Verwendung von einem bzw. zwei unterschiedlichen Links bezeichnet man diese Methoden als Radius-1- bzw. Radius-2-Methoden (vgl. Abbildung 3.3).

I) Nachbarschaften (Radius-1)

Bei einer Text-Klassifikation werden die Wörter der Seite x_i als Features für eine Klassifikation verwendet. Bei einer Klassifikation, die auf den Klassen der Nachbarschaftsdokumente basiert, werden anstelle der Wörter die *Klassen der Nachbarschaftsseiten* $I(x_i)$ und $O(x_i)$ als Features genutzt.

Der Vorteil dieser Methode liegt in der wesentlich geringeren Anzahl der Features (=Anzahl der Klassen) gegenüber denen einer Text-Klassifikation. Eine derartige Reduzierung der Feature-Menge kann sich jedoch negativ auf eine Klassifikation auswirken, da (möglicherweise relevante) Informationen der Seiten verloren gehen.

Chakrabarti et al. kombinierten die Features einer Text-Klassifikation (Wörter) mit den Features, die aus den Informationen der Nachbarschaften (Klassen) gewonnen wurden (vgl. Abschnitt 6.2.2:Verwendung mehrerer Klassifikatoren).

Eine Überprüfung der Radius-1-Methoden erfolgte mit den schon in Abschnitt 3.3.1:Texte der Nachbarn verwendeten Daten. Bei den Experimenten wurde angenommen, dass die Klassen aller Nachbarschaftsdokumente einer zu klassifizierenden Seite bekannt sind (vgl. Abschnitt 3.3.2.4:Berechnung von Klassen der Nachbarschaftsdokumente). Die Autoren sprechen in diesem Fall von einem *supervised case*.

Die alleinige Verwendung der Klassen ergab einen Error-Wert von 34%, die Kombination von Klassen und eigenem Text einen Error-Wert von 21%. Diese Fehlerraten liegen unter dem Error-Wert von 36%, der bei der alleinigen Verwendung des eigenen Textes erreicht wurde.

II) Ko-Zitierungen (Radius-2)

a) IO Bridges

Bei dieser Methode werden die Klassen Ko-Zitierungen $Co(x_i)$ als Features für die Klassifikation der Seite x_i verwendet. Bei dem in Abschnitt 3.3.2.1 durchgeführten Experiment wurde gezeigt, dass die Klassen der Ko-Zitierungen eine hohe Korrelation zu den Klassen der zu klassifizierenden Dokumente aufweisen (vgl. Abbildung 3.6). Auch bei dieser Methode entspricht die Anzahl der Features der Anzahl der Klassen.

Ein Nachteil beim Verwenden von IO Bridges ist, dass nicht alle zu klassifizierenden Seiten eine gemeinsame Vorgängerseite besitzen. Als Folge davon können einige Seiten nicht klassifiziert werden. Dies wirkt sich negativ auf die *Coverage* einer Methode aus, die den Anteil der Seiten beschreibt, die durch eine Methode klassifiziert werden können.

b) Local IO Bridges

Webseiten können Listen von Verweisen (Out-Links) auf Nachfolgerseiten besitzen, die thematisch zusammenhängen. So könnte z. B. auf der Webseite einer Online-Enzyklopädie eine zusammenhängende Liste von Verweisen (Out-Links) zu einem bestimmten Thema existieren (Abbildung 3.7).

Solche Informationen wären für eine Klassifikation hilfreich, da wahrscheinlich alle diese Verweise einer gemeinsamen Klasse zugeordnet werden könnten. Diesen Ansatz verfolgen Local IO Bridges.

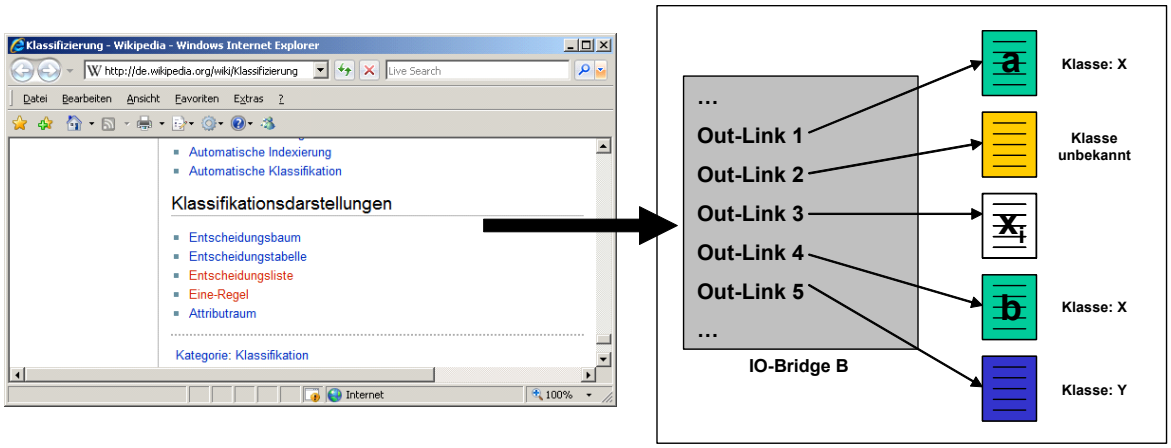

Abbildung 3.7: Local IO-Bridges

Bei Local IO Bridges wird eine Klasse c als Feature für eine Klassifikation der Seite x_i verwendet, falls alle folgenden Bedingungen zutreffen:

1. Es existiert eine IO Bridge B, welche auf x_i als Nachfolgerseite (durch einen Out-Link) verweist.
2. In dieser IO Bridge B existieren drei Verweise (Out-Links) auf die Nachfolgerseiten: a, x_i, b in dieser Reihenfolge.
3. Die Klassen der Dokumente a und b sind bekannt und gleich.
4. Es existieren keine Verweise (Out-Links) *zwischen* den Verweisen auf a und b, welche auf Dokumente mit bekannten Klassen verweisen.

Die Anzahl der Features entspricht bei dieser Methode wiederum der Anzahl der existierenden Klassen. Bei der Verwendung von Local IO Bridges besteht, wie im vorherigen Abschnitt, das Problem der geringen Coverage dieser Methode.

Chakrabarti et al. überprüften Radius-2-Methoden anhand einer Datensammlung, die aus dem Webverzeichnis „Yahoo!“ entnommen wurde. Diese Sammlung umfasste 849 Dokumente aus 13 verschiedenen Klassen.

Bei der Überprüfung stellten sie fest, dass die Verwendung von IO Bridges einen Error-Wert von 25%, die Verwendung von Local IO Bridges einen Error-Wert von 21% erbrachte. Der Vergleichswert dieser Datenbank (Text-Klassifikation) liegt bei einem Error-Wert von 68%.

Allerdings mussten die geringeren Error-Werte der Methoden auch durch eine geringere Coverage (der Anteil der Seiten, welche klassifiziert werden können) „erkauft“ werden. Während die Text-Klassifikation eine Coverage von 100% erreicht, liegt dieser Wert bei der Verwendung von IO Bridges nur bei 75% und Local IO Bridges erreichen lediglich eine Coverage von 72%.

3.3.2.3 Methoden von Lu & Getoor (2003)

Während die Methoden von Chakrabarti et al. Nachbarschaften und Ko-Zitierungen getrennt behandeln, haben Lu & Getoor diese in gemeinsamen Modellen vereinigt. Hierbei werden für die einzelnen Link-Gruppen (In-Links, Out-Links und Co-Links) *Link-Feature-Modelle* erstellt, die Statistiken über die Nachbarschaften bzw. Ko-Zitierungen enthalten (Abbildung 3.8):

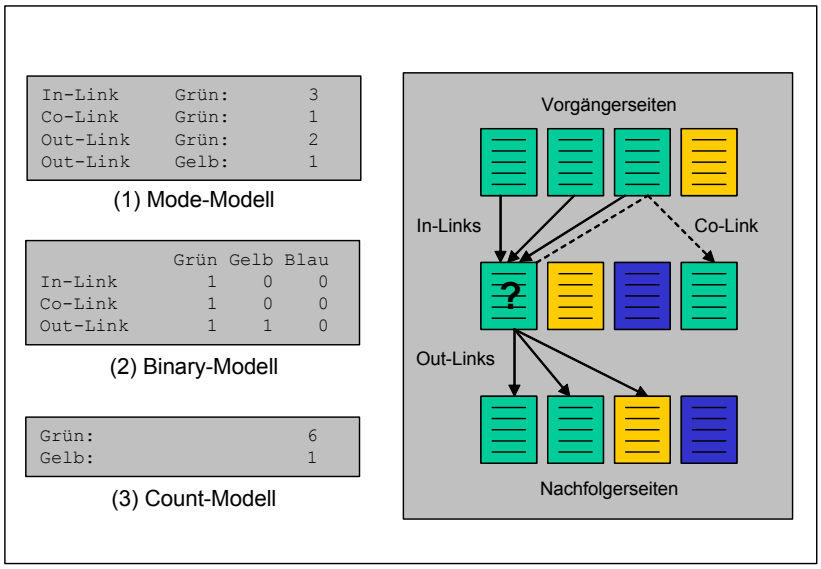

Abbildung 3.8: Link-Feature-Modelle

Bei Link-Feature-Modellen werden Nachbarschaften durch die Link-Typen In-Link bzw. Out-Link abgebildet, Ko-Zitierungen hingegen durch den Link-Typ Co-Link. Diese Modelle sind nach absteigendem Informationsgehalt über die enthaltenen Link-Strukturen sortiert:

(1) Mode-Modell: Dieses stellt die ausführlichsten Statistiken für die Nachbarschaften (durch In-/ bzw. Out-Links) und die Ko-Zitierungen (Co-Links) einer Seite bereit. In diesem Modell werden die Anzahl und die Klasse (wie z. B. „Grün“) von Links, geordnet nach den Link-Typen In-Links, Out-Links und Co-Links abgebildet.

(2) Binary-Modell: Bei diesem Modell werden die Klassen der Link-Typen in einem Binärvektor aufgezeigt. Das n-te Element dieses Binärvektors beschreibt, ob bei diesem Linktyp mindestens ein Link auf die Klasse n vorhanden ist. Bei dem Binary-Modell gehen Detailinformationen über die Anzahl von Links verloren.

(3) Count-Modell: Hierbei wird nur noch die Anzahl aller Link-Typen einer Klasse abgebildet. Dies stellt die einfachste Art der Informationen über Links dar.

Um mit Hilfe dieser Statistiken eine Klassifikation einer Seite x_i durchzuführen, wird das folgende Verfahren angewendet:

Die durch die Modelle (1)-(3) gewonnenen Statistiken über die Link-Strukturen einer Seite x_i liefern die *Link Features*. Eine zweite Feature-Gruppe besteht aus dem Text, welcher in x_i enthalten ist (*Object Features*). Aus diesen beiden Feature-Gruppen können folgende unterschiedliche Kombinationen gebildet werden:

> **Flat-Mode**: Beide Feature-Gruppen (Object Features & Link Features) werden in einem gemeinsamen Meta-Dokument zusammengefasst (vgl. Abschnitt 6.2.1.1:Merging).
>
> **Link-Mode**: Für jede Feature-Gruppe wird eine eigene Klassifikation durchgeführt; die Ergebnisse beider Klassifikationen (Wahrscheinlichkeiten) bilden die Grundlage für die Vorhersage einer Seite (vgl. Abschnitt 6.2.2:Verwendung mehrerer Klassifikatoren).

Le & Getoor überprüften diese Methoden anhand der folgenden Datensammlungen:

- einer Auswahl von 3.181 Dokumenten aus dem Cora-Datenset (einer Sammlung von Arbeiten aus dem Bereich des maschinellen Lernens);
- einer Auswahl von 3.600 Dokumenten aus dem Cite-Seer-Datenset, welche 7.522 Zitate auf diese Dokumente enthält, und
- einer Auswahl von 700 Dokumenten aus der WebKB-Datenset, einer Sammlung von Webseiten amerikanischer Universitäten.

Die Ergebnisse dieser Untersuchung sind sehr vielfältig, da alle Link-Feature-Modelle im Flat-Mode bzw. im Link-Mode getestet wurden. Prinzipiell stellte sich heraus, dass der Flat-Mode hierbei eine höhere Vorhersagegenauigkeit aufweist.

Bei einem Vergleich der F_1-Werte (vgl. Abschnitt 5.1.1:Kennzahlen zur Evaluierung) zwischen dem Count-Modell im Link-Mode und einer Text-Klassifikation wurden die folgenden Ergebnisse festgestellt:

- Bei dem Cora-Datenset wurde, gegenüber einem Wert von 0,643 bei einer Text-Klassifikation, ein Wert von 0,741 erreicht,
- bei dem Cite-Seer-Datenset ein Wert von 0,606 (gegenüber 0,551) und
- bei dem WebKB-Datenset ein Wert von 0,858 (gegenüber 0,832).

3.3.2.4 Berechnung von Klassen der Nachbarschaftsdokumente

Bei der Klassifikation einer Seite x_i durch die Klassen der Nachbarschaften $I(x_i)$ und $O(x_i)$ wird vorausgesetzt, dass die Klassen der Nachbarschaftsseiten a priori bekannt sind. Chakrabarti et al. nennen dies den *supervised case*. In der Realität, d. h. bei der

Verwendung von Seiten, die z. B. aus dem Internet entnommen wurden, ist dies nicht immer der Fall, da solche Daten eine hohe Anzahl von Nachbarschaftsseiten besitzen können (vgl. Abschnitt 3.3.2.1:Experiment: Nachbarschaftsstrukturen).

Hier können nur Teile der Klasseninformationen (*partially supervised case*) oder gar keine Informationen über die Klassen der Nachbarschaftsseiten vorliegen (*unsupervised case*). Um dennoch eine Klassifikation durchführen zu können, wird ein *Relaxation-labeling-Verfahren* genutzt, bei dem versucht wird, die Klassen der Nachbarschaftsseiten iterativ zu berechnen, die auf dem Text einer Seite basieren.

Abbildung 3.9 liefert eine schematische Darstellung dieses Verfahrens am Beispiel der Radius-1-Methode von Chakrabarti et al.

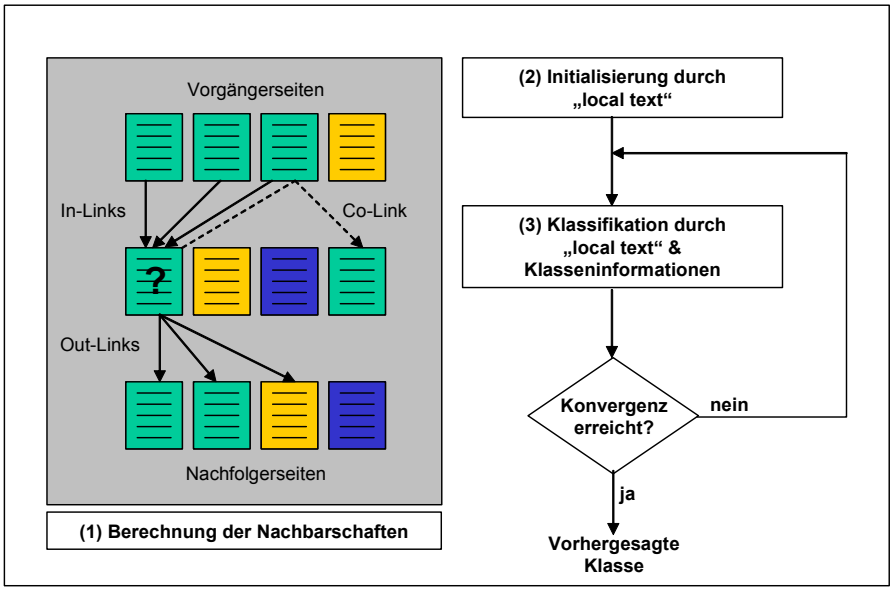

Abbildung 3.9: Iteratives Verfahren zur Berechnung der Klassen

Gegeben ist die zu klassifizierende Seite x_i,

(1) Für x_i werden die Mengen der Nachbarschaftsseiten $I(x_i)$ und $O(x_i)$ ermittelt.

(2) Initialisierung: Alle Nachbarschaftsseiten der Seite x_i werden mit einem Text-Klassifikator vorklassifiziert, welcher den *local text* verwendet.

Der folgende Schritt wird so oft wiederholt, bis eine Konvergenz eintritt, d. h. bis sich z. B. die Klassenverteilungen der Nachbarschaftsseiten von x_i stabilisiert haben:

(3) Iteration: Alle Nachbarschaftsseiten der Seite x_i werden mit einer Kombination aus dem *local text* und den Klasseninformationen neu berechnet (vergl. Abschnitt 6.2.2:Verwendung mehrerer Klassifikatoren).

Lee & Getoor verwendeten das gleiche Verfahren, wobei vor dem Schritt (3) noch eine Neuberechnung der Link-Statistiken (Link-Feature-Modelle) erfolgt (Abbildung 3.8: Link-Feature-Modelle).

Chakrabarti et al. stellten fest, dass die Error-Rate von 21%, die ihr Verfahren im supervised case erreichte, auf ca. 24% im partially supervised case bzw. auf 26% im unsupervised case anstieg. Schon nach wenigen Iterationen wird ein brauchbares Ergebnis erzielt ; weitere Iterationen liefern dann nur noch Veränderungen der Wahscheinlichkeiten im Bereich von 10^{-30}. Für eine vollständige Berechnung der Wahrscheinlichkeiten (k Klassen von n Nachfolgerseiten) bedarf es jedoch n^k Rechenschritte.

3.3.3 Link-Local-Methoden

Die bisher vorgestellten Methoden verwenden als Features für eine Klassifikation die Wörter der Nachbarschaftsseiten (wobei zu viele irrelevante Wörter in eine Klassifikation eingehen können) bzw. die Klassen der Nachbarschaftsseiten (dabei gehen evtl. relevante Informationen verloren).

Im Gegensatz dazu verwenden die Link-Local-Methoden nur die relevanten Wörter der Vorgängerseiten als Features. Als relevant werden hierbei die Wörter betrachtet, welche in einem bestimmten lokalen Kontext des In-Links eines Dokumentes auftreten können (Link Local). Dies schließt die Verwendung von Out-Links für eine Klassifikation mit Link-Local-Methoden aus, da ein Out-Link immer auf eine komplette Seite verweist; eine Ausnahme wäre die Verwendung des Titels einer Nachfolgerseite.

Diese Features bilden, wie bei einer Text-Klassifikation, die Grundlage für die Klassifikation einer Seite. Ein hierfür geeignetes Feature ist z. B. die *Link-Description*, d. h. die textuelle Beschreibung des Links, wie diese im Webbrowser dargestellt wird (Abbildung 3.10):

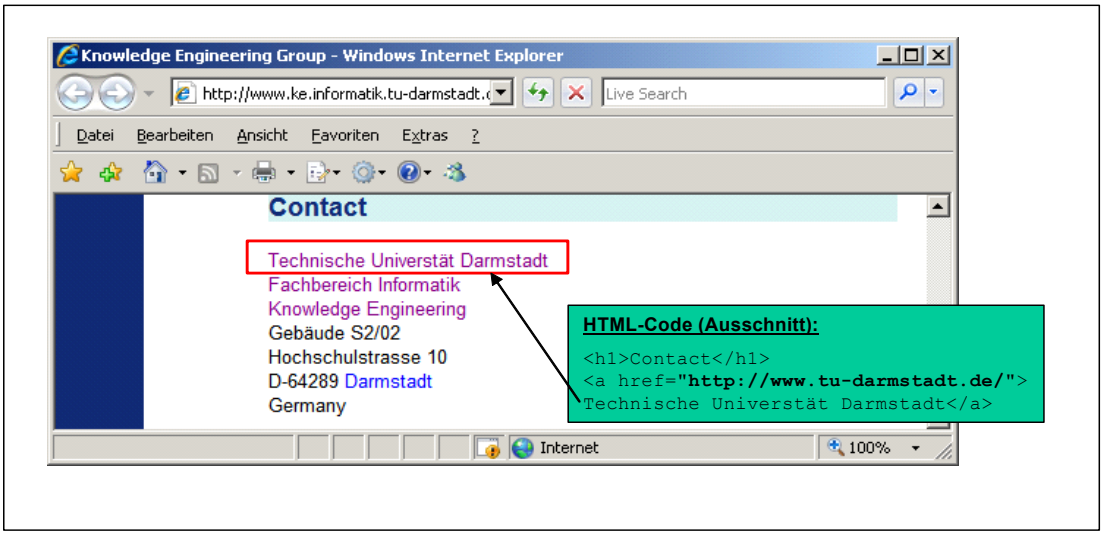

Abbildung 3.10: Link Local Feature: „Link Description“

In der Arbeit von Utard (2005), die auf den Ergebnissen Fürnkranz’ (2002) basiert, wurden die folgenden Link-Local-Methoden untersucht, wobei die Extraktion der

Features durch X-Path-Methoden erfolgte (vgl. Abschnitt 6.3.2:Mining der Links durch XPath):

Own Text: Der Text einer Seite ohne seine HTML-Steueranweisungen (HTML-Tags). Wie auch in den vorangegangenen Methoden dient diese Methode einem Vergleich gegenüber der Text-Klassifikation.

Link Description: Die Beschreibung, d. h. die textuelle Darstellung, eines Links in einem Webbrowser (Abbildung 3.10).

Link Paragraph: Wörter des Abschnitts (Paragraph), in welchem der Link enthalten ist. Technisch betrachtet ist dies der Bereich, der durch die HTML-Tags und im HTML-Code einer Seite begrenzt wird.

Link Neighborhood (Words Around): Eine fest definierte Anzahl von Wörtern, welche vor bzw. nach einem Link verwendet wird, wobei die Link-Descriptions nicht berücksichtigt werden.

Link Headings: Die Überschrift desjenigen Abschnitts, in dem ein Link enthalten ist. Technisch wird hierbei ein Abschnitt durch die HTML-Tags <h1> bis <h3> im HTML-Code einer Seite eingeleitet.

Link List Headings: Falls ein Link innerhalb einer Liste verwendet wird, liefert diese Methode die Überschrift der Liste. Technisch wird eine Liste durch die Verwendung der HTML-Tags <ul> bzw. im HTML-Code der Seite definiert.

Eine Überprüfung der Link-Local-Methoden erfolgte von Utard in den folgenden Datenbanken:

- WebKB, eine bereits von Lu & Getoor verwendete Datensammlung (vgl. Abschnitt 3.3.2.3:Methoden von Lu & Getoor (2003). In dieser Datenbank wurde eine 5-fold Cross-Validation verwendet (vgl. Abschnitt 5.2:Cross-Validation.
- Eine Sammlung von 3.998 Dokumenten (mit fünf verschiedenen Klassen), die mit Hilfe des deutschsprachigen Web-Kataloges Allesklar[6] gewonnen wurde.

Als Klassifikator kam bei diesen Untersuchungen eine *Support-Vector-Machine* (SVM-Light) zum Einsatz.

Untersuchungen der Link-Neighborhood-Methoden mit einer wachsenden Anzahl von Wörtern (Features) hatten keine Auswirkungen auf das Ergebnis einer Klassifikation. Dieses steht im Widerspruch zu den Ergebnissen von Chakrabarti et al., die bei der Verwendung vieler (möglicherweise irrelevanter) Features einen Rückgang der Vorhersagegenauigkeit beobachteten (vgl. Abschnitt 3.3.1:Texte der Nachbarn).

[6] http://www.allesklar.de

Dieser Widerspruch wird jedoch durch den Umstand relativiert, dass Chakrabarti et al. bei ihren Untersuchungen die Texte aller Nachbarschaftsseiten $I(x_i)$ und $O(x_i)$ incl. des eigenen Textes x_i einbezogen, Utard hingegen nur die Texte der Vorgängerseiten $I(x_i)$ untersuchte.

Eine Untersuchung der einzelnen Link-Local-Methoden zeigte, dass auf der Web-KB-Datensammlung keine Steigerung der F_1-Werte durch die Anwendung dieser Methoden erreicht wurde. Während eine Text-Klassifikation bei dieser Datenbank einen F_1-Wert von 0,3199 erbrachte, lieferte der Einsatz der Link-Local-Methode (*Link Description*) einen F_1-Wert von 0,2668.

In der Allesklar-Datensammlung jedoch wurde, bei einem Vergleichswert von 0,4441 (Text-Klassifikation), bei *Link Neighborhood* ein Wert von 0,2497, bei *Link Description* sogar ein Wert von 0,2668 erreicht.

Eine systematische Kombination zweier verschiedener Features, bei der die Features duch Merging zusammengeführt wurden (vgl. Abschnitt 6.2.1.1:Merging), zeigte, dass der F_1-Wert nochmals gesteigert werden kann. Durch die Kombination von *Link List Heading* und *Link Description* wurde ein Wert von 0,7508 in der Allesklar-Datensammlung gemessen.

Wie in diesem Kapitel mehrfach aufgezeigt bedarf es für eine Evaluierung der Methoden geeigneter Testdaten. Um die unterschiedlichen Ansätze der Hypertext-Klassifikation vergleichen zu können, werden Datenbanken benötigt, die einen solchen Vergleich auch ermöglichen. Im nächsten Kapitel erfolgt eine Beschreibung der Datenquellen, die Aufarbeitung und Zusammenstellung der Daten sowie eine Betrachtung der Daten hinsichtlich ihrer Nachbarschaftsstrukturen.

4 Datenbanken zur Evaluierung

Dieses Kapitel beschreibt die Motivationen für die Auswahl der Datenquellen, welche für eine Evaluierung der Methoden verwendet wurden. Nach der Beschaffung und der Aufbereitung in ein standardisiertes Format (XHTML) werden die erhaltenen Daten in Datenbanken (Sammlungen von Dokumenten mit einer einheitlichen Struktur) zusammengefasst. Abschließend werden die Strukturen der Nachbarschaften innerhalb der Datenbanken analysiert.

4.1 Auswahl der Datenquellen

Eine Auswahl von Datenquellen erfolgte anhand der Anforderung an die zu erstellenden Testdaten:

- Es sollte ein ausgewogenes Verhältnis zwischen Standard-Datenbanken bzw. aktuellen „real life“-Daten (Daten aktueller Webseiten) bestehen.
- Die Klassenverteilung sowie die Anzahl der vorhandenen Dokumente sollten variieren, um so differenziertere Ergebnisse zu erlangen.
- Die Trennschärfe der Klassen sollte variieren, d. h. es sollte für einen Klassifizierer unterschiedlich schwierig sein, eine Klasse zu bestimmen.
- Da bei den Methoden zur Hypertext-Klassifikation die Verlinkung der Dokumente eine große Rolle spielt, sollten Datenbanken betrachtet werden, die unterschiedlich starke Verlinkungen enthalten.

Leider erfüllen die meisten „klassischen“ Datensammlungen diese Anforderungen nicht, da sie zum Zwecke einer Text-Klassifikation generiert wurden (vgl. Abschnitt 2.1:Datensammlungen). Diese besitzen entweder keine Links auf Nachbarschaftsseiten oder die Links weisen keinen lokalen Kontext auf, da sie meist am Ende eines Dokumentes zusammengefasst werden.

So ist z. B. eine Datenbank, die von Chakrabarti et al. verwendet wurde (IBM-Patentdaten-Sammlung), für eine Evaluierung von Link-Local-Methoden nicht geeignet. Auch die bekannte Cite-Seer-Datensammlung kann aus diesem Grunde nicht genutzt werden.

4.1.1 WebKB

Um dennoch eine Vergleichbarkeit mit den vorherigen Arbeiten herzustellen, wurde die WebKB-Datenbank ausgewählt, mit der sowohl Utard als auch Lu & Geetor Messungen

durchführten. Diese Datensammlung besteht aus Hypertext-Dokumenten von vier amerikanischen Universitäten (Cornell, Texas, Washington und Wisconsin) plus einer fünften Gruppe, in der Hypertext-Dokumente von anderen amerikanischen Universitäten zusammengefasst wurden. Diese Hypertext-Dokumente sind in sieben Klassen vorklassifiziert (vgl. Tabelle 4.1).

Von den insgesamt 8.282 Hypertext-Dokumenten wurden nur diejenigen Dokumente ausgewählt, welche mindestens eine Nachfolgerseite in der Datensammlung enthalten. Diese Selektion erfolgte durch eine Umwandlung der im HTML-Format vorliegenden Dokumente in ein XHTML-Format, um so mittels X-Path eine Extraktion der Vorgänger- bzw. Nachfolgerdokumente zu erreichen (vgl. Abschnitt 4.2:Umwandlung der Dokumente in XHTML). Durch diesen Vorgang wurden ca. 72% der ursprünglichen Dokumente ausgewählt. Tabelle 4.1 zeigt die Klassenverteilung der somit erhaltenen 6.004 Dokumente.

Klasse	Anzahl Dokumente	Anteil
course	555	9,24%
department	117	1,95%
faculty	700	11,66%
other	3.193	53,18%
project	375	6,25%
staff	83	1,38%
student	981	16,34%
Summe	**6.004**	**100,00%**

Tabelle 4.1: Klassenverteilung WebKB

Bei der Betrachtung der Verteilung fällt auf, dass über 50% der Hypertext-Dokumente der Klasse „other“ zugeordnet werden, während die Klasse „staff“ nur ca. 1% der Dokumente ausmacht. Diese Verteilung von Dokumenten ist daher sehr schwierig zu klassifizieren.

4.1.2 BankSearch

Die zweite verwendete Datenbank ist die BankSearch-Datensammlung. Hierbei handelt es sich um eine Sammlung von 11.000 Hypertexten, welche in 11 verschiedene Klassen eingeteilt wurden (Sinka & Corne, 2002). Innerhalb dieser Sammlung gibt es Klassen, die eine hohe Ähnlichkeit aufweisen, wie z. B. Klassen von Investmentbanken neben Klassen von Publikumsbanken. Aufgrund dieser Eigenschaft wurde diese Sammlung für einen Vergleich ausgewählt.

Von den insgesamt 11.000 Dokumenten dieser Datensammlung wurden auch hier nur diejenigen Dokumente ausgewählt, welche mindestens eine Nachbarschaftsseite innerhalb der Datensammlung aufweisen (Tabelle 4.2):

Klasse	Anzahl Dokumente	Anteil
Banking-CommercialBanks	453	**7,46%**
Banking-BuildingSocieties	503	**8,28%**
Banking-InsuranceAgencies	674	**11,09%**
Programming-Java	665	**10,94%**
Programming-C	559	**9,20%**
Programming-VB	650	**10,70%**
Science-Astronomy	618	**10,17%**
Science-Biology	520	**8,56%**
Sport-Soccer	440	**7,24%**
Sport-MotorSport	496	**8,16%**
Sport-NoSoccerMotorSport	498	**8,20%**
Summe	**6.076**	**100,00%**

Tabelle 4.2: Klassenverteilung BankSearch2

Um Aussagen über Klassifikationen mit einer unterschiedlichen Anzahl von Dokumenten zu treffen, wurde eine zweite (kleinere) Version dieser Datenbank verwendet, welche nur aus den folgenden Klassen besteht (Tabelle 4.3):

Klasse	Anzahl Dokumente	Anteil
banks	453	**20,82%**
java	665	**30,56%**
astronomy	618	**28,40%**
soccer	440	**20,22%**
Summe	**2.176**	**100,00%**

Tabelle 4.3: Klassenverteilung BankSearch1

4.1.3 ODP

Die dritte Datenbank stellt eine Sammlung von Webseiten dar, welche aus dem ODP-Webverzeichnis (Abbildung 1.1) entnommen wurde. Dieses Webverzeichnis stellt einen Katalog (von Teilen) des Internets dar, der von einer offenen Gemeinschaft von ca. 75.000 Autoren gepflegt wird. Insgesamt werden in diesem Katalog ca. 4 Mio. Seiten in ca. 590.000 Klassen hierarchisch klassifiziert (Stand: Januar 2007). Die Klassifikationen aller Webseiten stehen im RDF-Format (W3C, 2004) für einen Download zur Verfügung.[7]

Als Testdaten wurden fünf Klassen aus der obersten Ebene, die aus 16 Klassen besteht, ausgewählt. Durch eine Breitensuche wurden aus jeder der ausgewählten Klassen (Arts, Business, Computers, Science und Sports) Hypertexte für eine Testdaten-Sammlung ausgesucht. Durch dieses Vorgehen wurde eine maximale Tiefe der Stufe 3 von der über 20 Stufen umfassenden Hierarchie des ODP Web-Verzeichnisses erreicht.

Um auch bei dieser Datenbank Aussagen über eine Klassifizierung verschiedener Datengrößen zu erreichen, wurden drei verschiedene Datenbanken mit einer

[7] http://rdf.dmoz.org/rdf/content.rdf.u8.gz

unterschiedlichen Anzahl von Dokumenten erstellt (Tabelle 4.4, Tabelle 4.5 und Tabelle 4.6):

Klasse	Anzahl Dokumente	Anteil
Arts	19	**20,00%**
Business	20	**21,05%**
Computers	16	**16,84%**
Science	20	**21,05%**
Sports	20	**21,05%**
Summe	**95**	**100,00%**

Tabelle 4.4: Klassenverteilung ODP100

Klasse	Anzahl Dokumente	Anteil
Arts	99	**20,50%**
Business	97	**20,08%**
Computers	98	**20,29%**
Science	90	**18,63%**
Sports	99	**20,50%**
Summe	**483**	**100,00%**

Tabelle 4.5: Klassenverteilung ODP500

Klasse	Anzahl Dokumente	Anteil
Arts	1.048	**19,94%**
Business	1.065	**20,26%**
Computers	1.036	**19,71%**
Science	1.060	**20,16%**
Sports	1.048	**19,94%**
Summe	**5.257**	**100,00%**

Tabelle 4.6: Klassenverteilung ODP5000

Ermittlung der Nachbarschaftsseiten

a) Die **Vorgängerseiten** dieser Sammlung von „realen“ Webseiten wurden mit Hilfe der Suchmaschine „AltaVista“ ermittelt. Durch den Aufruf von „`link:Webseite`“ bekommt man eine nach einem Ranking sortierte HTML-Liste der Vorgängerseiten für die zu untersuchende Webseite. Aus dieser Liste wurden die jeweils 25 ersten Vorgängerseiten (falls vorhanden) ausgewählt und zu der Datensammlung hinzugefügt.

Durch dieses Vorgehen wurden im Mittel ca. 37 Vorgängerseiten für eine Webseite (ODP5000) ermittelt, da eine Vorgängerseite für eine bestimmte Webseite ja auch eine Vorgängerseite anderer Webseiten sein kann, welche nicht in den ausgewählten AltaVista-Einträgen enthalten ist.

b) Die **Nachfolgerseiten** wurden aus den jeweiligen Out-Links der Webseiten ermittelt und (falls vorhanden) zu der Datensammlung hinzugefügt. Im Mittel wurden dadurch ca. 28 Nachfolgerseiten für eine Webseite gefunden und zu der Datensammlung hinzugefügt.

4.2 Umwandlung der Dokumente in XHTML

Für eine Extraktion der Links bzw. für ein Mining der Link-Local Features wurde eine XPath-Methode angewendet (vgl. Abschnitt 6.3.2:Mining der Links durch XPath). Voraussetzung für die Anwendung eines solchen Verfahrens ist die Umwandlung der Dokumente in ein standardisiertes Format XHTML „Extensible HyperText Markup Language“ (W3C, 2000). Durch den standardisierten Aufbau wird sichergestellt, dass bei einem HTML-Dokument alle erforderlichen Merkmale für eine maschinelle Verarbeitung der Daten vorhanden sind, wie es für die Anwendung von XPath-Methoden erforderlich ist. Die Umwandlung in XHTML erfolgt in dieser Arbeit durch eine Java-Implementierung des Programms „Tidy“.[8]

4.3 Zusammenstellung von Archiven

Für jede der Datensammlungen (WebKB, BankSearch1, BankSearch2, ODP100, ODP500 und ODP5000) wurden alle Daten in einem Archiv zusammengefasst.

Diese Archive beinhalten die folgenden Daten:

- die zu testenden HTML-Seiten
- Vorgängerseiten
- Nachfolgerseiten
- eine Liste der Vorgängerseiten im XML-Format
- eine Liste der Nachfolgerseiten im XML-Format
- eine Liste der Ko-Zitierungen im XML-Format
- eine Liste der (bekannten) Klassen aller Test- und Nachbarschaftsseiten

Alle Archive der verwendeten Daten sind auf der beiliegenden DVD enthalten. Eine Beschreibung der Dateien dieses Archivs befindet sich in Tabelle B.2: Archive der Testdaten.

4.4 Strukturen der Nachbarschaften

Datenbanken zur Evaluierung von Methoden zur Hypertext-Klassifikation sollten eine möglichst hohe Anzahl von Nachbarschaftsseiten enthalten, da die meisten Methoden auf den Informationen dieser Nachbarschaftsseiten beruhen. Auch für die Aussage, ob eine Vielzahl von (möglicherweise irrelevanten) Daten einen Einfluss auf die

[8] http://sourceforge.net/projects/jtidy

Vorhersagegenauigkeit einer Klassifikation ausübt, werden möglichst viele Dokumente der Nachbarschaft benötigt.

Eine Analyse der Anzahl von Vorgängerseiten, verbunden durch In-Links, findet im nächsten Abschnitt statt.

4.4.1 In-Links

Bei der Analyse werden nur diejenigen Testdokumente betrachtet, welche mindestens eine Vorgängerseite aufweisen.

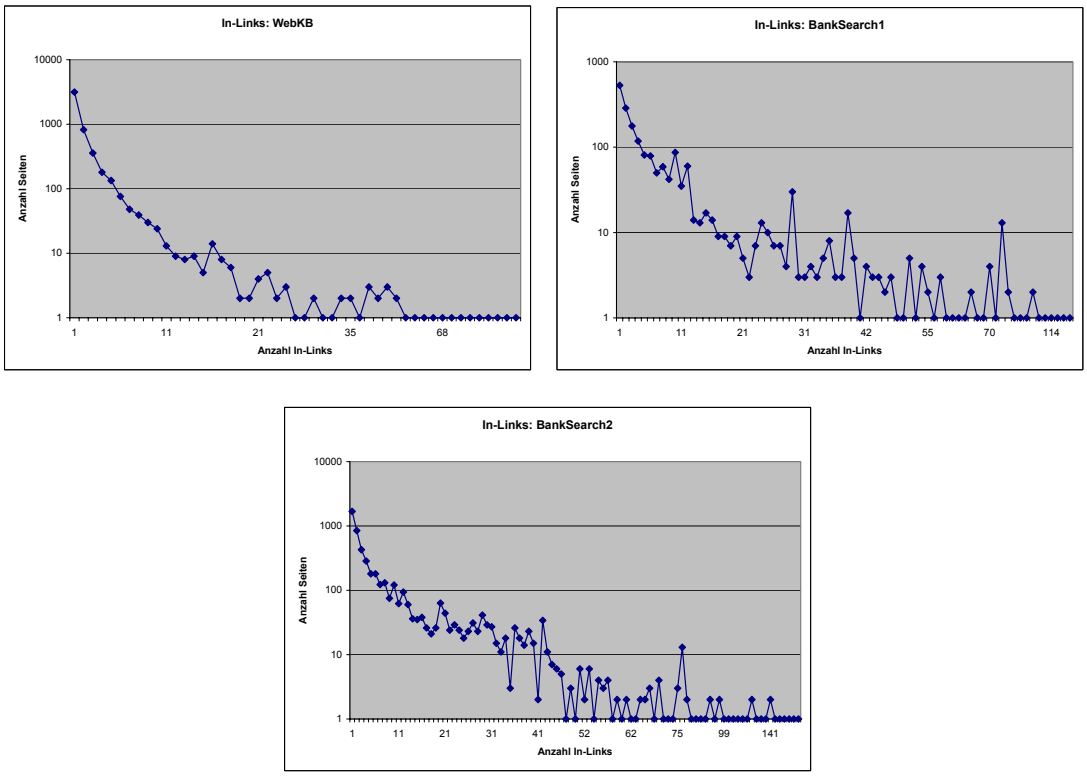

Abbildung 4.1: Verteilung der In-Links in den Datenbanken WebKB und BankSearch

Bei den WebKB- bzw. BankSearch-Datenbanken ist die Anzahl der Vorgängerseiten dadurch beschränkt, dass alle Vorgängerseiten ebenfalls innerhalb der Datenbank liegen. Hierdurch ergeben sich die folgenden Anzahlen von Vorgängerseiten je Testseite (Tabelle 4.7):

Datenbank	Anteil der Seiten mit In-Links	Durchschnittliche Anzahl der In-Links
WebKB	83,0%	2,4
BankSearch1	87,8%	8,7
BankSearch2	84,4%	8,4

Tabelle 4.7: Anzahl In-Links (WebKB und BankSearch)

Bei den ODP-Datenbanken hingegen wurde versucht, durch die in Abschnitt 4.1.3: ODP beschriebenen Methoden eine möglichst große Anzahl von Nachbarschaftsseiten zu

sammeln. Mit steigender Größe der Datenmenge bildet sich eine deutliche Kurve bei der Verteilung der In-Links heraus (Abbildung 4.2).

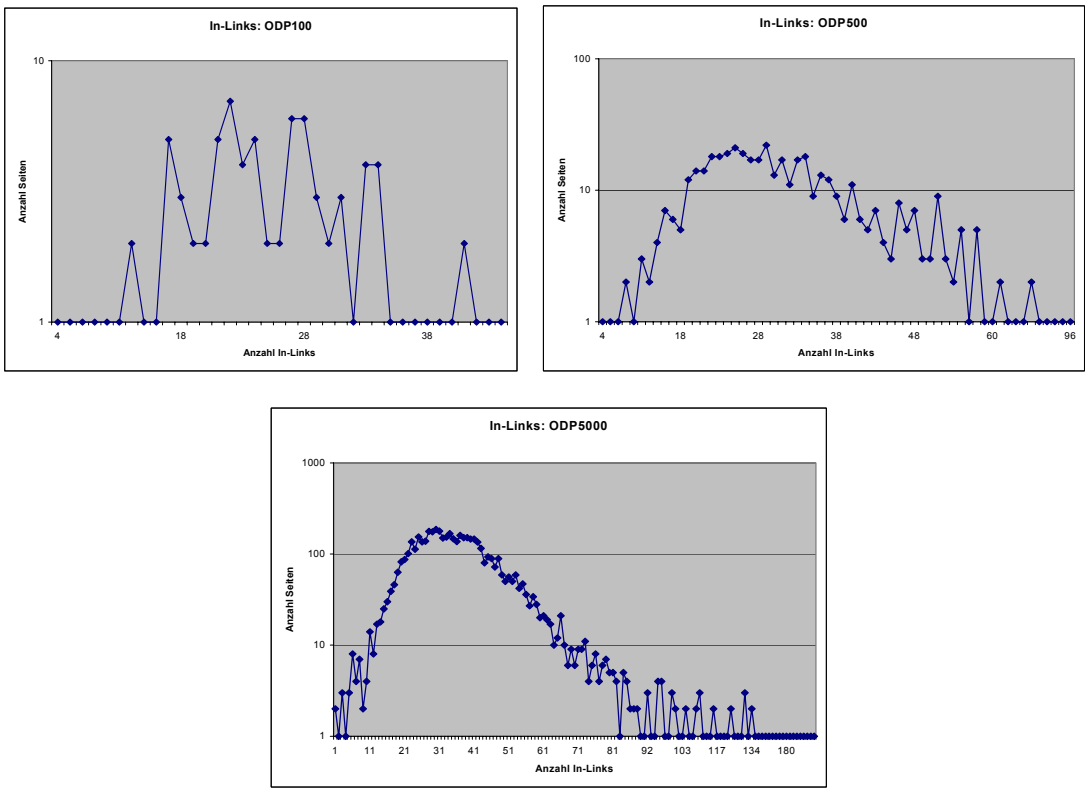

Abbildung 4.2: Verteilung der In-Links in den Datenbanken ODP

Die durchschnittliche Anzahl der Vorgängerseiten einer Testseite der ODP-Datenbanken liegt wesentlich höher als bei den „abgeschlossenen“ WebKB- bzw. BankSearch-Datenbanken (Tabelle 4.8):

Datenbank	Anteil der Seiten mit In-Links	Durchschnittliche Anzahl der In-Links
ODP100	91,6%	29,9
ODP500	93,2%	32,1
ODP5000	94,0%	37,6

Tabelle 4.8: Anzahl In-Links (ODP)

4.4.2 IO-Bridges

Auch bei den (lokalen) IO-Bridges trägt deren Anzahl wesentlich zu der Vorhersagegenauigkeit einer Klassifikation bei. Als Beispiel für eine Link-Verteilung wurde die WebKB- bzw. ODP5000-Datenbank ausgewählt. In Abbildung 4.3 wird eine Verteilung der Anzahl von IO-Bridges dargestellt, wobei nur diejenigen Testseiten analysiert werden, welche auch eine IO-Bridge bzw. eine lokale IO-Bridge aufweisen.

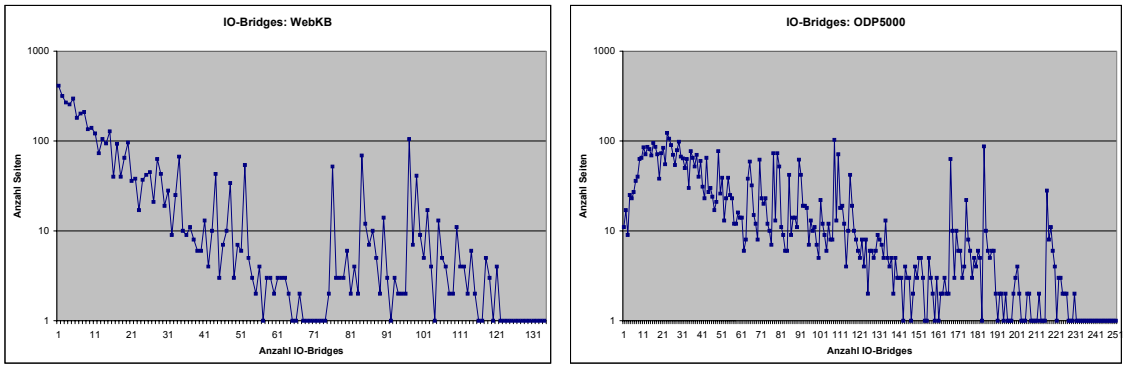

Abbildung 4.3: Verteilung der IO-Bridges in den Datenbanken WebKB und ODP5000

In der Datenbank WebKB beträgt die durchschnittliche Anzahl der IO-Bridges nur 24,4 (75,5% der Testdaten enthalten IO-Bridges), während in der ODP5000-Datenbank durchschnittlich 62,4 IO-Bridges (93,9% der Testdaten enthalten IO-Bridges) je Testseite existieren.

Auch bei den lokalen IO-Bridges unterscheiden sich die Verteilungen der Anzahl (Abbildung 4.4):

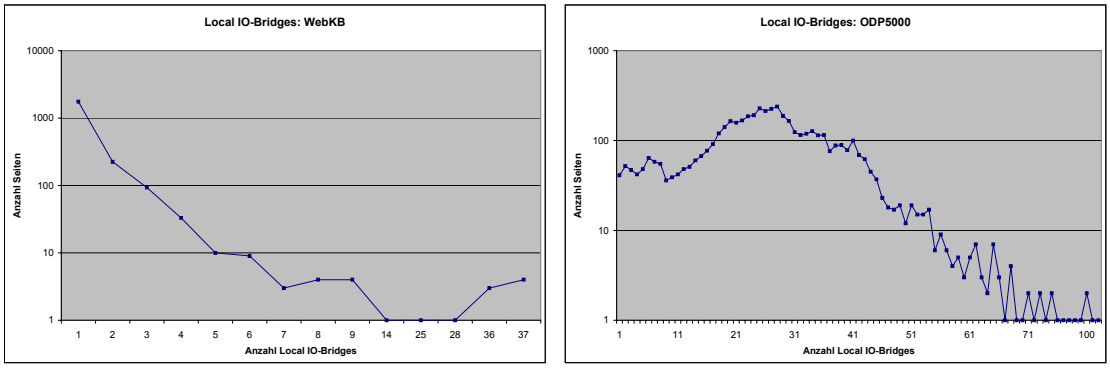

Abbildung 4.4: Verteilung der Local IO-Bridges in den Datenbanken WebKB und ODP5000

In der Datenbank WebKB beträgt die durchschnittliche Anzahl der lokalen IO-Bridges 1,5 (35,6% der Testdaten enthalten lokale IO-Bridges), während in der ODP5000-Datenbank durchschnittlich 26,6 lokale IO-Bridges (93,2% der Testdaten enthalten lokale IO-Bridges) je Testseite vorhanden sind.

4.5 Feature Reduction

Bei den Vergleichen von Methoden zur Hypertext-Klassifikation wurde die *term frequency* (TF), d. h. die Häufigkeit der Features (Wörter) in einem Dokument, als Basis für eine Klassifikation verwendet (vgl. Abschnitt 6.1:Der Naive-Bayes-Klassifikator).

Für ein Experiment wurden 100.000 Features der Datenbank ODP5000 nach Häufigkeiten sortiert. Es wurde untersucht, wie sich eine Reduzierung der Anzahl von

verwendeten Features (*feature reduction*) auf die Vorhersagegenauigkeit einer Klassifikation auswirkt (Abbildung 4.5).

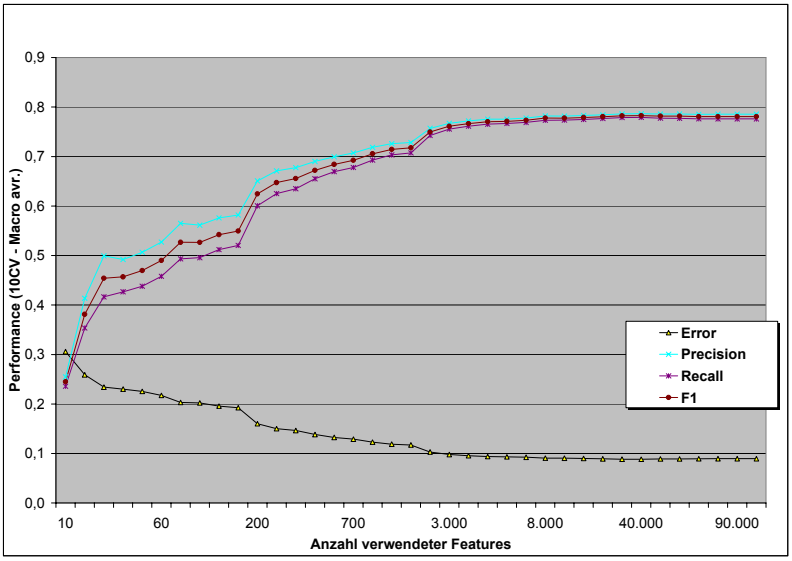

Abbildung 4.5: Vorhersagegenauigkeit bei unterschiedlicher Anzahl verwendeter Features

Hierbei wurde festgestellt, dass schon bei der Verwendung von ca. 3% der Features eine Klassifikation ohne einen nennenswerten Verlust der Vorhersagegenauigkeits-Werte möglich ist. Diese Ergebnisse entsprechen den Ergebnissen einer Studie, in der verschiedene Methoden zur Feature Selection miteinander verglichen wurden (Yang & Pedersen, 1997). In der Studie wurde festgestellt, dass die Term Frequency eine effiziente Methode darstellt, um die Dimensionalität einer Klassifikation ohne (nennenswerte) negative Auswirkungen auf die Vorhersagegenauigkeit zu reduzieren.

Da die Laufzeit einer Klassifikation mit der Anzahl der verwendeten Features überproportional anwächst, wurden die Messungen dieser Arbeit mit einer Feature Reduction von 1:10 durchgeführt, d. h. es werden nur die 10% der am häufigsten auftretenden Features für eine Klassifikation verwendet.

Eine Ausnahme hiervon bilden Feature-Mengen, die Klassen von Dokumenten darstellen (vgl. Abschnitt 3.3.2:Klassen der Nachbarn). Aufgrund der niedrigen Feature-Anzahl (diese entspricht der Anzahl der Klassen) wird hierbei auf eine Feature Reduction verzichtet.

Um die in diesem Kapitel beschriebenen Datenbanken für eine systematische Evaluierung der Methoden zur Hypertext-Klassifikation zu nutzen, bedarf es einer geeigneten Messmethodik. Diese wird in dem folgenden Kapitel beschrieben.

5 Messmethodik

In diesem Kapitel werden die Kennzahlen zur Evaluierung, das in den Experimenten angewandte Verfahren der Cross-Validation sowie die Vergleichsmethodik für die durchgeführten Experimente dargestellt.

5.1 Evaluierung der Ergebnisse

5.1.1 Kennzahlen zur Evaluierung

Um die Ergebnisse unterschiedlicher Methoden zur Klassifikation vergleichen zu können, bedarf es einheitlicher Evaluierungsmaßstäbe. Bei einer Klassifikation werden hierfür üblicherweise die folgenden Kennzahlen verwendet: Accuracy, *Precision*, *Recall* sowie die Funktion F_β. (Rijsbergen, 1979).

Grundlage der Berechnung dieser Kennzahlen ist die *Confusion-Matrix*, in der die Ergebnisse einer Klassifikation abgebildet werden. In dem folgenden Beispiel werden die Ergebnisse eines Klassifikators dargestellt, der Dokumente in zwei verschiedene bekannte Klassen (positiv, negativ) einordnet (Tabelle 5.1):

	positiv klassifiziert	negativ klassifiziert	
Ist positiv	**a**	**c**	a+c
Ist negativ	**b**	**d**	b+d
	a+b	c+d	n

Tabelle 5.1: Ergebnisse einer Klassifikation: 2x2 Confusion-Matrix

Aus dieser Confusion-Matrix werden die folgenden Kennzahlen berechnet:

1. ***Accuracy A***: Beschreibt den Anteil der Dokumente, welche richtig klassifiziert werden: $A = \frac{(a+d)}{n}$. Die Verwendung der Accuracy kann jedoch zu Fehleinschätzungen führen, da diese z. B. mit der Anzahl der irrelevanten Dokumente (d) anwächst.
2. ***Error E***: Bezeichnet die Fehlerrate E = 1-A.
3. ***Precision*** π: Beschreibt den Anteil der positiven Dokumente von allen positiv klassifizierten Dokumenten: $\pi = \frac{a}{(a+b)}$.
4. ***Recall*** ρ: Beschreibt den Anteil der positiv klassifizierten Dokumente von allen positiven Dokumenten: $\rho = \frac{a}{(a+c)}$.

5. ***Funktion F_β***: Die Funktion $F_\beta = \frac{(\beta^2 + 1)\rho\pi}{\beta^2(\rho + \pi)}$ ist eine gewichtete Kombination von Precision und Recall. Üblicherweise wird der Wert von $\mathbf{F_1}$ verwendet, der eine Gleichgewichtung zwischen Precision und Recall darstellt: $F_1 = \frac{2\rho\pi}{\rho + \pi}$.

Um die Ergebnisse dieser Arbeit vergleichen zu können, werden alle diese Kennzahlen in den Experimenten gemessen. Eine detaillierte Aufstellung der Messergebnisse befindet sich im Anhang A:Tabellen der Messergebnisse.

5.1.2 Mittelwertbildung der Kennzahlen über mehrere Klassen

Bei der Evaluierung von Klassifikatoren, die mehr als zwei Klassen zurückliefern, besteht die Notwendigkeit, die in 5.1.1 beschriebenen Kennzahlen für jede Klasse zu ermitteln. Für eine Mittelwertbildung dieser Kennzahlen existieren zwei verschiedene Verfahren:

1. **Micro-averaging** (Bildung eines Datenmittelwertes): Hierbei werden die Ergebnisse einer Klassifikation für jede Klasse getrennt in einer 2x2-Confusion-Matrix ermittelt. Diese Matrizen werden in einer globalen Confusion-Matrix addiert. Die Berechnung der Kennzahlen erfolgt nun nach den in 5.1.1 beschriebenen Methoden auf der globalen 2x2-Confusion-Matrix.
2. **Macro-averaging** (Bildung eines Klassenmittelwertes): Bei dieser Methode werden erst die Kennzahlen für jede Klasse getrennt berechnet und dann ein Mittelwert über alle Klassen gebildet.

Das Verfahren des Micro-averaging berücksichtigt eher Klassen mit vielen Dokumenten bei der Berechnung der Kennzahlen, während Kennzahlen, die mit Macro-averaging berechnet werden, eher auch von Klassen mit einer geringeren Anzahl von Dokumenten beeinflusst werden (Yang, Y. & Liu, X., 1999).

Aus diesem Grund wurden die Kennzahlen dieser Arbeit mit Hilfe des Macro-averaging-Verfahrens berechnet, wobei der Unterschied zwischen diesen Verfahren im Fall der ODP-Datenbanken vernachlässigbar ist, da diese eine annähernde Gleichverteilung der Klassen aufweisen (vgl. Tabelle 4.4 bis Tabelle 4.6).

5.1.3 Beispiel einer Evaluierung

Als Beispiel werden die Kennzahlen einer Klassifikation mit drei Klassen berechnet. In Tabelle 5.2 befindet sich die Confusion-Matrix dieses Klassifikators sowie die berechneten Werte für Precision und Recall je Klasse.

	Klassifiziert als Klasse A	als Klasse B	als Klasse C	Recall
Ist Klasse A	**10**	3	2	0,67
Ist Klasse B	5	**20**	4	0,69
Ist Klasse C	1	1	**2**	0,50
Precision	0,63	0,83	0,25	

Tabelle 5.2: Beispiel einer Evaluierung mit drei Klassen

1. Micro-averaging:

a) Berechnung der 2x2-Confusion-Matrix (für jede Klasse)

Klassifiziert	A		B		C	
	+	-	+	-	+	-
Ist A / Ist B / Ist C	10	5	20	9	2	2
Ist nicht A / Ist nicht B / Ist nicht C	6	27	4	15	6	38

b) Berechnung der globalen 2x2-Confusion-Matrix (Summen)

	Klassifiziert pos.	neg.
Ist positiv	**32**	**16**
Ist negativ	**16**	**80**

c) Berechnung der Kennzahlen

Precision: 32/(32+16) = 0,67

Recall: 32/(32+16) = 0,67

F_1: (2*0,67*0,67)/(0,67+0,67) = 0,67

2. Macro-averaging:

Precision: (0,63+0,83+0,25)/3 = 0,57

Recall: (0,67+0,69+0,5)/3 = 0,62

F_1: (2*0,57*0,62)/(0,57+0,62) = 0,71/1,19 = 0,59

Accuracy/Error: Diese sind sowohl beim Micro- als auch beim Macro-averaging-Verfahren identisch. Am einfachsten lassen sich diese Kennzahlen aus der globalen 2x2-Confusion-Matrix (b) berechnen. In diesem Beispiel beträgt die Accuracy = (32+80)/144 = 0,78. Der Error liegt bei 1-0,78 = 0,22.

5.2 Cross-Validation

Um ein Modell auf seine Gültigkeit zu überprüfen, wird die zur Verfügung stehende Datenmenge in eine (größere) Trainingsmenge und in eine (kleinere) Testmenge

aufgeteilt (*Holdout Testing*). Die Trainingsmenge wird dazu verwendet, das Modell aufzustellen; die Testmenge soll das Modell bestätigen (Lohninger, 2006).

Bei einer *k-fold Cross-Validation (CV)* von C Klassen wird die zur Verfügung stehende Datenmenge D^c einer Klasse $c \in C$ in k annähernd gleich große exklusive Teilmengen (folds) $D^c_1,\ldots,D^c_k$ aufgeteilt. Dieser Vorgang wird für alle Klassen $c \in C$ wiederholt. Dadurch wird sichergestellt, dass die proportionale Verteilung der Klassen in den Teilmengen annähernd erhalten bleibt (*stratified cross-validation*).

Die Zuordnung zu einer Teilmenge erfolgt durch eine *Holdout-Funktion*, welche sicherstellt, dass ein Dokument nur einer einzigen Teilmenge zugeordnet wird. In dieser Arbeit wird als Holdout-Funktion die Funktion $f(Doc_c)=(Doc_c\text{-}1 \bmod k)+1$ verwendet. Hierbei ist Doc_c die Ordnung eines Dokumentes innerhalb der Klasse c.

Nun werden k Testdurchläufe gestartet; bei jedem Durchlauf $t \in \{1, 2, \ldots, k\}$ wird die Teilmenge D_t als Testmenge und die verbleibenden Teilmengen $D \backslash D_t$ als Trainingsmengen verwendet (Kohavi, 1995).

Alle Messergebnisse dieser Arbeit wurden mit k=10 berechnet (10-fold CV). Um eine Vergleichbarkeit mit vorhergehenden Arbeiten sicherzustellen, wurde bei der WebKB-Datenbank zusätzlich eine 4-fold CV und eine 5-fold CV durchgeführt.

5.3 Vergleichsmethodik

Um die verschiedenen Methoden zur Hypertext-Klassifikation mit unterschiedlichen Datenbanken (Sammlungen von Dokumenten) vergleichen zu können, wurde das folgende Vorgehen gewählt:

Für jede Datenbank werden die *Basiswerte* der Kennzahlen ermittelt. Diese bestehen aus den in Abschnitt 5.1:Evaluierung der Ergebnisse beschriebenen Kennzahlen, die bei einer Text-Klassifikation der Dokumente erreicht werden (Methode: „Own Text“).

Für ein Beispiel werden die Basiswerte der Kennzahlen für die Datenbank „WebKB“ dargestellt (Tabelle 5.3):

	Messwerte				
Methode	**Accuracy**	**Error**	**Precision**	**Recall**	**F1**
OwnText	0,7813	0,2187	0,3166	0,3340	0,3250

Tabelle 5.3: Basiswerte der Kennzahlen

Nun werden die Kennzahlen mit der zu vergleichenden Methode berechnet. In diesem Beispiel wird die Methode „Words Around 10“ (WA 10) in der gleichen Datenbank getestet (Tabelle 5.4):

Messwerte					
Methode	Accuracy	Error	Precision	Recall	F1
WA 10	0,8709	0,1291	0,4485	0,4786	0,4631

Tabelle 5.4: Kennzahlen der Methode „WA 10“

Für einen Vergleich werden nun die *prozentualen Abweichungen* zu den jeweiligen Basiswerten betrachtet (Tabelle 5.5):

Abweichungen in % (zu Own Text)			
Accuracy	Precision	Recall	F1
11,47%	41,66%	43,29%	42,49%

Tabelle 5.5: Prozentuale Abweichungen der Methode „WA 10“

Hier liegt z. B. der F_1-Wert der Methode „WA 10“ um 42,49% höher als der F_1-Wert einer Text-Klassifikation.

Die in diesem Kapitel beschriebene Messmethodik dient als Grundlage für die im Rahmen dieser Arbeit durchgeführten Experimente, deren Aufbau in dem folgenden Kapitel beschrieben wird.

6 Experimente

In diesem Kapitel werden der Aufbau und die Durchführung der vorgenommenen Experimente beschrieben. Es beinhaltet eine Beschreibung des verwendeten Text-Klassifikators (ein Naive-Bayes-Klassifikator), eine Vorstellung des implementierten Testframeworks, die Vorverarbeitung der Daten sowie die technischen Bedingungen und Laufzeiten, unter denen die Experimente durchgeführt wurden.

6.1 Der Naive-Bayes-Klassifikator

Naive-Bayes-Klassifikatoren sind im Allgemeinen einfach zu verstehende Klassifikatoren, die schnelle Ergebnisse erbringen, da sie nur einen Durchlauf durch die Trainingsdaten benötigen (Kohavi, 1996).

6.1.1 Bayes-Theorem

Bayes-Klassifikatoren berechnen die Wahrscheinlichkeit, dass ein Objekt x einer der bekannten Klassen ($C_1 \ldots C_k$) angehört. Grundlage für die Berechnungen ist das *Bayes-Theorem* (6.1). Es besagt, dass die *A-posteriori*-Wahrscheinlichkeit einer Hypothese A unter Annahme einer Hypothese B anhand der *A-priori*-Wahrscheinlichkeiten von A und B berechnet werden kann.

$$p(A \mid B) = \frac{p(B \mid A)\,p(A)}{p(B)} \tag{6.1}$$

6.1.2 Naive-Bayes-Klassifikatoren

Der Naive-Bayes-Klassifikator beruht auf der Annahme, dass sich die einzelnen Attribute x_i eines Objektes x nicht untereinander beeinflussen (6.5). Er bestimmt die Klasse C_i eines Objektes x, indem er die Klasse mit der höchsten A-posteriori-Wahrscheinlichkeit $p(C_i|x)$ auswählt. Diese Wahrscheinlichkeiten lassen sich mit (6.2) berechnen.

$$p(C_i \mid x) = \frac{p(x \mid C_i)\,p(C_i)}{p(x)} \tag{6.2}$$

Da p(x) für alle Klassen identisch ist, wird diejenige Klasse ausgewählt, für die $p(x|C_i)p(C_i)$ maximal ist (6.3).

$$\arg\max_{C_i \in (C_1 \ldots C_k)} p(x \mid C_i)\,p(C_i) \tag{6.3}$$

Die Werte von $p(C_i)$ können durch die Verteilung der k Klassen in den Trainingsdaten T berechnet werden. Hierbei wird die Anzahl der Trainingsdokumente in der Klasse C_i durch die Anzahl aller Trainingsdokumente |T| geteilt (6.4).

$$p(C_i) = \frac{\left|\{x \in T \mid x \in C_i\}\right|}{\left|T\right|} \tag{6.4}$$

$p(x|C_i)$ lässt sich unter der Annahme berechnen, dass sich die einzelnen Attribute x_i eines Objektes x (hier: die Wörter der d Dokumente) nicht gegenseitig beeinflussen (*Unabhängigkeitsannahme*), wie es z. B. bei dem mehrmaligen Werfen eines Würfels der Fall ist (6.5).

$$p(x \mid C_i) = \prod_{j=1}^{d} p(x_j \mid C_i) \tag{6.5}$$

Die Berechnung der Werte von $p(x_j|C_i)$ erfolgt durch eine Auswertung der Trainingsdaten T. Hierbei werden die Häufigkeiten der Attribute in den einzelnen Dokumenten ermittelt (6.6).

$$p(x_j \mid C_i) = \frac{\left|\{y \in T \mid y \in C_i \wedge y_j = x_j\}\right|}{\left|\{y \in T \mid y \in C_i\}\right|} \tag{6.6}$$

In der Praxis kommt es durchaus vor, dass nicht alle verschiedenen Attribute x_j in einem Objekt x vorhanden sein werden (hier: nicht alle verschiedenen Wörter einer Dokumentensammlung kommen in jedem einzelnen Dokument x vor). In einem solchen Fall würde die Wahrscheinlichkeit $p(x_j|C_i)$ auf Null gesetzt werden. Um dieser Fehlerquelle entgegenzuwirken, werden die Werte z. B. mit Hilfe der *Laplace-Korrektur* ermittelt, welche die Anzahl V der verschiedenen Wörter einer Dokumentensammlung verwendet (6.7).

$$p(x_j \mid C_i) = \frac{\left|\{y \in T \mid y \in C_i \wedge y_j = x_j\}\right| + 1}{\left|\{y \in T \mid y \in C_i\}\right| + V} \tag{6.7}$$

Naive-Bayes-Klassifikatoren erreichen zwar nicht die Vorhersagegenauigkeit anderer Methoden zur Text-Klassifikation, wie z. B. Support-Vektor-Maschinen, jedoch ist ihre Genauigkeit für die Zwecke einer Text-Klassifikation ausreichend (Chakrabarti, S., Roy, S. & Soundalgekar, M., 2003). Dank ihrer hohen Geschwindigkeit bei der Klassifikation besitzen sie ein gutes Verhältnis zwischen Geschwindigkeit und Genauigkeit.

Aufgrund der Tatsache, dass es sich bei den folgenden Experimenten um Datenmengen mit einer hohen Anzahl von Attributen (Features) handelt, wurden die Klassifikationen durch einen eigens hierfür implementierten Naive-Bayes-Klassifikator durchgeführt.

6.2 Verwendung unterschiedlicher Feature-Gruppen

Bei der Untersuchung von Methoden zur Hypertext-Klassifikation ist es oft notwendig, die Ergebnisse verschiedener Methoden miteinander zu kombinieren. Jede Methode liefert hierbei eine eigene Menge von Attributen (Features).

6.2.1 Verwendung eines einzelnen Klassifikators

Um die Features verschiedener Methoden unter Verwendung eines einzelnen Klassifikators zu kombinieren, gibt es zwei unterschiedliche Verfahren: *Merging* und *Tagging*. Beide Verfahren nutzen ein gemeinsames Dokument (*Meta-Dokument*), um die Features zusammenzuführen. Das Meta-Dokument wird dann für das Training und das Testen des Klassifikators verwendet (Utard, 2005).

6.2.1.1 Merging

Beim Merging erfolgt keine Differenzierung nach Methoden. Falls ein Feature in mehreren Methoden vorhanden ist, wird es auch entsprechend oft im Meta-Dokument aufgenommen (Abbildung 6.1):

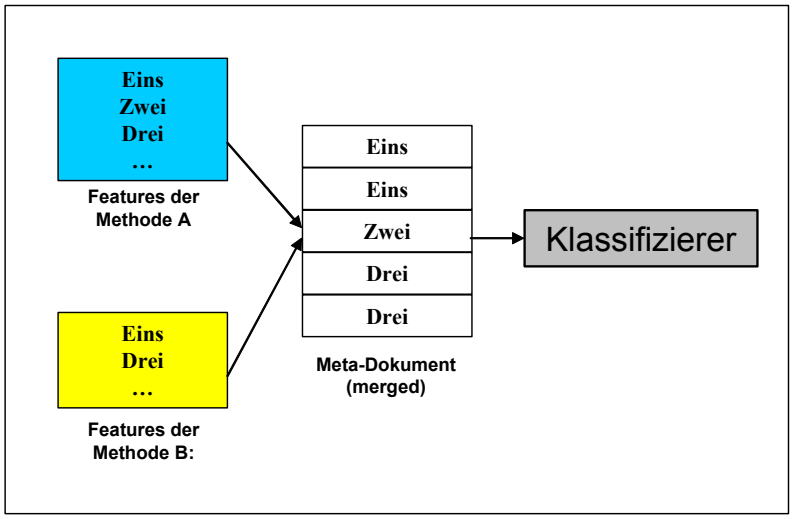

Abbildung 6.1: Merging von Features

Bei diesem Verfahren können gleiche Features unterschiedlicher Methoden vom Klassifikator nicht mehr unterschieden werden, jedoch wird die Redundanz oft auftretender Features erhöht.

6.2.1.2 Tagging

Beim Tagging wird die Methode eines Features durch ein entsprechendes Attribut (Tag) im Meta-Dokument gekennzeichnet. Durch dieses Verfahren werden gleiche Features unterschiedlicher Methoden vom Klassifikator getrennt behandelt (Abbildung 6.2):

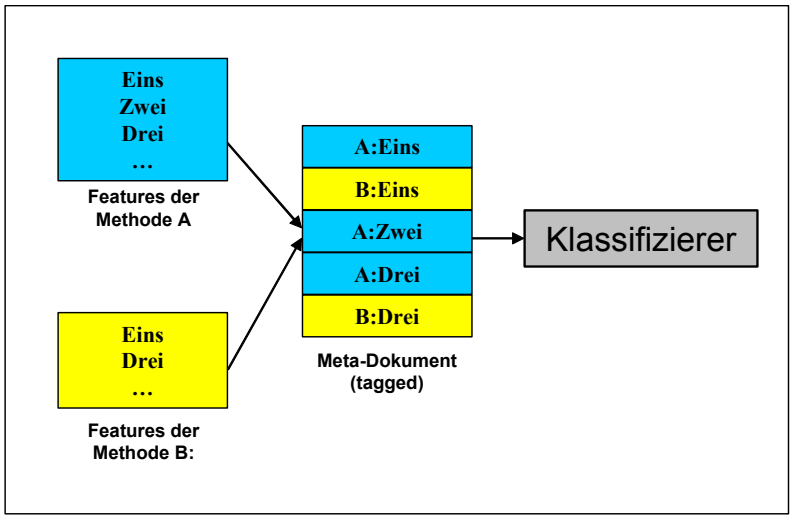

Abbildung 6.2: Tagging von Features

Der Nachteil dieser Methode ist, dass dadurch die Redundanz oft auftretender Features herabgesetzt wird.

Utard (2005) untersuchte beide Verfahren im Rahmen von Versuchen zu Link-Local-Methoden. Hierbei wurde festgestellt, das Merging eine geeignete Methode zur Verwendung von mehreren Features mit nur einem Klassifikator darstellt.

6.2.2 Verwendung mehrerer Klassifikatoren

Hierbei werden die Features der unterschiedlichen Methoden auf getrennten Klassifikatoren trainiert und getestet (Abbildung 6.3):

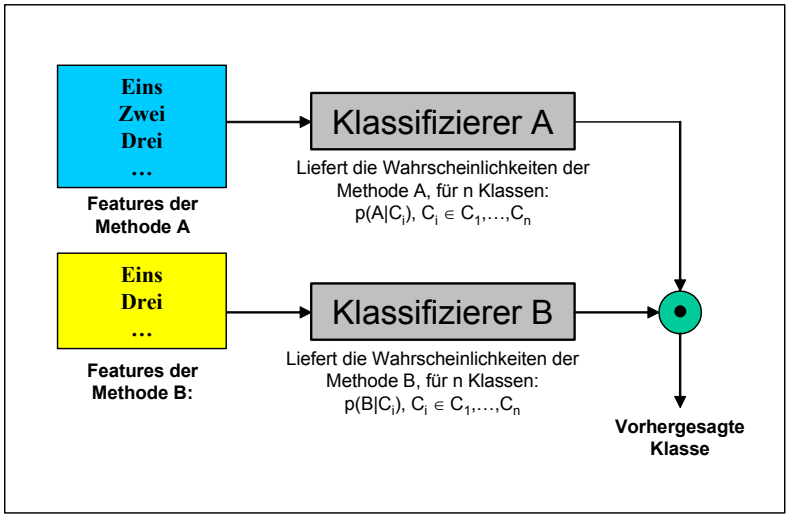

Abbildung 6.3: Verwendung mehrerer Klassifikatoren

Um die Vorhersage für eine Klasse $c \in \{c_1,...,c_n\}$ zu erhalten, werden die Ergebnisse der einzelnen Klassifikatoren (die Wahrscheinlichkeiten für eine Klasse c) durch eine Multiplikation zusammengeführt. Bei einer Klassifikation von x bildet die Klasse mit dem höchsten Produkt der Wahrscheinlichkeiten die vorhergesagte Klasse c_x (6.8):

$$c_x = \underset{C_i \in (C_1 \ldots C_k)}{\arg\max}\, p(A_x \mid C_i)\, p(B_x \mid C_i)\, p(C_i) \tag{6.8}$$

Hierbei wird auch die A-priori-Wahrscheinlichkeit $p(C_i)$ berücksichtigt, die aus der Klassenverteilung der Trainingsdaten berechnet wird (6.4).

Diese Zusammenführung der Wahrscheinlichkeiten erzielte auch bei den Untersuchungen von Fürnkranz (2002) gute Ergebnisse. Eine Alternative hierzu bietet das *Voting* von Ergebnissen. Dabei werden die Ergebnisse (vorausgesagte Klassen) der einzelnen Klassifizierer durch einen Abstimmungsprozess zusammengeführt (Utard, 2005).

6.3 Durchführung der Experimente

6.3.1 Vorverarbeitung der Dokumente

Die Vorverarbeitung der Dokumente erfolgt in mehreren Schritten. Zuerst werden evtl. vorhandene *Header* (wie sie z. B. bei WebKB-Dokumenten existieren) entfernt. Diese Header beinhalten Angaben zur Klassifikation und könnten somit eine Verfälschung der Messergebnisse bewirken.

Anschließend werden die *HTML-Tags* aus den Dokumenten entfernt. Diese Tags steuern die Formatierung der HTML-Seite und die Darstellung in einem Webbrowser. Durch eine Filterung der Dokumente mit Hilfe des *regulären Ausdrucks* „<[^>]+>“ kann eine effiziente Entfernung der Tags erreicht werden.

Nach der Umwandlung des Textes in *Kleinbuchstaben* werden *Akzentzeichen* entfernt und *Umlaute* ersetzt; beispielsweise wird der Buchstabe „ä“ gegen „ae“ ausgetauscht.

Wörter, die in Texten häufig vorkommen (*Stoppwörter*), werden durch den Vergleich mit einer Stoppwortliste entfernt. Da es sich bei den Dokumenten um englischsprachige Texte handelt, wird die weit verbreitete SMART-Stoppwortliste[9] verwendet, die aus 571 häufig benutzten englischen Wörtern besteht. Ein *Stemming*, d. h. ein Zurückführen der einzelnen Wörter auf ihren Wortstamm, wird nicht durchgeführt.

Abschließend werden alle *Ziffernfolgen* durch ein „D“ bzw. alle *Sonderzeichenfolgen* durch ein „_“ ersetzt sowie Sonderzeichen am Anfang und am Ende eines Wortes entfernt.

[9] Verfügbar unter: ftp://ftp.cs.cornell.edu/pub/smart/

6.3.2 Mining der Links durch XPath

Die Extraktion der Links eines Dokumentes und die Extraktion (Mining) der Features für die *Link-Local-Methoden* erfolgt mittels Zugriff auf die in XHTML-Form abgespeicherten Daten durch XPath-Methoden (W3C, 1999). Da Link-Local-Methoden nur relevante Wörter der Vorgängerseiten als Features verwenden (vgl. Abschnitt 3.3.3: Link-Local-Methoden), werden XPath-Extraktionen nur bei In-Links durchgeführt.

XPath ist eine vom W3C-Konsortium standardisierte Anfragesprache, um Teile eines XML-Dokumentes zu adressieren, die als Baum zurückgegeben werden. Für das Mining der Features für Link-Local-Methoden wurden die in Tabelle 6.1 aufgeführten XPath-Anfragen verwendet (Utard, 2005).

Beispielsweise liefert die XPath-Anfrage `"//a[@href='"+linkName+"']"` als Ergebnis eine Liste der Link-Descriptions des Links „linkname“.

Feature	XPath-Anfrage
Link Description	"//a[@href=' + linkName + "']"
Link Paragraph	"//a[@href=' + linkName + "']/ancestor::p[last()]"
Link Headers	"//a[@href=' + linkName + "']/preceding::h1[last()]" + "| //a[@href=' + linkName + "']/preceding::h2[last()]" + "| //a[@href=' + linkName + "']/preceding::h3[last()]"
Link List Headers	"//a[@href=' + linkName + "']/ancestor" + "::ul/preceding::h1[last()]" + "| //a[@href=' + linkName + "']/ancestor" + "::ul/preceding::h2[last()]" + "| //a[@href=' + linkName + "']/ancestor" + "::ul/preceding::h3[last()]"

Tabelle 6.1: Mining von Link-Local-Features mit XPath

6.3.3 Implementierung eines Test-Frameworks

Das für diese Experimente implementierte Framework wurde implementiert, um die folgenden Ziele zu erreichen:

- Sammlung der Daten durch den Aufruf von HTML-Seiten
- Aufbereitung der Daten (wie z. B. die Konvertierung in XHTML)
- Durchführung der Experimente
- Reporting der Ergebnisse und Export in andere Systeme zur graphischen Darstellung (wie z. B. MS-Excel)

Eine Implementierung sollte in einer allgemein bekannten und frei verfügbaren Programmiersprache erfolgen. Hierdurch lassen sich schon vorhandene Module nutzen, wie sie z. B. für die Umwandlung in ein XHTML-Format (durch JTidy) existieren.

Auch sollte eine Portierbarkeit des Frameworks auf verschiedene System-Plattformen (wie z. B. UNIX oder MS-Windows) gewährleistet sein.

Aus diesen Gründen wurde eine Implementierung in der Programmiersprache *JAVA* (Sun Microsystems, 2007) vorgenommen; eine Beschreibung der implementierten Klassen befindet sich im Anhang B.1:Java-Klassen des Test-Frameworks. Als Entwicklungsumgebung wurde Eclipse (Eclipse Foundation, 2007) eingesetzt.

6.3.4 Verwendete Rechner

Alle Experimente wurden auf Rechnern mit den folgenden Kenndaten durchgeführt:

(1) Linux Workstation

2x AMD-Opteron Prozessor (2,4 GHz) unter dem Linux Kernel 2.6.11

(2) Dell Latitude D820

Intel T2500-Prozessor (2 GHz) unter MS-Windows XP, Version 5.1 SP2

6.3.5 Laufzeiten der Experimente

Eine Berechnung aller Methoden einer Datenbank (Anhang A:Tabellen der Messergebnisse) auf der beschriebenen Linux-Workstation (1) benötigte die in Tabelle 6.2 angegebenen Laufzeiten. Die Laufzeiten unter MS-Windows (2) liegen, unter vergleichbaren Bedingungen, um den Faktor 1,4 höher. Die Laufzeiten hängen dabei von der Anzahl der Testdokumente sowie von der Anzahl der analysierten Nachbarschaftsseiten ab.

Datenbank	Laufzeit (in hh:min)
WebKB	02:03
BankSearch1	01:42
BankSearch2	05:49
ODP100	00:21
ODP500	01:57
ODP5000	23:01

Tabelle 6.2: Laufzeiten der Experimente

Die Ergebnisse der Experimente, die mit dem implementierten Test-Framework erzielt wurden, werden im nächsten Kapitel dargestellt.

7 Ergebnisse

In diesem Kapitel werden die Ergebnisse der durchgeführten Experimente präsentiert. Diese wurden auf den im Kapitel 4:Datenbanken zur Evaluierung vorgestellten Testdaten durchgeführt.

Um zu bestimmen, welchen Einfluss die Größe der Nachbarschaft eines In-Links auf die Vorhersagegenauigkeit einer Klassifikation ausübt, wurden im Abschnitt 7.1 Messungen bei verschiedenen Größen dieser Nachbarschaft durchgeführt. In Abschnitt 7.2 wird in einem Experiment die Frage untersucht, inwieweit die Anzahl verwendeter In-Links einen Einfluss auf die Vorhersagegenauigkeit einer Klassifikation ausübt. Diese einleitenden Experimente dienen zur Bestimmung von Parametern bei den nachfolgenden Untersuchungen.

Im Hauptteil dieses Kapitels erfolgt ein systematischer Vergleich der Methoden zur Hypertext-Klassifikation. (Details der Messergebnisse können im Anhang A:Tabellen der Messergebnisse nachgeschlagen werden.) Hierbei wird besonders die Vorhersagegenauigkeit (F_1) einer Klassifikation untersucht (Abschnitt 7.3).

Um einen Vergleich der Methoden über verschiedene Datensammlungen zu erhalten, erfolgt eine Mittelwertbildung der Vorhersagegenauigkeiten über alle verwendeten Testdaten (Abschnitt 7.4) sowie eine anschließende Betrachtung der Coverage (Abschnitt 7.5). Messungen von Vorhersagegenauigkeits-Werten bei gleicher Coverage ergänzen den Vergleich der Methoden (Abschnitt 7.6).

Ein Experiment zeigt, dass durch die Kombination verschiedener Methoden noch eine weitere Steigerung der Vorhersagegenauigkeit möglich ist (Abschnitt 7.7).

Als Abschluss und als (praktisches) Resultat dieser Arbeit wird ein universeller Klassifizierer vorgestellt, der eine Steigerung der Vorhersagegenauigkeit bei voller Coverage erreicht (Abschnitt 7.8).

7.1 Nachbarschaften eines In-LInks

Chakrabarti et al. stellten in ihrer Arbeit von 1998 fest, dass bei der Verwendung von vielen (möglicherweise irrelevanten) Features ein Einbruch der Vorhersagegenauigkeit einer Klassifikation erfolgt (vgl. Abschnitt 3.3.1:Texte der Nachbarn).

Um diesen Effekt nachzuvollziehen, wurde in dem folgenden Experiment die Menge der verwendeten Features einer Klassifikation schrittweise erhöht und dabei die Vorhersagegenauigkeit der Klassifikation beobachtet (Abbildung 7.1). Hierfür wurde die Link-Local-Methode *Words Around* verwendet, die Wörter in einem Umfeld der In-

Links als Features für eine Klassifikation zugrunde legt. Durch eine Erhöhung dieses Radius werden immer mehr Features verwendet, bis schließlich alle Wörter der Vorgängerseiten als Features einbezogen werden:

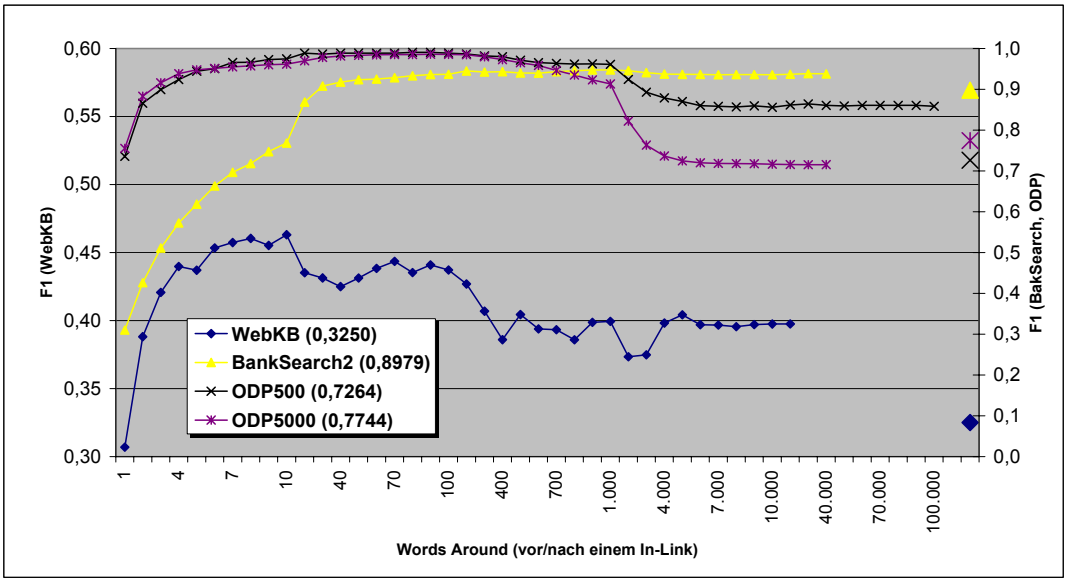

Abbildung 7.1: Vorhersagegenauigkeiten in Abhängigkeit von der Anzahl der Worte vor/nach einem In-Link

Nach einer anfänglichen Steigerung der Vorhersagegenauigkeit kann zwar ein Rückgang (im Bereich von ca. 1000 Wörtern) beobachtet werden, jedoch fällt die Vorhersagegenauigkeit in der Regel nicht unter den Wert einer Text-Klassifikation, welcher auf der rechten Seite der Skala markiert ist.

Diese Ergebnisse decken sich mit den Ergebnissen, die Utard in seiner Arbeit von 2005 beschrieben hat und stehen damit im Gegensatz zu den Ergebnissen von Chakrabarti et al. Eine Ausnahme bilden die Testdaten der Datenbank ODP5000: Hier kann ein Rückgang der Vorhersagegenauigkeit gegenüber einer Text-Klassifikation beobachtet werden.

7.2 Verwendung von Links

Hypertext-Seiten zeichnen sich durch eine hohe Anzahl von Links auf Nachbarschaftsseiten aus. Die Informationen, welche aus den Nachbarschaftsseiten gewonnen werden, bilden die Grundlage der Methoden zur Hypertext-Klassifikation. Um den Einfluss der Anzahl von Links auf die Vorhersagegenauigkeit einer Klassifikation zu untersuchen, wurde das folgende Experiment durchgeführt:

Dafür wurde die bereits in dem vorhergehenden Experiment verwendete Link-Local-Methode *Words Around* eingesetzt, allerdings mit dem festen Radius von 10 Wörtern. Von allen vorhandenen Links wurde in diesem Experiment jeweils eine Teilmenge der Links für eine Klassifikation verwendet. Abbildung 7.2 zeigt die Entwicklung der

Vorhersagegenauigkeit einer Klassifikation bei einer schrittweisen Vergrößerung dieser Teilmenge. Um die unterschiedlichen Werte der Datenbanken auf einer Grafik darstellen zu können, befinden sich die Werte der WebKB-Datensammlung auf der linken Achse; alle anderen Werte sind auf der rechten Achse des Diagramms abzulesen:

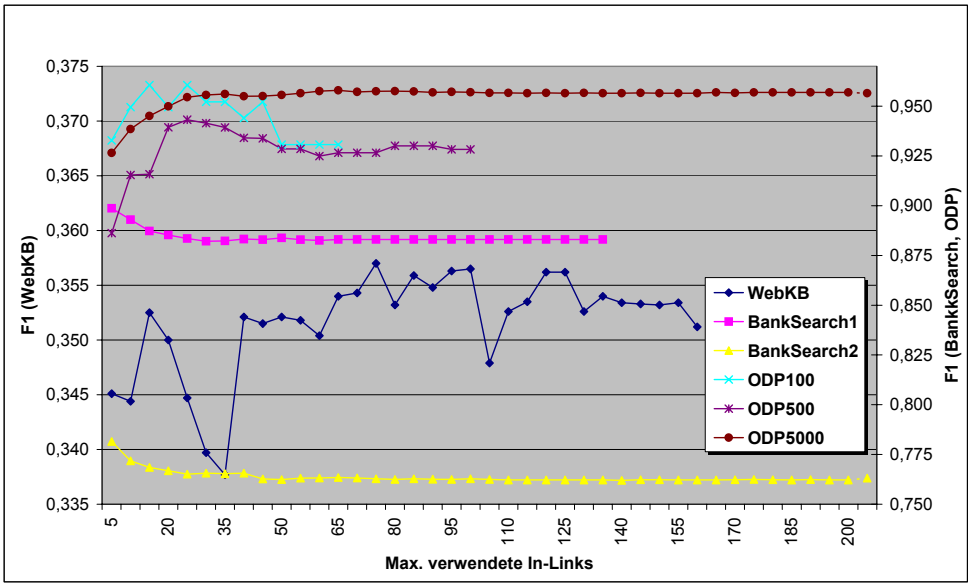

Abbildung 7.2: Vorhersagegenauigkeit bei einer unterschiedlichen Anzahl verwendeter In-Links

Hierbei zeigt sich, dass bereits eine geringe Anzahl In-Links ausreicht, um eine Klassifikation durchzuführen. Schon bei der Verwendung von ca. 50 In-Links finden keine wesentlichen Veränderungen der Vorhersagegenauigkeit mehr statt.

Um die Ursache dieses Effektes zu untersuchen, wurde in dem folgenden Experiment überprüft, wie sich die Feature-Menge bei gleichem Versuchsaufbau verändert (Abbildung 7.3). Auf der linken Achse befindet sich hierbei die Vorhersagegenauigkeit, auf der rechten Achse die Anzahl der Features, die für eine Klassifikation verwendet werden:

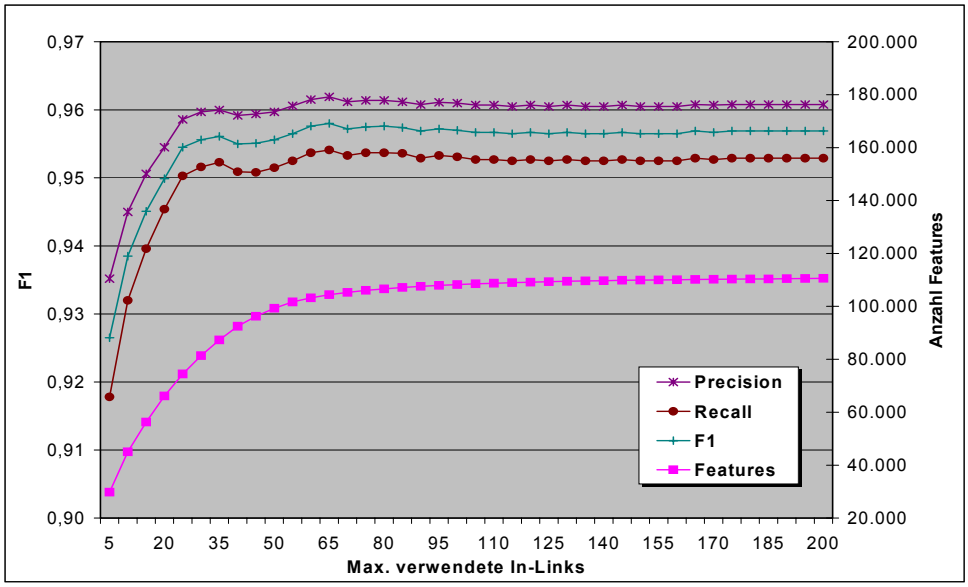

Abbildung 7.3: Features bei einer unterschiedlichen Anzahl verwendeter In-Links (Datenbank: ODP5000)

Es stellt sich heraus, dass nach der Verwendung von ca. 50 In-Links keine wesentliche Vergrößerung der Feature-Menge mehr stattfindet, was wiederum keine Veränderungen der Vorhersagegenauigkeit zur Folge hat.

7.3 Vergleich der Methoden

Die Vorhersagegenauigkeit ist das wichtigste Kriterium für den Vergleich von Methoden. Im kommenden Abschnitt wird die Vorhersagegenauigkeit der verschiedenen Methoden zur Hypertext-Klassifikation anhand des F_1-Wertes verglichen. Dabei wird die in Kapitel 5 vorgestellte Messmethodik eingesetzt, d. h. für einen Vergleich werden die prozentualen Unterschiede des F_1-Wertes zu einer Text-Klassifikation verwendet. Andere Vorhersagegenauigkeits-Werte befinden sich im Anhang A:Tabellen der Messergebnisse.

7.3.1 Texte der Nachbarn

Im Gegensatz zu den Untersuchungen von Chakrabarti et al. wurde in dem folgenden Experiment zwischen der Verwendung von Vorgängerseiten (InText) und Nachfolgerseiten (OutText) differenziert. Die Methode AllText entspricht dem Versuchsaufbau von Chakrabarti et al. Hier werden für eine Klassifizierung die Texte aller Nachbarschaftsdokumente incl. des Own Textes verwendet.

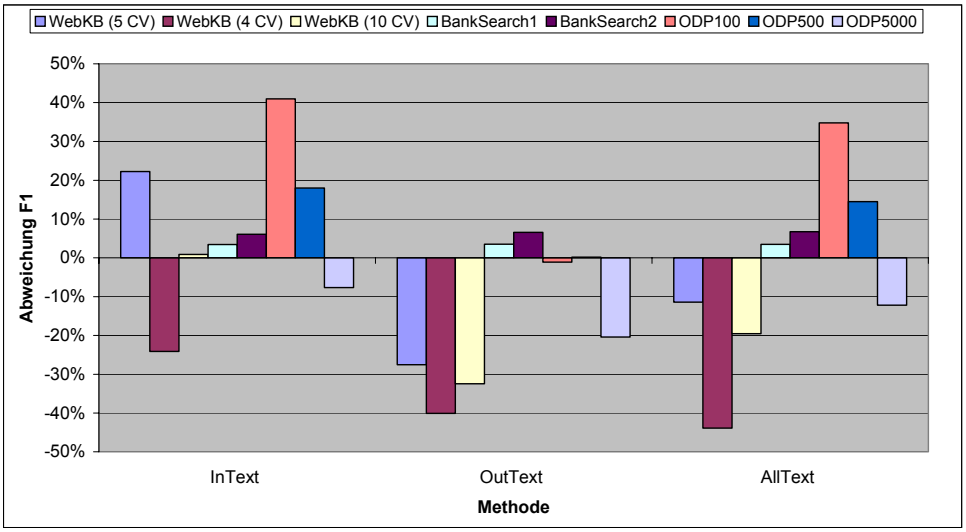

Abbildung 7.4: Vergleich der Methoden (Texte der Nachbarn)

Das Zugrundelegen von Texten der Vorgängerseiten (InText) für eine Klassifikation erbringt meist eine Steigerung der Vorhersagegenauigkeit. Hierbei zeigt sich jedoch auch, dass die Verwendung von Texten der Nachfolgerseiten (OutText) einen negativen Einfluss auf die Vorhersagegenauigkeit einer Klassifikation ausübt. Dies könnte den Rückgang erklären, den Chakrabarti bei seinen Untersuchungen von Texten der Nachbarschaftsseiten beobachtete.

7.3.2 Klassen der Nachbarn

7.3.2.1 Methoden von Chakrabarti et al.

Auch bei diesen Experimenten wurde zwischen der Verwendung von Klasseninformationen aus Vorgängerseiten (InLink) und Nachfolgerseiten (OutLink) differenziert. Der Versuchsaufbau von Chakrabarti et al. wird durch die Methode AllLink nachgebildet. Hierbei wird zwischen der reinen Verwendung von Links bzw. der Kombination von Links und Own-Text unterschieden. Zusätzlich wurden verschiedene Methoden angewendet, um Feature-Gruppen zu kombinieren (vgl. Abschnitt 6.2:Verwendung unterschiedlicher Feature-Gruppen). Diese Methoden sind durch „flat“ bzw. „link“ gekennzeichnet (Abbildung 7.5).

Bei diesen Kombinationen wurde angenommen, dass alle Klassen der Nachbarschaftsseiten bekannt sind (supervised case). Die gemessenen Werte bilden somit eine obere Schranke für die Vorhersagegenauigkeit dieser Methoden, da Chakrabarti et al. bei ihren Messungen einen leichten Rückgang der Vorhersagegenauigkeit beobachteten, falls nur Teile der Klasseninformationen (partially supervised case) bzw. keine Informationen über die Klassen der Nachbarschaftsseiten (unsupervised case) vorliegen (vgl. Abschnitt 3.3.2.4:Berechnung von Klassen der Nachbarschaftsdokumente).

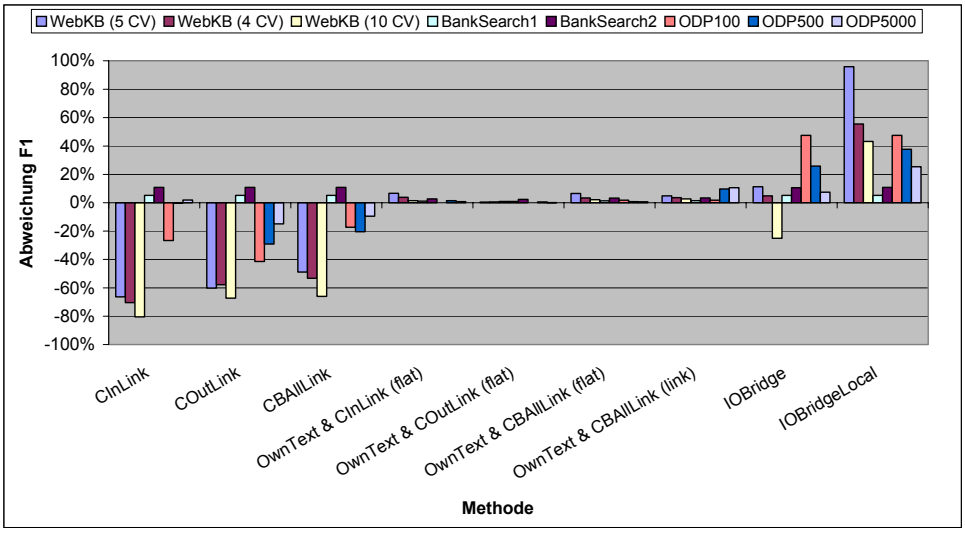

Abbildung 7.5: Vergleich der Methoden (Chakrabarti et al.)

Es zeigt sich, dass eine alleinige Verwendung von Klassen sich meist negativ auf die Vorhersagegenauigkeit auswirkt (CBAllLink). Bei einem Vergleich der Nachfolgerseiten besitzen die Vorgängerseiten (CInLink) eine höhere Vorhersagegenauigkeit als die Nachfolgerseiten (COutLink). Eine Besonderheit bilden die Ergebnisse der BankSearch-Daten. Hier führt eine alleinige Verwendung der Klassen zu einer Erhöhung der Vorhersagegenauigkeit. Diese Ergebnisse lassen auf eine exakte Trennung der Klassen innerhalb der Datenbank schließen.

Eine Kombination von Klasseninformationen und eigenem Text führt hingegen zu einer Steigerung der Vorhersagegenauigkeit, besonders wenn beide Feature-Gruppen im Link-Mode kombiniert werden (Own Text & CBAllLink, link). Die Nutzung von Klasseninformationen aus IO-Bridges und lokalen IO-Bridges bringt eine drastische Erhöhung der Vorhersagegenauigkeit einer Klassifikation. Auf der WebKB-Datenbank konnte hiermit eine beachtliche Steigerung des F1-Wertes von ca. 96% erreicht werden.

7.3.2.2 Methoden von Lu & Getoor

Die getesteten Methoden entsprechen denen, welche auch Lu & Getoor in ihren Untersuchungen verwendeten. Auch bei diesen Methoden wird vorausgesetzt, dass die Klassen der Nachbarschaftsseiten bekannt sind (supervised case). Zusätzlich wurden noch Messungen an den Count-Modellen durchgeführt, bei denen ausschließlich die In- bzw. Out-Links verwendet wurden (Abbildung 7.6):

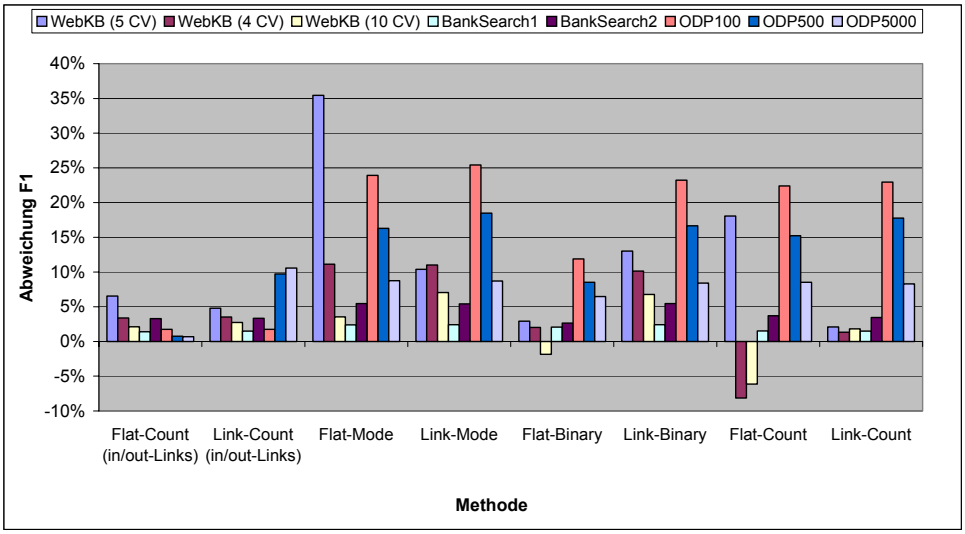

Abbildung 7.6: Vergleich der Methoden (Lu & Getoor)

Hier ist festzustellen, dass Link-Methoden eine generell höhere Vorhersagegenauigkeit als Flat-Methoden aufweisen. Dieses Ergebnis deckt sich mit den Ergebnissen, welche Lu & Getoor in ihren Experimenten beobachteten. Eine eindeutige Unterscheidung der Vorhersagegenauigkeit von verschiedenen Statistik-Modellen ist hier jedoch schwierig zu erkennen. In dem Abschnitt 7.4:Mittelwerte können die Unterschiede der Statistik-Modelle anhand einer Mittelwertbildung besser beobachtet werden.

7.3.3 Link-Local-Methoden

Bei den Link-Local-Methoden wurde, zusätzlich zu den von Utard untersuchten Methoden, auch eine Untersuchung der Seitentitel in verschiedenen Variationen (OwnTitle, InTitle, OutTitle, AllTitle) durchgeführt. Der Titel einer Webseite kann durch den Text, der sich zwischen den HTML-Tags <title> bzw. </title> befindet, extrahiert werden.

Ein Vergleich der Vorhersagegenauigkeit unterschiedlicher Link-Local-Methoden wird in Abbildung 7.7 dargestellt:

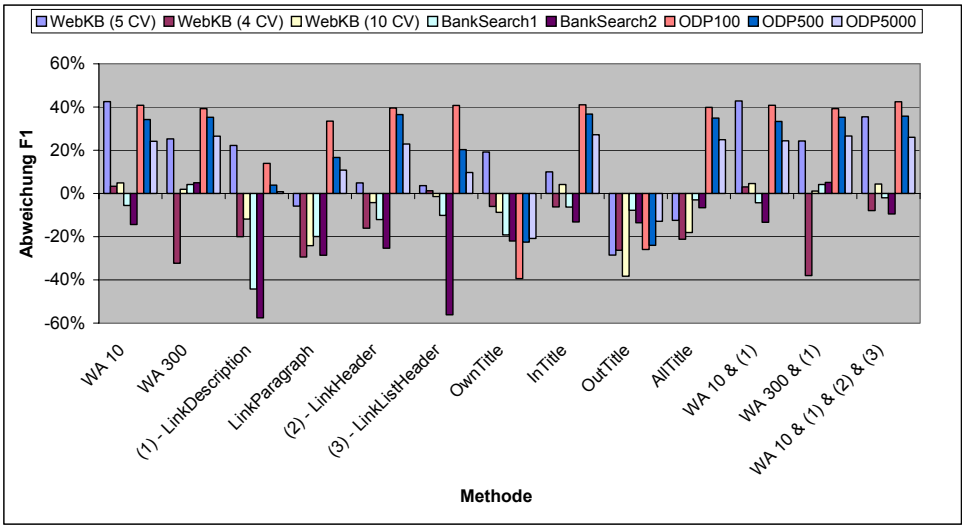

Abbildung 7.7: Vergleich der Methoden (Link Local)

Eine Betrachtung der Ergebnisse zeigt, dass Link-Local-Methoden, je nach den verwendeten Testdaten, unterschiedliche Ergebnisse aufweisen. Die Vorhersagegenauigkeit dieser Methoden ist dabei auf „echten“ Web-Daten (ODP) wesentlich höher als bei anderen Datenbanken. Dieses Ergebnis deckt sich mit den Ergebnissen von Utard, der in der WebKB-Datensammlung eine geringere Vorhersagegenauigkeit, in einer Web-Datenbank jedoch eine höhere Vorhersagegenauigkeit gemessen hat.

Das gute Abschneiden der LinkHeader- bzw. der LinkListHeader-Methoden lässt auf eine hohe „Qualität“ von Seiten mit Link-Listen schließen („*Hubs*“). Diese Methoden nutzen solche Seiten, vergleichbar mit der Methode der lokalen IO-Bridge.

Die Verwendung von Titeln einer Vorgängerseite kann zu einer Verbesserung der Vorhersagegenauigkeit führen, nicht jedoch die Verwendung der Titel von Nachfolgerseiten. Dies entspricht den Ergebnissen, wie sie auch bei der Verwendung von Texten der Nachbarschaftsseiten beobachtet wurden.

7.4 Mittelwerte

Für eine Mittelwertbildung wurden die Testdaten über alle verwendeten Datenbanken im gleichen Verhältnis gemittelt. Um eine bessere Übersicht zu erhalten, wurde aus jeder Gruppe von Methoden die Methode mit der höchsten Vorhersagegenauigkeit ausgewählt (Abbildung 7.8). Die Statistikwerte (Standardabweichung, Standardfehler) der einzelnen Methoden befinden sich in Tabelle 7.1.

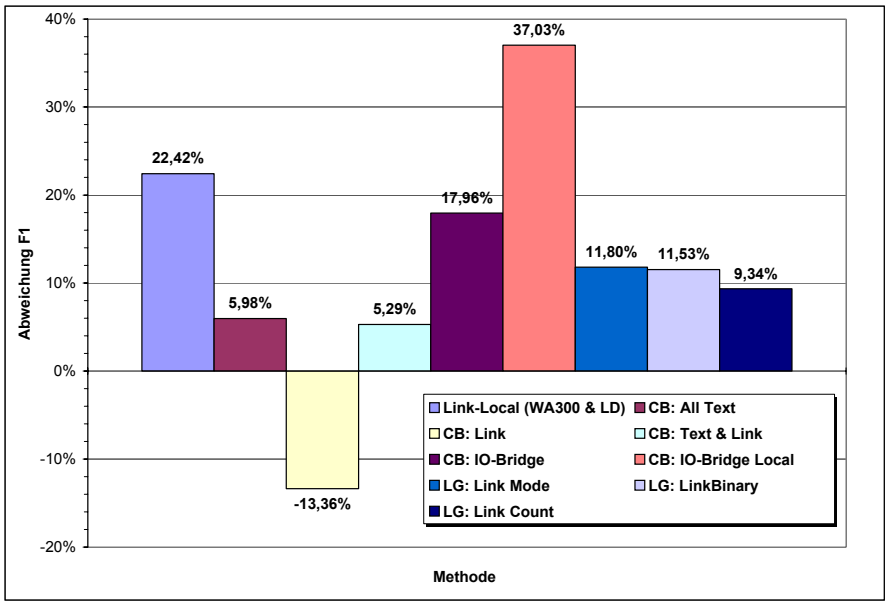

Abbildung 7.8: Vorhersagegenauigkeit (F_1) der Methoden

Methode	Mittel	Sdev	Serr
Link-Local (WA300 & LD)	22,42%	14,85%	6,06%
CB: All Text	5,98%	17,55%	7,17%
CB: Link	-13,36%	21,31%	8,70%
CB: Text & Link	5,29%	3,99%	1,63%
CB: IO-Bridge	17,96%	16,15%	6,59%
CB: IO-Bridge Local	37,03%	32,88%	13,42%
LG: Link Mode	11,80%	8,60%	3,51%
LG: LinkBinary	11,53%	7,68%	3,13%
LG: Link Count	9,34%	9,01%	3,68%
Mittel	12,00%	14,67%	5,99%

Tabelle 7.1: Mittelwerte, Standardabweichungen und -fehler der Vorhersagegenauigkeit (F_1)

Hierbei können die folgenden Ergebnisse festgestellt werden:

Die Verwendung von *Link-Locals* erreicht eine ca. 22% höhere Vorhersagegenauigkeit (Abweichung F_1) als eine Text-Klassifikation.

Die Verwendung *aller Dokumente* der Nachbarschaft erbringt einen Gewinn der Vorhersagegenauigkeit von ca. 6%. Dieses Resultat steht im Gegensatz zu den Ergebnissen von Chakrabarti et al. in anderen Datenbanken.

Eine Klassifikation nur nach den *Klassen der Nachbarschaft* ist nicht erfolgreicher als eine Text-Klassifikation (-13%). Die gemeinsame Verwendung von *Text- und Klasseninformationen* kann einen Gewinn der Vorhersagegenauigkeit von ca. 5% (Chakrabarti) bis zu 12% (Lu & Getoor) einbringen.

Die Verwendung von *IO-Bridges* bzw. *Local IO-Bridges* bringt den höchsten Zugewinn an Vorhersagegenauigkeit (18% bzw. 37%).

7.5 Coverage der Methoden

Neben den Vorhersagegenauigkeits-Werten stellt die Coverage ein wichtiges Kriterium für den Vergleich von Methoden dar. Durch die Coverage wird beschrieben, welche Anteile der Testdokumente sich mit einer Methode klassifizieren lassen. Auch bei diesem Vergleich wird der prozentuale Unterschied zu einer Text-Klassifizierung verwendet (Abbildung 7.9). Die statistischen Angaben werden in Tabelle 7.2 dargestellt.

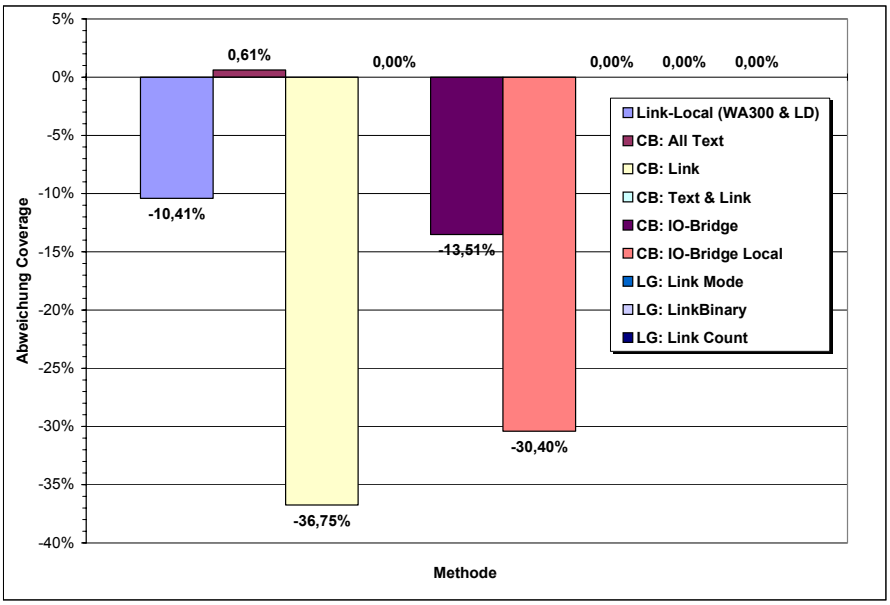

Abbildung 7.9: Coverage der Methoden

Methode	Mittel	Sdev	Serr
Link-Local (WA300 & LD)	-10,41%	5,30%	2,16%
CB: All Text	0,61%	0,68%	0,28%
CB: Link	-36,75%	40,30%	16,45%
CB: Text & Link	0,00%	0,00%	0,00%
CB: IO-Bridge	-13,51%	7,83%	3,20%
CB: IO-Bridge Local	-30,40%	25,16%	10,27%
LG: Link Mode	0,00%	0,00%	0,00%
LG: LinkBinary	0,00%	0,00%	0,00%
LG: Link Count	0,00%	0,00%	0,00%
Mittel	-10,05%	8,81%	3,60%

Tabelle 7.2: Mittelwerte, Standardabweichungen und -fehler der Coverage

Ein Vergleich der Coverage zeigt, dass Link-Local-Methoden einen Rückgang der Coverage von ca. 10% aufweisen. Dies lässt sich dadurch erklären, dass nicht jede zu klassifizierende Seite auch eine Vorgängerseite besitzt. Die Methode AllText hingegen erreicht eine leicht höhere Coverage (+0,6%), da sich mit dieser Methode auch Seiten klassifizieren lassen, welche keinen eigenen (verwertbaren) Text aufweisen.

Der drastische Rückgang bei der alleinigen Verwendung von Links (Klassen der Nachbarschaftsseiten) lässt sich auf die hohe Anzahl von Nachbarschaftsseiten zurückführen, deren Klassen nicht festgestellt werden konnten. Methoden, die eine Kombination von Features durch die Verwendung von Own Text nutzen, haben die Coverage einer Text-Klassifikation. IO-Bridges bzw. lokale IO-Bridges haben eine deutlich geringere Coverage, da nicht alle Seiten auch IO-Bridges ausbilden.

7.6 Messungen bei gleicher Coverage

Bei den vorhergehenden Untersuchungen wurden alle Methoden mit einem einzigen Referenzwert (Own Text) verglichen. Bei dieser Untersuchung jedoch wird für den Referenzwert einer Methode der Wert mit der gleichen Coverage verwendet, d. h. dass bei jeder Methode diejenigen Dokumente den Referenzwert (Own Text) bilden, welche auch mit dieser Methode klassifiziert werden können (Abbildung 7.10). Die Mittelwerte, Standardabweichungen und -fehler dieser Messungen werden in Tabelle 7.3 dargestellt.

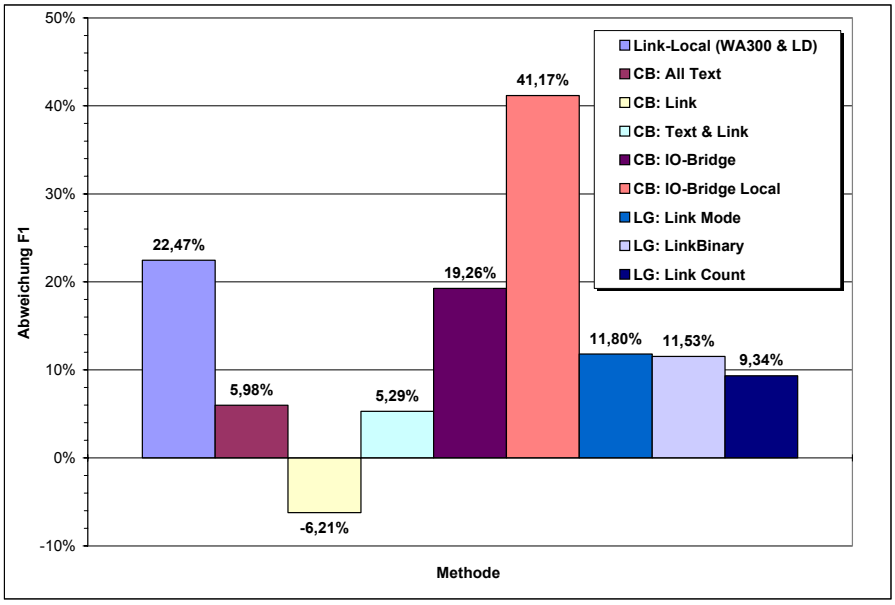

Abbildung 7.10: Vorhersagegenauigkeit (F_1) bei gleicher Coverage

Methode	Mittel	Sdev	Serr
Link-Local (WA300 & LD)	22,47%	16,71%	6,82%
CB: All Text	5,98%	17,55%	7,17%
CB: Link	-6,21%	23,28%	9,51%
CB: Text & Link	5,29%	3,99%	1,63%
CB: IO-Bridge	19,26%	23,01%	9,39%
CB: IO-Bridge Local	41,17%	37,61%	15,35%
LG: Link Mode	11,80%	8,60%	3,51%
LG: LinkBinary	11,53%	7,68%	3,13%
LG: Link Count	9,34%	9,01%	3,68%
Mittel	13,40%	16,38%	6,69%

Tabelle 7.3: Mittelwerte, Standardabweichungen und -fehler bei gleicher Coverage

Es ist erkennbar, dass sich an dem „Ranking“ der Methoden nichts verändert hat; die Methoden mit den höchsten Werten konnten ihren Vorsprung sogar noch ausbauen (vgl. Abbildung 7.8).

Als Erklärung wird von der Annahme ausgegangen, dass die Wahrscheinlichkeit für die Coverage einer Methode in diesem Fall unabhängig vom eigenen Text ist. So ist z. B. die Wahrscheinlichkeit, dass ein Dokument eine IO-Bridge ausbildet, nicht vom Own-Text abhängig (eher von der „inhaltlichen Qualität“ des Own-Textes).

Wenn man jetzt den Referenzwert (den Own-Text) nur mit den gleichen Dokumenten der Methode (IO-Bridge) berechnet, erreicht man lediglich, dass der Referenzwert mit weniger Datensätzen auskommen muss. Da aber die Qualität des Referenzwertes mit fallender Anzahl der Datensätze abnimmt, ist der prozentuale Gewinn der Methode (IO-Bridge) nun höher. Dieser Vorsprung ist umso größer, je geringer die (ursprüngliche) Coverage der Methode ist. Dieses Ergebnis lässt sich in Abbildung 7.10 gut beobachten. Die statistischen Angaben hierfür befinden sich in Tabelle 7.3.

7.7 Kombinationen von Methoden

Durch die Kombination von Features kann, wie am Beispiel der Link-Local-Methode (Link Description & Words Around) aufgezeigt, eine Steigerung der Vorhersagegenauigkeit erreicht werden. Es ist zu vermuten, dass durch eine Kombination der bisher vorgestellten *Methoden* eine nochmalige Steigerung der Vorhersagegenauigkeits-Werte erzielt werden könnte.

Durch ein Experiment wurde diese Vermutung bestätigt (Tabelle 7.4): Hierbei wurden zwei erfolgreiche Methoden im Flat-Mode bzw. im Link-Mode kombiniert, vgl. Abschnitt 3.3.2.3:Methoden von Lu & Getoor (2003)

Methode	F1-Wert
OwnText	0,7744
(1) Link-Local (WA300 & LD)	0,9802
(2) CB: IO-Bridge Local	**0,9707**
Kombination von (1) und (2) **Flat**	0,9873
Kombination von (1) und (2) **Link**	**0,9884**
Flat = Sammlung aller Features einer Methode in einem gemeinsamen Meta-Dokument (merging) **Link** = Getrennte Betrachtung gleichartiger Features, durch die Verwendung von 2 getrennten Klassifizierern	

Tabelle 7.4: Vorhersagegenauigkeit (F_1) der Methoden sowie deren Kombination (Datenbank: ODP5000)

Es wurde festgestellt, dass durch die Kombination zweier erfolgreicher Methoden eine nochmalige Steigerung der Vorhersagegenauigkeit (ca. 1% F_1) in der Datenbank ODP5000 erreicht werden kann. Ob diese Ergebnisse aber übertragbar sind, bleibt fraglich, da Messungen in anderen Datenbanken keine eindeutigen Ergebnisse erbrachten.

7.8 Universeller Klassifikator

Methoden zur Hypertext-Klassifikation bieten, wie in dieser Arbeit aufgezeigt wurde, vielfältige Möglichkeiten. Hierbei entsteht jedoch das Problem, dass bei gegebenen Testdaten nicht alle Methoden zum Einsatz kommen können (Coverage).

Um nun einen Klassifikator zu konstruieren, welcher die individuellen Vorteile der verschiedenen Methoden mit einer Coverage aller Testdokumente verbindet (universeller Klassifikator), wurde das folgende Vorgehen gewählt:

Aus jeder der Gruppen Texte der Nachbarn, Klassen der Nachbarn und Link-Local wurde eine Methode ausgewählt. Beginnend mit einer Methode, die sich in den vorangegangenen Experimenten als die aussichtsreichste erwiesen hat, werden alle Datensätze klassifiziert. Falls diese Methode einen Datensatz nicht verarbeiten kann, wird zu der „nächst schlechteren“ Methode übergegangen.

Dieser Vorgang wird so lange wiederholt, bis alle Datensätze, wiederum beginnend bei der stärksten Methode, klassifiziert worden sind. Durch dieses Verfahren wird eine vollständige Coverage erreicht, da im „worst case“ eine Methode verwendet wird, die eine vollständige Coverage aufweist (Texte der Nachbarn).

Für eine Überprüfung dieses Verfahrens wurden die folgenden Methoden ausgewählt:

- Aus der Gruppe der Methoden, welche die Klassen der Nachbarn verwenden, wurde die Methode *Lokale IO-Bridges* ausgewählt, die aber eine geringe Coverage aufweist.
- Link-Locals, welche einen Kompromiss aus Vorhersagegenauigkeit und Coverage darstellen, liefern die zweite Methode. Hier wurde die Kombination von *Link-Description & Words Around 300* (merged) ausgewählt.
- Aus der Gruppe der Texte der Nachbarn wurde *AllText* ausgewählt, welche zwar eine geringere Vorhersagegenauigkeit, jedoch eine hohe Coverage aufweist.

Eine Überprüfung in den Test-Datenbanken lieferte die folgenden Ergebnisse, wobei als Vergleich die Werte einer Text-Klassifikation (Own Text) verwendet wurden (Tabelle 7.5):

		Messwerte			
DB	**Methode**	**Accuracy**	**Precision**	**Recall**	**F1**
WebKB	OwnText	0,7813	0,3166	0,3340	0,3250
	Universeller Klassifizierer	0,8637	0,5080	0,4532	0,4790
	Abweichung zu (Own Text)	**10,55%**	**60,45%**	**35,69%**	**47,38%**
BankSearch1	OwnText	0,9754	0,9548	0,9473	0,9510
	Universeller Klassifizierer	0,9961	0,9933	0,9930	0,9931
	Abweichung zu (Own Text)	**2,12%**	**4,03%**	**4,82%**	**4,43%**
BankSearch2	OwnText	0,9813	0,9034	0,8925	0,8979
	Universeller Klassifizierer	0,9933	0,9653	0,9631	0,9642
	Abweichung zu (Own Text)	**1,22%**	**6,85%**	**7,91%**	**7,38%**
ODP100	OwnText	0,8681	0,6801	0,6766	0,6783
	Universeller Klassifizierer	0,9832	0,9587	0,9600	0,9594
	Abweichung zu (Own Text)	**13,26%**	**40,96%**	**41,89%**	**41,44%**
ODP500	OwnText	0,8880	0,7400	0,7132	0,7264
	Universeller Klassifizierer	0,9942	0,9865	0,9853	0,9859
	Abweichung zu (Own Text)	**11,96%**	**33,31%**	**38,15%**	**35,72%**
ODP5000	OwnText	0,9081	0,7788	0,7701	0,7744
	Universeller Klassifizierer	0,9842	0,9608	0,9605	0,9606
	Abweichung zu (Own Text)	**8,38%**	**23,37%**	**24,72%**	**24,04%**
Mittelwerte		**7,92%**	**28,16%**	**25,53%**	**26,73%**
Standardabweichung		5,11%	21,41%	15,94%	17,90%
Standardfehler (sdev/sqr(N))		2,08%	8,74%	6,51%	7,31%

Tabelle 7.5: Ergebnisse des universellen Klassifikators

Der Durchschnittswert aus allen sechs verwendeten Datenbanken zeigt, dass dieser Ansatz eine allgemein höhere Vorhersagegenauigkeit aufweist als der Ansatz einer Text-Klassifikation (z. B. +27% beim F_1-Wert).

Bei der Betrachtung der Ergebnisse der einzelnen Datenbanken fällt auf, dass diese maßgeblich von den Ergebnissen der Methode *Lokale IO-Bridges* bestimmt werden. So spiegelt sich z. B. die Abnahme der Vorhersagegenauigkeit bei zunehmender Größe der Datenbanken (ODP100, ODP500, ODP5000) auch in den entsprechenden Ergebnissen der Methode *Lokale IO-Bridges* wider (vgl. Tabelle A.9 im Anhang).

Bei der WebKB-Datenbank ist es auch die Link-Local-Methode *Link-Description & Words Around 300*, welche für die hohe Vorhersagegenauigkeit verantwortlich ist, da bei dieser Datenbank die Methode *Lokale IO-Bridges* nur eine geringe Coverage aufweist (vgl. Tabelle A.15 bzw. Tabelle A.7 im Anhang).

8 Zusammenfassung und Ausblick

In dieser Arbeit wurden verschiedene Methoden zur Hypertext-Klassifikation vorgestellt und anhand einer definierten Messmethodik miteinander verglichen. Die vorgestellten Methoden können in Gruppen zusammengefasst werden, je nachdem, welche Informationen aus den Nachbarschaften (verlinkte Dokumente) verwendet werden: Texte, Klassen oder Texte, die im Umfeld der Links auftreten (Link-Locals).

Die für einen Vergleich notwendigen Datenbanken wurden aus verschiedenen Datenquellen extrahiert und in eine standardisierte Form gebracht. Statistiken der In- bzw. der Out-Links geben einen Einblick in die Verlinkungsstruktur der Datenbanken.

In den Experimenten wurden die vorgestellten Methoden unter gleichen Bedingungen verglichen; die Vergleiche liefern als Ergebnis die prozentualen Abweichungen gegenüber einer konventionellen Text-Klassifikation. Die durchgeführten Experimente erzielten in den hierfür zusammengestellten Datenbanken im Einzelnen die folgenden Ergebnisse:

- Die Verwendung *aller Dokumente* der Nachbarschaft erbringt einen Gewinn der Vorhersagegenauigkeit von ca. 6% gegenüber einer Text-Klassifikation, wobei die Vorhersagegenauigkeit stark von den verwendeten Testdaten abhängt. Dieses Ergebnis steht im Gegensatz zu den Ergebnissen von Chakrabarti et al. in anderen Datenbanken.
- Eine Klassifikation nur nach den *Klassen der Nachbarschaft* ist nicht erfolgreicher als eine Text-Klassifikation (-13%).
- Die gemeinsame Verwendung von *Text- und Klasseninformationen* kann einen Gewinn der Vorhersagegenauigkeit von ca. 5% (Chakrabarti) bis zu 12% (Lu & Getoor) bei gleicher Coverage erzielen.
- Die Verwendung von *Link-Locals* erreicht eine um ca. 22% höhere Vorhersagegenauigkeit (Abweichung F_1) als eine Text-Klassifikation; allerdings muss dieser Vorteil durch eine geringere Coverage (-10%) erkauft werden.
- Die Verwendung von *IO-Bridges* bzw. *Local IO-Bridges* bringt den höchsten Zugewinn an Vorhersagegenauigkeit (18% bzw. 37%), jedoch auch nur mit einem deutlichen Rückgang der Coverage (-13% bzw. -30%).

Bei den Untersuchungen wurde festgestellt, dass die Anzahl der Trainingsdokumente bzw. die Anzahl der verwendeten Dokumente der Nachbarschaft keinen wesentlichen Einfluss auf die Auswahl einer geeigneten Methode hat.

Es bleibt zu prüfen, ob eine systematische Kombination dieser Methoden zu einer nochmaligen Steigerung der Vorhersagegenauigkeit führen könnte. Versuche hierzu sind viel versprechend, wenngleich auch nicht bei allen verwendeten Datenbanken reproduzierbar.

Für ein abschließendes Ergebnis wurde ein universeller Klassifikator implementiert, der die individuellen Vorteile der Methoden mit einer vollständigen Coverage vereint. Die F_1-Vorhersagegenauigkeit dieses universellen Klassifikators liegt, bei einer Evaluierung mit allen erstellten Datenbanken, im Mittel um 27% über den Werten einer Text-Klassifikation.

Die Auswahl bzw. die Reihenfolge der Methoden, die dieser universelle Klassifikator verwendet, könnte Gegenstand weiterer interessanter Untersuchungen sein.

Literaturverzeichnis

Borges, M. & Levene, M. (1999). Data mining of user navigation patterns. In *Proceedings of the WBBKDD'99 Workshop on Web Usage Analysis and User Profiling*. pp. 31-36. Retrieved 01.03.2007, from http://www.cse.yorku.ca/course_archive/2001-02/W/6490C/papers/borges.ps

Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks, 30(1–7). pp. 107–117. In *Proceedings of the 7th International World Wide Web Conference (WWW-7)*, Brisbane, Australia.

Chakrabarti, S. et al. (1998). Enhanced hypertext categorization using hyperlinks. In *Proceedings SIGMOD-98*. pp. 307-318.

Chakrabarti, S. (2000). Data mining for hypertext: A tutorial survey. In *SIGKDD Explorations* 1(2): pp. 1-11.

Chakrabarti, S. (2002). Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann. pp. 130-131.

Chakrabarti S., Roy, S. & Soundalgekar, M. (2003). Fast and accurate text classification via multiple linear discriminant projections. In *VLDB Journal*, 12(2). pp. 170-185.

Conklin, J. (1987). Hypertext: An introduction and survey. In *IEEE Computer*, 20(9). pp. 17-41.

Craven, M. et al. (2000). Learning to construct knowledge bases from the World Wide Web. In *Artificial Intelligence* 118(1). pp. 69–113. Retrieved 01.03.2007, from http://www.cs.cmu.edu/~knigam/papers/webkb-aij00.pdf

Eclipse Foundation (2007). Eclipse. Projekt Homepage. Retrieved 01.03.2007, from http://www.eclipse.org

Fürnkranz, J. (2002). Hyperlink Ensembles: A Case Study in Hypertext Classification. In *Information Fusion* 3(4): pp. 299-312, December 2002. Special Issue on Fusion of Multiple Classifiers.

Kan, M.-Y. (2005). Natural Language Processing / Information Retrieval Software Repository. School of Computing. National University of Singapore. Retrieved 01.03.2007, from: http://www.comp.nus.edu.sg/~rpnlpir/

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Articial Intelligence*. pp. 37-43. Retrieved 01.03.2007, from http://robotics.stanford.edu/~ronnyk/accEst.ps

Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. p. 202.

Lohninger, H. (2006). Grundlagen der Statistik. Online-Book. Retrieved 01.03.2007, from http://www.statistics4u.com/fundstat_germ/

Lu, Q. & Getoor, L. (2003). Link-based Classification. In *Proceedings of the International Conference on Machine Learning*, Washington, DC, August 2003.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. In *IBM Journal of Research and Development*. April 1958. pp.159-165. Retrieved 01.03.2007, from http://www.research.ibm.com/journal/rd/022/luhn.pdf

Madria, K. et al. (1999). Research issues in web data mining. In *Proceedings of Data Warehousing and Knowledge Discovery. First International Conference. DaWaK '99*. pp. 303-312. Retrieved 01.03.2007, from http://www.dmresearch.net/download/paper/spider/Research%20issues%20in%20web%20data%20mining.pdf

McCallum, A. (2003). Andrew McCallums Code and Data. Department of Computer Science, University of Massachusetts. Retrieved 01.03.2007, from http://www.cs.umass.edu/~mccallum/code-data.html

Murray, B. & Moore, A. (2000). Sizing the internet. White paper, Cyveillance. Retrieved 01.03.2007, from http://www.cyveillance.com/web/downloads/Sizing_the_Internet.pdf

Rijsbergen van, C. J. (1979). Information Retrieval. London; Boston. Butterworth, 2nd Edition. Retrieved 01.03.2007, from http://www.dcs.gla.ac.uk/Keith/Preface.html

Sinka, M. & Corne, D. (2002). A Large Benchmark Dataset for Web Document Clustering. In *Soft Computing Systems: Design, Management and Applications*, Vol. 87 of Frontiers in Artificial Intelligence and Applications. pp. 881–890.

Sun Microsystems (2007). Java & Technologies. Homepage. Retrieved 01.03.2007, from http://www.sun.com/download

Utard, H. (2005). Hypertext Classification. Diplomarbeit. FG Knowledge Engineering, TU Darmstadt.

Utard, H. & Fürnkranz, J. (2006). Link-local features for hypertext classification. In Berendt, B., Grobelnik, M., Hotho, A., Mladenic, D., Semeraro, G., van Someren, M., Spiliopoulou, M., Stumme, G., and Svatek, V., editors, *Semantics, Web and Mining*. Springer-Verlag.

WebKB (1997). The 4 Universities Data Set. Projekt Homepage. School of Computer Science. Carnegie Mellon University. Retrieved 01.03.2007, from: http://www.cs.cmu.edu/afs/cs.cmu.edu/projekt/theo20/www/data

W3C (1999). XML Path Language (XPath) Version 1.0. W3C Recommendation. XML Linking Working Group. Retrieved 01.03.2007, from http://www.w3.org/TR/xpath

W3C (2000). XHTML 1.0 The Extensible HyperText Markup Language. W3C Recommendation. Retrieved 01.03.2007, from http://www.w3.org/TR/xhtml1/

W3C (2004). Resource Description Framework (RDF). RDF Core Working Group. Retrieved 01.03.2007, from http://www.w3.org/RDF/

Yang, Y. & Pedersen J.P. (1997). A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*. pp. 412-420.

Yang, Y. & Liu, X. (1999). A Re-examination of Text Categorization Methods. In *Proceedings of ACM SIGIR'99 conference*. pp. 42-49.

Anhang A: Tabellen der Messergebnisse

In diesem Anhang befinden sich die detaillierten Messergebnisse der Methoden, welche im Abschnitt 3.3:Methoden zur Hypertext-Klassifikation vorgestellt wurden:

- Texte der Nachbarn (vgl. Abschnitt 3.3.1)
- Klassen der Nachbarn
  - Methoden von Chakrabarti et al. (vgl. Abschnitt 3.3.2.2)
  - Methoden von Lu & Getoor (vgl. Abschnitt 3.3.2.3)
- Link-Local-Methoden (vgl. Abschnitt 3.3.3)

Jede Tabelle beinhaltet die Messwerte der Experimente (Abbildung A.1) sowie die jeweiligen Abweichungen zu einem Basiswert (Abbildung A.2). Tabelle A.1 beschreibt die Datenfelder der Messergebnisse.

A) Messwerte:

Messwerte										
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1

Abbildung A.1 Kopfzeile der Messwerte

Feldname	Beschreibung
DB	Die verwendete Datenbank (sowie die Art der CV)
Methode	Getestete Methode zur Klassifizierung
Docs	Anzahl der gesamten Dokumente
NoCoverage	Anzahl Dokumente, die nicht klassifiziert werden können
Coverage	Anzahl Dokumente, die klassifiziert werden können
Features	Anzahl der vorhandenen Features
Accuracy	Performance-Wert (Macro avr.)
Error	Performance-Wert (Macro avr.)
Precision	Performance-Wert (Macro avr.)
Recall	Performance-Wert (Macro avr.)
F1	Performance-Wert (Macro avr.)

Tabelle A.1 Datenfelder der Messergebnisse

B) Prozentuale Abweichungen:

Abweichungen in % (zu Own Text)					
Coverage	Features	Accuracy	Precision	Recall	F1

Abbildung A.2 Kopfzeile der Abweichungen

Hier werden die prozentualen Abweichungen zur Methode: „Own-Text" (Basiswert) angegeben. Die Werte werden grün dargestellt, falls der Wert über dem Basiswert liegt; ansonsten erfolgt eine Darstellung in Rot. Die Methode mit der höchsten Vorhersagegenauigkeit hinsichtlich ihres F_1-Wertes einer Datenbank wird durch **Fettdruck** hervorgehoben.

A.1 Texte der Nachbarn

In Tabelle A.2 findet sich eine Beschreibung der getesteten Methoden. Die Messwerte befinden sich in Tabelle A.3, Tabelle A.4, Tabelle A.5.

Methode	Beschreibung
OwnText	Eigener Text (Basiswert)
InText	Text aller In-Links
OutText	Text aller Out-Links
AllText	Gesamter Text (Own-Text & Text aller Links)

Tabelle A.2: Getestete Methoden mit Texten der Nachbarn

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
WebKB (5 CV)	OwnText	6.004	0	6.004	36.624	0,7813	0,2187	0,3166	0,3340	0,3250						
	InText	**6.004**	**1.023**	**4.981**	**41.110**	**0,8109**	**0,1891**	**0,4052**	**0,3897**	**0,3973**	**-17,04%**	**12,25%**	**3,79%**	**27,98%**	**16,68%**	**22,25%**
	OutText	6.004	2.422	3.582	44.619	0,7629	0,2371	0,2207	0,2525	0,2355	-40,34%	21,83%	-2,36%	-30,29%	-24,40%	-27,54%
	AllText	6.004	0	6.004	75.923	0,7726	0,2274	0,2733	0,3041	0,2879	0,00%	107,30%	-1,11%	-13,68%	-8,95%	-11,42%
WebKB (4 CV)	OwnText	6.004	0	6.004	36.624	0,8127	0,1873	0,3089	0,5913	0,4058						
	InText	**6.004**	**1.023**	**4.981**	**41.110**	**0,7995**	**0,2005**	**0,2553**	**0,3879**	**0,3079**	**-17,04%**	**12,25%**	**-1,62%**	**-17,35%**	**-34,40%**	**-24,13%**
	OutText	6.004	2.422	3.582	44.619	0,7691	0,2309	0,2074	0,2937	0,2431	-40,34%	21,83%	-5,36%	-32,86%	-50,33%	-40,09%
	AllText	6.004	0	6.004	75.923	0,7708	0,2292	0,1961	0,2719	0,2278	0,00%	107,30%	-5,16%	-36,52%	-54,02%	-43,86%
WebKB (10 CV)	OwnText	6.004	0	6.004	36.624	0,8596	0,1404	0,4745	0,6782	0,5583						
	InText	**6.004**	**1.023**	**4.981**	**41.110**	**0,8816**	**0,1184**	**0,5294**	**0,6019**	**0,5633**	**-17,04%**	**12,25%**	**2,56%**	**11,57%**	**-11,25%**	**0,90%**
	OutText	6.004	2.422	3.582	44.619	0,8274	0,1726	0,3572	0,3993	0,3770	-40,34%	21,83%	-3,75%	-24,72%	-41,12%	-32,47%
	AllText	6.004	0	6.004	75.923	0,8389	0,1611	0,4058	0,5027	0,4491	0,00%	107,30%	-2,41%	-14,48%	-25,88%	-19,56%

Tabelle A.3: Ergebnisse mit Texten der Nachbarn (Datenbanken WebKB)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
BankSearch1 (10 CV)	OwnText	2.176	0	2.176	43.690	0,9754	0,0246	0,9548	0,9473	0,9510						
	InText	2.176	265	1.911	59.913	0,9908	0,0092	0,9848	0,9824	0,9836	-12,18%	37,13%	1,58%	3,14%	3,71%	3,43%
	OutText	**2.176**	**290**	**1.886**	**58.525**	**0,9920**	**0,0080**	**0,9859**	**0,9831**	**0,9845**	**-13,33%**	**33,96%**	**1,70%**	**3,26%**	**3,78%**	**3,52%**
	AllText	2.176	0	2.176	76.924	0,9913	0,0087	0,9851	0,9831	0,9841	0,00%	76,07%	1,63%	3,17%	3,78%	3,48%
BankSearch2 (10 CV)	OwnText	6.076	0	6.076	90.736	0,9813	0,0187	0,9034	0,8925	0,8979						
	InText	6.076	951	5.125	127.955	0,9910	0,0090	0,9526	0,9518	0,9522	-15,65%	41,02%	0,99%	5,45%	6,64%	6,05%
	OutText	6.076	846	5.230	121.800	0,9917	0,0083	0,9579	0,9558	0,9568	-13,92%	34,24%	1,06%	6,03%	7,09%	6,56%
	AllText	**6.076**	**0**	**6.076**	**168.061**	**0,9922**	**0,0078**	**0,9592**	**0,9574**	**0,9583**	**0,00%**	**85,22%**	**1,11%**	**6,18%**	**7,27%**	**6,73%**

Tabelle A.4: Ergebnisse mit Texten der Nachbarn (Datenbanken BankSearch)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
ODP100 (10 CV)	OwnText	95	1	94	3.791	0,8681	0,1319	0,6801	0,6766	0,6783						
	InText	**95**	**8**	**87**	**70.312**	**0,9816**	**0,0184**	**0,9524**	**0,9600**	**0,9562**	**-7,45%**	**1754,71%**	**13,07%**	**40,04%**	**41,89%**	**40,97%**
	OutText	95	49	46	28.429	0,8522	0,1478	0,7222	0,6261	0,6707	-51,06%	649,91%	-1,83%	6,19%	-7,46%	-1,12%
	AllText	95	0	95	84.343	0,9663	0,0337	0,9183	0,9100	0,9141	1,06%	2124,82%	11,31%	35,02%	34,50%	34,76%
ODP500 (10 CV)	OwnText	483	8	475	12.857	0,8880	0,1120	0,7400	0,7132	0,7264						
	InText	**483**	**33**	**450**	**233.817**	**0,9342**	**0,0658**	**0,8831**	**0,8325**	**0,8570**	**-5,26%**	**1718,60%**	**5,20%**	**19,34%**	**16,73%**	**17,98%**
	OutText	483	254	229	78.386	0,8760	0,1240	0,7851	0,6780	0,7276	-51,79%	509,68%	-1,35%	6,09%	-4,94%	0,17%
	AllText	483	0	483	272.497	0,9195	0,0805	0,8712	0,7957	0,8317	1,68%	2019,44%	3,55%	17,73%	11,57%	14,50%
ODP5000 (10 CV)	OwnText	5.257	62	5.195	71.268	0,9081	0,0919	0,7788	0,7701	0,7744						
	InText	**5.257**	**316**	**4.941**	**1.128.644**	**0,8480**	**0,1520**	**0,8402**	**0,6221**	**0,7149**	**-4,89%**	**1483,66%**	**-6,62%**	**7,88%**	**-19,22%**	**-7,68%**
	OutText	5.257	2.765	2.492	497.288	0,8268	0,1732	0,6941	0,5544	0,6164	-52,03%	597,77%	-8,95%	-10,88%	-28,01%	-20,40%
	AllText	5.257	0	5.257	1.445.780	0,8305	0,1695	0,8270	0,5773	0,6800	1,19%	1928,65%	-8,55%	6,19%	-25,04%	-12,19%

Tabelle A.5: Ergebnisse mit Texten der Nachbarn (Datenbanken ODP)

A.2 Methoden von Chakrabarti et al.

In Tabelle A.6 findet sich eine Beschreibung der getesteten Methoden. Die Messwerte befinden sich in Tabelle A.7, Tabelle A.8, Tabelle A.9.

Methode	Beschreibung
OwnText	Eigener Text (Basiswert)
CInLink	Klassen der In-Links
COutLink	Klassen der Out-Links
CBAllLink	Klassen aller Links
OwnText & CInLink (flat)	Own-Text & Klassen der In-Links **(Flat)**
OwnText & COutLink (flat)	Own-Text & Klassen der Out-Links **(Flat)**
OwnText & CBAllLink (flat)	Own Text & Klassen aller Links **(Flat)**
OwnText & CBAllLink (link)	Own-Text & Klassen aller Links **(Link)**
IOBridge	IO-Bridges
IOBridgeLocal	Lokale IO-Bridges
Flat = Sammlung aller Features einer Methode in einem gemeinsamen Meta-Dokument (merging)	
Link = Getrennte Betrachtung gleichartiger Features durch die Verwendung von 2 getrennten Klassifizierern	

Tabelle A.6: Getestete Methoden von Chakrabarti et al.

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
WebKB (5 CV)	OwnText	6.004	0	6.004	36.624	0,7813	0,2187	0,3166	0,3340	0,3250						
	CInLink	6.004	1.024	4.980	7	0,8910	0,1090	0,0884	0,1429	0,1092	-17,06%	-99,98%	14,04%	-72,08%	-57,22%	-66,40%
	COutLink	6.004	2.423	3.581	7	0,8045	0,1955	0,1163	0,1466	0,1297	-40,36%	-99,98%	2,97%	-63,27%	-56,11%	-60,09%
	CBAllLink	6.004	1	6.003	7	0,8681	0,1319	0,1788	0,1551	0,1661	-0,02%	-99,98%	11,11%	-43,52%	-53,56%	-48,89%
	OwnText & CInLink (flat)	6.004	0	6.004	36.631	0,7874	0,2126	0,3393	0,3537	0,3464	0,00%	0,02%	0,78%	7,17%	5,90%	6,58%
	OwnText & COutLink (flat)	6.004	0	6.004	36.631	0,7833	0,2167	0,3173	0,3362	0,3265	0,00%	0,02%	0,26%	0,22%	0,66%	0,46%
	OwnText & CBAllLink (flat)	6.004	0	6.004	36.631	0,7890	0,2110	0,3358	0,3575	0,3463	0,00%	0,02%	0,99%	6,06%	7,04%	6,55%
	OwnText & CBAllLink (link)	6.004	0	6.004	36.624	0,7973	0,2027	0,3272	0,3552	0,3406	0,00%	0,00%	2,05%	3,35%	6,35%	4,80%
	IOBridge	6.004	1.473	4.531	7	0,8953	0,1047	0,4375	0,3083	0,3617	-24,53%	-99,98%	14,59%	38,19%	-7,69%	11,29%
	IOBridgeLocal	**6.004**	**3.866**	**2.138**	**7**	**0,9763**	**0,0237**	**0,6449**	**0,6281**	**0,6364**	**-64,39%**	**-99,98%**	**24,96%**	**103,70%**	**88,05%**	**95,82%**
WebKB (4 CV)	OwnText	6.004	0	6.004	36.624	0,8127	0,1873	0,3089	0,5913	0,4058						
	CInLink	6.004	1.024	4.980	7	0,9220	0,0780	0,1039	0,1429	0,1203	-17,06%	-99,98%	13,45%	-66,36%	-75,83%	-70,35%
	COutLink	6.004	2.423	3.581	7	0,8487	0,1513	0,1600	0,1849	0,1716	-40,36%	-99,98%	4,43%	-48,20%	-68,73%	-57,71%
	CBAllLink	6.004	1	6.003	7	0,9257	0,0743	0,2099	0,1728	0,1895	-0,02%	-99,98%	13,90%	-32,05%	-70,78%	-53,30%
	OwnText & CInLink (flat)	6.004	0	6.004	36.631	0,8172	0,1828	0,3236	0,6024	0,4211	0,00%	0,02%	0,55%	4,76%	1,88%	3,77%
	OwnText & COutLink (flat)	6.004	0	6.004	36.631	0,8160	0,1840	0,3103	0,5946	0,4078	0,00%	0,02%	0,41%	0,45%	0,56%	0,49%
	OwnText & CBAllLink (flat)	6.004	0	6.004	36.631	0,8193	0,1807	0,3207	0,6061	0,4195	0,00%	0,02%	0,81%	3,82%	2,50%	3,38%
	OwnText & CBAllLink (link)	6.004	0	6.004	36.624	0,8348	0,1652	0,3178	0,6193	0,4201	0,00%	0,00%	2,72%	2,88%	4,74%	3,52%
	IOBridge	6.004	1.473	4.531	7	0,9213	0,0787	0,4676	0,3902	0,4254	-24,53%	-99,98%	13,36%	51,38%	-34,01%	4,83%
	IOBridgeLocal	**6.004**	**3.866**	**2.138**	**7**	**0,9844**	**0,0156**	**0,6486**	**0,6148**	**0,6312**	**-64,39%**	**-99,98%**	**21,13%**	**109,97%**	**3,97%**	**55,54%**
WebKB (10 CV)	OwnText	6.004	0	6.004	36.624	0,8596	0,1404	0,4745	0,6782	0,5583						
	CInLink	6.004	1.024	4.980	7	0,8910	0,1090	0,0884	0,1429	0,1092	-17,06%	-99,98%	3,65%	-81,37%	-78,93%	-80,44%
	COutLink	6.004	2.423	3.581	7	0,8299	0,1701	0,1712	0,1960	0,1828	-40,36%	-99,98%	-3,46%	-63,92%	-71,10%	-67,26%
	CBAllLink	6.004	1	6.003	7	0,8715	0,1285	0,2190	0,1674	0,1897	-0,02%	-99,98%	1,38%	-53,85%	-75,32%	-66,02%
	OwnText & CInLink (flat)	6.004	0	6.004	36.631	0,8626	0,1374	0,4815	0,6877	0,5664	0,00%	0,02%	0,35%	1,48%	1,40%	1,45%
	OwnText & COutLink (flat)	6.004	0	6.004	36.631	0,8616	0,1384	0,4770	0,6884	0,5635	0,00%	0,02%	0,23%	0,53%	1,50%	0,93%
	OwnText & CBAllLink (flat)	6.004	0	6.004	36.631	0,8645	0,1355	0,4825	0,6968	0,5701	0,00%	0,02%	0,57%	1,69%	2,74%	2,11%
	OwnText & CBAllLink (link)	6.004	0	6.004	36.624	0,8771	0,1229	0,4895	0,6922	0,5735	0,00%	0,00%	2,04%	3,16%	2,06%	2,72%
	IOBridge	6.004	1.473	4.531	7	0,9042	0,0958	0,4747	0,3736	0,4182	-24,53%	-99,98%	5,19%	0,04%	-44,91%	-25,09%
	IOBridgeLocal	**6.004**	**3.866**	**2.138**	**7**	**0,9789**	**0,0211**	**0,8074**	**0,7917**	**0,7995**	**-64,39%**	**-99,98%**	**13,88%**	**70,16%**	**16,74%**	**43,20%**

Tabelle A.7: Ergebnisse der Methoden von Chakrabarti et al. (Datenbanken WebKB)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	**Methode**	**Docs**	**NoCoverage**	**Coverage**	**Features**	**Accuracy**	**Error**	**Precision**	**Recall**	**F1**	**Coverage**	**Features**	**Accuracy**	**Precision**	**Recall**	**F1**
BankSearch1 (10 CV)	OwnText	2.176	0	2.176	43.690	0,9754	0,0246	0,9548	0,9473	0,9510						
	CInLink	2.176	266	1.910	4	1,0000	0,0000	1,0000	1,0000	1,0000	-12,22%	-99,99%	2,52%	4,73%	5,56%	5,15%
	COutLink	2.176	290	1.886	4	1,0000	0,0000	1,0000	1,0000	1,0000	-13,33%	-99,99%	2,52%	4,73%	5,56%	5,15%
	CBAllLink	2.176	1	2.175	4	1,0000	0,0000	1,0000	1,0000	1,0000	-0,05%	-99,99%	2,52%	4,73%	5,56%	5,15%
	OwnText & CInLink (flat)	2.176	0	2.176	43.694	0,9812	0,0188	0,9641	0,9593	0,9617	0,00%	0,01%	0,59%	0,97%	1,27%	1,13%
	OwnText & COutLink (flat)	2.176	0	2.176	43.694	0,9800	0,0200	0,9621	0,9568	0,9595	0,00%	0,01%	0,47%	0,76%	1,00%	0,89%
	OwnText & CBAllLink (flat)	2.176	0	2.176	43.694	0,9825	0,0175	0,9664	0,9622	0,9643	0,00%	0,01%	0,73%	1,21%	1,57%	1,40%
	OwnText & CBAllLink (link)	2.176	0	2.176	43.690	0,9830	0,0170	0,9673	0,9630	0,9651	0,00%	0,00%	0,78%	1,31%	1,66%	1,48%
	IOBridge	2.176	352	1.824	4	1,0000	0,0000	1,0000	1,0000	1,0000	-16,18%	-99,99%	2,52%	4,73%	5,56%	5,15%
	IOBridgeLocal	**2.176**	**990**	**1.186**	**4**	**1,0000**	**0,0000**	**1,0000**	**1,0000**	**1,0000**	**-45,50%**	**-99,99%**	**2,52%**	**4,73%**	**5,56%**	**5,15%**
BankSearch2 (10 CV)	OwnText	6.076	0	6.076	90.736	0,9813	0,0187	0,9034	0,8925	0,8979						
	CInLink	6.076	952	5.124	11	0,9992	0,0008	0,9948	0,9948	0,9948	-15,67%	-99,99%	1,82%	10,12%	11,46%	10,79%
	COutLink	6.076	846	5.230	11	0,9992	0,0008	0,9953	0,9953	0,9953	-13,92%	-99,99%	1,82%	10,17%	11,52%	10,85%
	CBAllLink	6.076	7	6.069	11	0,9992	0,0008	0,9949	0,9949	0,9949	-0,12%	-99,99%	1,82%	10,13%	11,47%	10,80%
	OwnText & CInLink (flat)	6.076	0	6.076	90.747	0,9858	0,0142	0,9261	0,9179	0,9219	0,00%	0,01%	0,46%	2,51%	2,85%	2,67%
	OwnText & COutLink (flat)	6.076	0	6.076	90.747	0,9852	0,0148	0,9229	0,9148	0,9188	0,00%	0,01%	0,40%	2,16%	2,50%	2,33%
	OwnText & CBAllLink (flat)	6.076	0	6.076	90.747	0,9868	0,0132	0,9311	0,9240	0,9275	0,00%	0,01%	0,56%	3,07%	3,53%	3,30%
	OwnText & CBAllLink (link)	6.076	0	6.076	90.736	0,9869	0,0131	0,9316	0,9243	0,9279	0,00%	0,00%	0,57%	3,12%	3,56%	3,34%
	IOBridge	6.076	1.169	4.907	11	0,9989	0,0011	0,9930	0,9931	0,9930	-19,24%	-99,99%	1,79%	9,92%	11,27%	10,59%
	IOBridgeLocal	**6.076**	**2.887**	**3.189**	**11**	**0,9991**	**0,0009**	**0,9939**	**0,9955**	**0,9947**	**-47,51%**	**-99,99%**	**1,81%**	**10,02%**	**11,54%**	**10,78%**

Tabelle A.8: Ergebnisse der Methoden von Chakrabarti et al. (Datenbanken BankSearch)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
ODP100 (10 CV)	OwnText	95	1	94	3.791	0,8681	0,1319	0,6801	0,6766	0,6783						
	CInLink	95	77	18	4	0,8611	0,1389	0,4861	0,5104	0,4980	-80,85%	-99,89%	-0,81%	-28,53%	-24,56%	-26,58%
	COutLink	95	75	20	4	0,7750	0,2250	0,3973	0,3973	0,3973	-78,72%	-99,89%	-10,72%	-41,58%	-41,28%	-41,43%
	CBAllLink	95	65	30	4	0,8500	0,1500	0,5611	0,5604	0,5608	-68,09%	-99,89%	-2,09%	-17,50%	-17,17%	-17,32%
	OwnText & CInLink (flat)	95	1	94	3.795	0,8681	0,1319	0,6801	0,6766	0,6783	0,00%	0,11%	0,00%	0,00%	0,00%	0,00%
	OwnText & COutLink (flat)	95	1	94	3.795	0,8681	0,1319	0,6801	0,6766	0,6783	0,00%	0,11%	0,00%	0,00%	0,00%	0,00%
	OwnText & CBAllLink (flat)	95	1	94	3.795	0,8723	0,1277	0,6911	0,6891	0,6901	0,00%	0,11%	0,48%	1,62%	1,85%	1,74%
	OwnText & CBAllLink (link)	95	1	94	3.791	0,8723	0,1277	0,6911	0,6891	0,6901	0,00%	0,00%	0,48%	1,62%	1,85%	1,74%
	IOBridge	**95**	**11**	**84**	**5**	**1,0000**	**0,0000**	**1,0000**	**1,0000**	**1,0000**	**-10,64%**	**-99,87%**	**15,19%**	**47,04%**	**47,80%**	**47,43%**
	IOBridgeLocal	95	13	82	5	1,0000	0,0000	1,0000	1,0000	1,0000	-12,77%	-99,87%	15,19%	47,04%	47,80%	47,43%
ODP500 (10 CV)	OwnText	483	8	475	12.857	0,8880	0,1120	0,7400	0,7132	0,7264						
	CInLink	483	419	64	5	0,9375	0,0625	0,7299	0,7212	0,7255	-86,53%	-99,96%	5,57%	-1,36%	1,12%	-0,12%
	COutLink	483	406	77	5	0,8597	0,1403	0,5193	0,5116	0,5154	-83,79%	-99,96%	-3,19%	-29,82%	-28,27%	-29,05%
	CBAllLink	483	369	114	5	0,8807	0,1193	0,6024	0,5556	0,5781	-76,00%	-99,96%	-0,82%	-18,59%	-22,10%	-20,42%
	OwnText & CInLink (flat)	483	8	475	12.862	0,8922	0,1078	0,7506	0,7242	0,7371	0,00%	0,04%	0,47%	1,43%	1,54%	1,47%
	OwnText & COutLink (flat)	483	8	475	12.862	0,8888	0,1112	0,7454	0,7151	0,7299	0,00%	0,04%	0,09%	0,73%	0,27%	0,48%
	OwnText & CBAllLink (flat)	483	8	475	12.862	0,8897	0,1103	0,7471	0,7174	0,7319	0,00%	0,04%	0,19%	0,96%	0,59%	0,76%
	OwnText & CBAllLink (link)	483	8	475	12.857	0,9018	0,0982	0,8488	0,7510	0,7969	0,00%	0,00%	1,55%	14,70%	5,30%	9,71%
	IOBridge	483	34	449	5	0,9635	0,0365	0,9220	0,9068	0,9143	-5,47%	-99,96%	8,50%	24,59%	27,15%	25,87%
	IOBridgeLocal	**483**	**39**	**444**	**5**	**1,0000**	**0,0000**	**1,0000**	**1,0000**	**1,0000**	**-6,53%**	**-99,96%**	**12,61%**	**35,14%**	**40,21%**	**37,67%**
ODP5000 (10 CV)	OwnText	5.257	62	5.195	71.268	0,9081	0,0919	0,7788	0,7701	0,7744						
	CInLink	5.257	4.648	609	5	0,9186	0,0814	0,7911	0,7867	0,7889	-88,28%	-99,99%	1,16%	1,58%	2,16%	1,87%
	COutLink	5.257	4.442	815	5	0,8621	0,1379	0,6961	0,6252	0,6587	-84,31%	-99,99%	-5,07%	-10,62%	-18,82%	-14,94%
	CBAllLink	5.257	4.022	1.235	5	0,8824	0,1176	0,7173	0,6854	0,7010	-76,23%	-99,99%	-2,83%	-7,90%	-11,00%	-9,48%
	OwnText & CInLink (flat)	5.257	53	5.204	71.273	0,9104	0,0896	0,7843	0,7759	0,7801	0,17%	0,01%	0,25%	0,71%	0,75%	0,74%
	OwnText & COutLink (flat)	5.257	62	5.195	71.273	0,9081	0,0919	0,7795	0,7703	0,7749	0,00%	0,01%	0,00%	0,09%	0,03%	0,06%
	OwnText & CBAllLink (flat)	5.257	53	5.204	71.273	0,9101	0,0899	0,7843	0,7753	0,7798	0,17%	0,01%	0,22%	0,71%	0,68%	0,70%
	OwnText & CBAllLink (link)	5.257	62	5.195	71.268	0,9436	0,0564	0,8648	0,8480	0,8563	0,00%	0,00%	3,91%	11,04%	10,12%	10,58%
	IOBridge	5.257	322	4.935	5	0,9247	0,0753	0,8550	0,8100	0,8319	-5,00%	-99,99%	1,83%	9,78%	5,18%	7,43%
	IOBridgeLocal	**5.257**	**359**	**4.898**	**5**	**0,9882**	**0,0118**	**0,9708**	**0,9706**	**0,9707**	**-5,72%**	**-99,99%**	**8,82%**	**24,65%**	**26,04%**	**25,35%**

Tabelle A.9: Ergebnisse der Methoden von Chakrabarti et al. (Datenbanken ODP)

A.3 Methoden von Lu & Getoor

In Tabelle A.10 findet sich eine Beschreibung der getesteten Methoden. Die Messwerte befinden sich in Tabelle A.11, Tabelle A.12, Tabelle A.13.

Methode	Beschreibung
OwnText	Eigener Text (Basiswert)
Flat-Count (In-/Out-Links)	Anzahl der In- bzw. Out-Links **(Flat)**
Link-Count (In-/Out-Links)	Anzahl der In- bzw. Out-Links **(Link)**
Flat-Mode	Anzahl & Art (In-, Out- & Co-Links) aller Links **(Flat)**
Link-Mode	Anzahl & Art (In-, Out- & Co-Links) aller Links **(Link)**
Flat-Binary	Binärvektor aller Links **(Flat)**
Link-Binary	Binärvektor aller Links **(Link)**
Flat-Count	Anzahl aller Links **(Flat)**
Link-Count	Anzahl aller Links **(Link)**
Flat = Sammlung aller Features einer Methode in einem gemeinsamen Meta-Dokument (merging) **Link** = Getrennte Betrachtung gleichartiger Features durch die Verwendung von 2 getrennten Klassifizierern	

Tabelle A.10: Getestete Methoden von Lu & Getoor

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
WebKB (5 CV)	OwnText	6.004	0	6.004	36.624	0,7813	0,2187	0,3166	0,3340	0,3250						
	Flat-Count (in/out-Links)	6.004	0	6.004	36.631	0,7890	0,2110	0,3358	0,3575	0,3463	0,00%	0,02%	0,99%	6,06%	7,04%	6,55%
	Link-Count (in/out-Links)	6.004	0	6.004	36.624	0,7973	0,2027	0,3272	0,3552	0,3406	0,00%	0,00%	2,05%	3,35%	6,35%	4,80%
	Flat-Mode	**6.004**	**0**	**6.004**	**36.645**	**0,8008**	**0,1992**	**0,4350**	**0,4455**	**0,4402**	**0,00%**	**0,06%**	**2,50%**	**37,40%**	**33,38%**	**35,45%**
	Link-Mode	6.004	0	6.004	36.624	0,8018	0,1982	0,3569	0,3607	0,3588	0,00%	0,00%	2,62%	12,73%	7,99%	10,40%
	Flat-Binary	6.004	0	6.004	36.666	0,7781	0,2219	0,3266	0,3429	0,3345	0,00%	0,11%	-0,41%	3,16%	2,66%	2,92%
	Link-Binary	6.004	0	6.004	36.624	0,8110	0,1890	0,3541	0,3814	0,3673	0,00%	0,00%	3,80%	11,84%	14,19%	13,02%
	Flat-Count	6.004	0	6.004	36.631	0,7913	0,2087	0,3508	0,4235	0,3837	0,00%	0,02%	1,28%	10,80%	26,80%	18,06%
	Link-Count	6.004	0	6.004	36.624	0,7908	0,2092	0,3182	0,3466	0,3318	0,00%	0,00%	1,22%	0,51%	3,77%	2,09%
WebKB (4 CV)	OwnText	6.004	0	6.004	36.624	0,8127	0,1873	0,3089	0,5913	0,4058						
	Flat-Count (in/out-Links)	6.004	0	6.004	36.631	0,8193	0,1807	0,3207	0,6061	0,4195	0,00%	0,02%	0,81%	3,82%	2,50%	3,38%
	Link-Count (in/out-Links)	6.004	0	6.004	36.624	0,8348	0,1652	0,3178	0,6193	0,4201	0,00%	0,00%	2,72%	2,88%	4,74%	3,52%
	Flat-Mode	6.004	0	6.004	36.645	0,8106	0,1894	0,3672	0,5844	0,4510	0,00%	0,06%	-0,26%	18,87%	-1,17%	11,14%
	Link-Mode	**6.004**	**0**	**6.004**	**36.624**	**0,8427**	**0,1573**	**0,3478**	**0,6389**	**0,4505**	**0,00%**	**0,00%**	**3,69%**	**12,59%**	**8,05%**	**11,02%**
	Flat-Binary	6.004	0	6.004	36.666	0,8030	0,1970	0,3167	0,5973	0,4140	0,00%	0,11%	-1,19%	2,53%	1,01%	2,02%
	Link-Binary	6.004	0	6.004	36.624	0,8493	0,1507	0,3417	0,6458	0,4469	0,00%	0,00%	4,50%	10,62%	9,22%	10,13%
	Flat-Count	6.004	0	6.004	36.631	0,7972	0,2028	0,2817	0,5510	0,3728	0,00%	0,02%	-1,91%	-8,81%	-6,82%	-8,13%
	Link-Count	6.004	0	6.004	36.624	0,8266	0,1734	0,3096	0,6119	0,4112	0,00%	0,00%	1,71%	0,23%	3,48%	1,33%
WebKB (10 CV)	OwnText	6.004	0	6.004	36.624	0,8596	0,1404	0,4745	0,6782	0,5583						
	Flat-Count (in/out-Links)	6.004	0	6.004	36.631	0,8645	0,1355	0,4825	0,6968	0,5701	0,00%	0,02%	0,57%	1,69%	2,74%	2,11%
	Link-Count (in/out-Links)	6.004	0	6.004	36.624	0,8771	0,1229	0,4895	0,6922	0,5735	0,00%	0,00%	2,04%	3,16%	2,06%	2,72%
	Flat-Mode	6.004	0	6.004	36.645	0,8503	0,1497	0,5054	0,6752	0,5781	0,00%	0,06%	-1,08%	6,51%	-0,44%	3,55%
	Link-Mode	**6.004**	**0**	**6.004**	**36.624**	**0,8841**	**0,1159**	**0,5117**	**0,7185**	**0,5977**	**0,00%**	**0,00%**	**2,85%**	**7,84%**	**5,94%**	**7,06%**
	Flat-Binary	6.004	0	6.004	36.666	0,8468	0,1532	0,4696	0,6574	0,5479	0,00%	0,11%	-1,49%	-1,03%	-3,07%	-1,86%
	Link-Binary	6.004	0	6.004	36.624	0,8874	0,1126	0,5100	0,7173	0,5961	0,00%	0,00%	3,23%	7,48%	5,77%	6,77%
	Flat-Count	6.004	0	6.004	36.631	0,8379	0,1621	0,4417	0,6439	0,5240	0,00%	0,02%	-2,52%	-6,91%	-5,06%	-6,14%
	Link-Count	6.004	0	6.004	36.624	0,8705	0,1295	0,4807	0,6954	0,5685	0,00%	0,00%	1,27%	1,31%	2,54%	1,83%
webkbLG4f	OwnText	548	0	548	6.763	0,8796	0,1204	0,7113	0,7291	0,7201						
	Flat-Count (in/out-Links)	548	0	548	6.767	0,8850	0,1150	0,7300	0,7467	0,7383	0,00%	0,06%	0,61%	2,63%	2,41%	2,53%
	Link-Count (in/out-Links)	548	0	548	6.763	0,8859	0,1141	0,7318	0,7371	0,7345	0,00%	0,00%	0,72%	2,88%	1,10%	2,00%
	Flat-Mode	548	0	548	6.775	0,8878	0,1122	0,7341	0,7572	0,7454	0,00%	0,18%	0,93%	3,21%	3,85%	3,51%
	Link-Mode	548	0	548	6.763	0,8914	0,1086	0,7374	0,7580	0,7475	0,00%	0,00%	1,34%	3,67%	3,96%	3,81%
	Flat-Binary	548	0	548	6.787	0,8786	0,1214	0,7316	0,7201	0,7258	0,00%	0,35%	-0,11%	2,85%	-1,23%	0,79%
	Link-Binary	**548**	**0**	**548**	**6.763**	**0,8914**	**0,1086**	**0,7402**	**0,7576**	**0,7488**	**0,00%**	**0,00%**	**1,34%**	**4,06%**	**3,91%**	**3,99%**
	Flat-Count	548	0	548	6.767	0,8805	0,1195	0,7208	0,7360	0,7283	0,00%	0,06%	0,10%	1,34%	0,95%	1,14%
	Link-Count	548	0	548	6.763	0,8786	0,1214	0,7152	0,7196	0,7174	0,00%	0,00%	-0,11%	0,55%	-1,30%	-0,37%

Tabelle A.11: Ergebnisse der Methoden von Lu & Getoor (Datenbanken WebKB)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
BankSearch1 (10 CV)	OwnText	2.176	0	2.176	43.690	0,9754	0,0246	0,9548	0,9473	0,9510						
	Flat-Count (in/out-Links)	2.176	0	2.176	43.694	0,9825	0,0175	0,9664	0,9622	0,9643	0,00%	0,01%	0,73%	1,21%	1,57%	1,40%
	Link-Count (in/out-Links)	2.176	0	2.176	43.690	0,9830	0,0170	0,9673	0,9630	0,9651	0,00%	0,00%	0,78%	1,31%	1,66%	1,48%
	Flat-Mode	2.176	0	2.176	43.702	0,9871	0,0129	0,9751	0,9722	0,9737	0,00%	0,03%	1,20%	2,13%	2,63%	2,39%
	Link-Mode	**2.176**	**0**	**2.176**	**43.690**	**0,9874**	**0,0126**	**0,9755**	**0,9726**	**0,9740**	**0,00%**	**0,00%**	**1,23%**	**2,17%**	**2,67%**	**2,42%**
	Flat-Binary	2.176	0	2.176	43.714	0,9858	0,0142	0,9723	0,9689	0,9706	0,00%	0,05%	1,07%	1,83%	2,28%	2,06%
	Link-Binary	2.176	0	2.176	43.690	0,9874	0,0126	0,9755	0,9726	0,9740	0,00%	0,00%	1,23%	2,17%	2,67%	2,42%
	Flat-Count	2.176	0	2.176	43.694	0,9832	0,0168	0,9676	0,9634	0,9655	0,00%	0,01%	0,80%	1,34%	1,70%	1,52%
	Link-Count	2.176	0	2.176	43.690	0,9830	0,0170	0,9673	0,9630	0,9651	0,00%	0,00%	0,78%	1,31%	1,66%	1,48%
BankSearch2 (10 CV)	OwnText	6.076	0	6.076	90.736	0,9813	0,0187	0,9034	0,8925	0,8979						
	Flat-Count (in/out-Links)	6.076	0	6.076	90.747	0,9868	0,0132	0,9311	0,9240	0,9275	0,00%	0,01%	0,56%	3,07%	3,53%	3,30%
	Link-Count (in/out-Links)	6.076	0	6.076	90.736	0,9869	0,0131	0,9316	0,9243	0,9279	0,00%	0,00%	0,57%	3,12%	3,56%	3,34%
	Flat-Mode	6.076	0	6.076	90.769	0,9905	0,0095	0,9491	0,9447	0,9469	0,00%	0,04%	0,94%	5,06%	5,85%	5,46%
	Link-Mode	6.076	0	6.076	90.736	0,9904	0,0096	0,9490	0,9441	0,9465	0,00%	0,00%	0,93%	5,05%	5,78%	5,41%
	Flat-Binary	6.076	0	6.076	90.802	0,9854	0,0146	0,9288	0,9144	0,9216	0,00%	0,07%	0,42%	2,81%	2,45%	2,64%
	Link-Binary	**6.076**	**0**	**6.076**	**90.736**	**0,9905**	**0,0095**	**0,9494**	**0,9445**	**0,9469**	**0,00%**	**0,00%**	**0,94%**	**5,09%**	**5,83%**	**5,46%**
	Flat-Count	6.076	0	6.076	90.747	0,9875	0,0125	0,9341	0,9280	0,9311	0,00%	0,01%	0,63%	3,40%	3,98%	3,70%
	Link-Count	6.076	0	6.076	90.736	0,9871	0,0129	0,9326	0,9254	0,9290	0,00%	0,00%	0,59%	3,23%	3,69%	3,46%

Tabelle A.12: Ergebnisse der Methoden von Lu & Getoor (Datenbanken BankSearch)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
ODP100 (10 CV)	OwnText	95	1	94	3.791	0,8681	0,1319	0,6801	0,6766	0,6783						
	Flat-Count (in/out-Links)	95	1	94	3.795	0,8723	0,1277	0,6911	0,6891	0,6901	0,00%	0,11%	0,48%	1,62%	1,85%	1,74%
	Link-Count (in/out-Links)	95	1	94	3.791	0,8723	0,1277	0,6911	0,6891	0,6901	0,00%	0,00%	0,48%	1,62%	1,85%	1,74%
	Flat-Mode	95	0	95	3.798	0,9284	0,0716	0,8541	0,8275	0,8406	1,06%	0,18%	6,95%	25,58%	22,30%	23,93%
	Link-Mode	**95**	**1**	**94**	**3.791**	**0,9333**	**0,0667**	**0,8574**	**0,8440**	**0,8506**	**0,00%**	**0,00%**	**7,51%**	**26,07%**	**24,74%**	**25,40%**
	Flat-Binary	95	0	95	3.813	0,8821	0,1179	0,8153	0,7100	0,7590	1,06%	0,58%	1,61%	19,88%	4,94%	11,90%
	Link-Binary	95	1	94	3.791	0,9242	0,0758	0,8549	0,8175	0,8358	0,00%	0,00%	6,46%	25,70%	20,82%	23,22%
	Flat-Count	95	0	95	3.795	0,9242	0,0758	0,8435	0,8175	0,8303	1,06%	0,11%	6,46%	24,03%	20,82%	22,41%
	Link-Count	95	1	94	3.791	0,9236	0,0764	0,8492	0,8191	0,8339	0,00%	0,00%	6,39%	24,86%	21,06%	22,94%
ODP500 (10 CV)	OwnText	483	8	475	12.857	0,8880	0,1120	0,7400	0,7132	0,7264						
	Flat-Count (in/out-Links)	483	8	475	12.862	0,8897	0,1103	0,7471	0,7174	0,7319	0,00%	0,04%	0,19%	0,96%	0,59%	0,76%
	Link-Count (in/out-Links)	483	8	475	12.857	0,9018	0,0982	0,8488	0,7510	0,7969	0,00%	0,00%	1,55%	14,70%	5,30%	9,71%
	Flat-Mode	483	1	482	12.867	0,9336	0,0664	0,8617	0,8285	0,8448	1,47%	0,08%	5,14%	16,45%	16,17%	16,30%
	Link-Mode	**483**	**8**	**475**	**12.857**	**0,9422**	**0,0578**	**0,8723**	**0,8492**	**0,8606**	**0,00%**	**0,00%**	**6,10%**	**17,88%**	**19,07%**	**18,47%**
	Flat-Binary	483	0	483	12.882	0,9097	0,0903	0,8098	0,7681	0,7884	1,68%	0,19%	2,44%	9,43%	7,70%	8,54%
	Link-Binary	483	8	475	12.857	0,9354	0,0646	0,8629	0,8326	0,8474	0,00%	0,00%	5,34%	16,61%	16,74%	16,66%
	Flat-Count	483	1	482	12.862	0,9303	0,0697	0,8543	0,8203	0,8370	1,47%	0,04%	4,76%	15,45%	15,02%	15,23%
	Link-Count	483	8	475	12.857	0,9406	0,0594	0,8663	0,8450	0,8555	0,00%	0,00%	5,92%	17,07%	18,48%	17,77%
ODP5000 (10 CV)	OwnText	5.257	62	5.195	71.268	0,9081	0,0919	0,7788	0,7701	0,7744						
	Flat-Count (in/out-Links)	5.257	53	5.204	71.273	0,9101	0,0899	0,7843	0,7753	0,7798	0,17%	0,01%	0,22%	0,71%	0,68%	0,70%
	Link-Count (in/out-Links)	**5.257**	**62**	**5.195**	**71.268**	**0,9436**	**0,0564**	**0,8648**	**0,8480**	**0,8563**	**0,00%**	**0,00%**	**3,91%**	**11,04%**	**10,12%**	**10,58%**
	Flat-Mode	5.257	3	5.254	71.283	0,9343	0,0657	0,8487	0,8358	0,8422	1,14%	0,02%	2,89%	8,98%	8,53%	8,76%
	Link-Mode	5.257	62	5.195	71.268	0,9353	0,0647	0,8455	0,8381	0,8418	0,00%	0,00%	3,00%	8,56%	8,83%	8,70%
	Flat-Binary	5.257	0	5.257	71.298	0,9268	0,0732	0,8323	0,8168	0,8245	1,19%	0,04%	2,06%	6,87%	6,06%	6,47%
	Link-Binary	5.257	62	5.195	71.268	0,9345	0,0655	0,8427	0,8362	0,8395	0,00%	0,00%	2,91%	8,20%	8,58%	8,41%
	Flat-Count	5.257	3	5.254	71.273	0,9336	0,0664	0,8468	0,8341	0,8404	1,14%	0,01%	2,81%	8,73%	8,31%	8,52%
	Link-Count	5.257	62	5.195	71.268	0,9339	0,0661	0,8426	0,8348	0,8387	0,00%	0,00%	2,84%	8,19%	8,40%	8,30%

Tabelle A.13: Ergebnisse der Methoden von Lu & Getoor (Datenbanken ODP)

A.4 Link-Local-Methoden

In Tabelle A.14 findet sich eine Beschreibung der getesteten Methoden. Die Messwerte befinden sich in Tabelle A.15, Tabelle A.16, Tabelle A.17.

Methode	Beschreibung
OwnText	Eigener Text (Basiswert)
WA 10	Words-Around mit einem Radius von 10 Worten (vor bzw. nach einem Link)
WA 300	Words-Around mit einem Radius von 300 Worten (vor bzw. nach einem Link)
(1) - LinkDescription	Textuelle HTML-Beschreibung eines Links (Bildschirmausgabe)
LinkParagraph	Gesamter Abschnitt eines Links
(2) - LinkHeader	Vorhergehende Überschrift
(3) - LinkListHeader	Vorhergehende Listenüberschrift
OwnTitle	Eigener Seitentitel
InTitle	Seitentitel aller In-Links
OutTitle	Seitentitel aller Out-Links
AllTitle	Seitentitel der eigenen Seite & aller Nachbarn
WA 10 & (1)	**Kombination** von WordsAround & Link-Description
WA 300 & (1)	**Kombination** von WordsAround & Link-Description
WA 10 & (1) & (2) & (3)	**Kombination** von WordsAround, Link-Description, Link-Header & Link-List-Header
Bei der **Kombination** von Features werden alle Features einer Methode in einem gemeinsamen Meta-Dokument gesammelt (merging)	

Tabelle A.14: Getestete Link-Local-Methoden

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
WebKB (5 CV)	OwnText	6.004	0	6.004	36.624	0,7813	0,2187	0,3166	0,3340	0,3250						
	WA 10	6.004	1.023	4.981	8.931	0,8709	0,1291	0,4485	0,4786	0,4631	-17,04%	-75,61%	11,47%	41,66%	43,29%	42,49%
	WA 300	6.004	1.023	4.981	33.298	0,8097	0,1903	0,4228	0,3921	0,4069	-17,04%	-9,08%	3,63%	33,54%	17,40%	25,20%
	(1) - LinkDescription	6.004	1.156	4.848	2.385	0,8870	0,1130	0,4988	0,3296	0,3970	-19,25%	-93,49%	13,53%	57,55%	-1,32%	22,15%
	LinkParagraph	6.004	4.882	1.122	3.497	0,8401	0,1599	0,2881	0,3262	0,3060	-81,31%	-90,45%	7,53%	-9,00%	-2,34%	-5,85%
	(2) - LinkHeader	6.004	1.405	4.599	2.937	0,8516	0,1484	0,3514	0,3310	0,3409	-23,40%	-91,98%	9,00%	10,99%	-0,90%	4,89%
	(3) - LinkListHeader	6.004	3.490	2.514	1.256	0,8479	0,1521	0,3403	0,3332	0,3367	-58,13%	-96,57%	8,52%	7,49%	-0,24%	3,60%
	OwnTitle	6.004	249	5.755	1.758	0,8808	0,1192	0,3849	0,3895	0,3872	-4,15%	-95,20%	12,74%	21,57%	16,62%	19,14%
	InTitle	6.004	1.130	4.874	2.434	0,8819	0,1181	0,3914	0,3290	0,3575	-18,82%	-93,35%	12,88%	23,63%	-1,50%	10,00%
	OutTitle	6.004	2.497	3.507	2.492	0,7909	0,2091	0,2367	0,2280	0,2323	-41,59%	-93,20%	1,23%	-25,24%	-31,74%	-28,52%
	AllTitle	6.004	23	5.981	5.510	0,8131	0,1869	0,2989	0,2713	0,2844	-0,38%	-84,96%	4,07%	-5,59%	-18,77%	-12,49%
	WA 10 & (1)	**6.004**	**1.023**	**4.981**	**9.478**	**0,8776**	**0,1224**	**0,4519**	**0,4768**	**0,4640**	**-17,04%**	**-74,12%**	**12,33%**	**42,74%**	**42,75%**	**42,77%**
	WA 300 & (1)	6.004	1.023	4.981	33.369	0,8134	0,1866	0,4258	0,3841	0,4039	-17,04%	-8,89%	4,11%	34,49%	15,00%	24,28%
	WA 10 & (1) & (2) & (3)	6.004	1.023	4.981	10.207	0,8536	0,1464	0,4230	0,4589	0,4402	-17,04%	-72,13%	9,25%	33,61%	37,40%	35,45%
WebKB (4 CV)	OwnText	6.004	0	6.004	36.624	0,8127	0,1873	0,3089	0,5913	0,4058						
	WA 10	**6.004**	**1.023**	**4.981**	**8.931**	**0,8774**	**0,1226**	**0,3888**	**0,4548**	**0,4192**	**-17,04%**	**-75,61%**	**7,96%**	**25,87%**	**-23,08%**	**3,30%**
	WA 300	6.004	1.023	4.981	33.298	0,8021	0,1979	0,2262	0,3493	0,2746	-17,04%	-9,08%	-1,30%	-26,77%	-40,93%	-32,33%
	(1) - LinkDescription	6.004	1.156	4.848	2.385	0,9074	0,0926	0,3612	0,2950	0,3247	-19,25%	-93,49%	11,65%	16,93%	-50,11%	-19,99%
	LinkParagraph	6.004	4.882	1.122	3.497	0,8614	0,1386	0,2494	0,3360	0,2863	-81,31%	-90,45%	5,99%	-19,26%	-43,18%	-29,45%
	(2) - LinkHeader	6.004	1.405	4.599	2.937	0,8613	0,1387	0,3273	0,3548	0,3405	-23,40%	-91,98%	5,98%	5,96%	-40,00%	-16,09%
	(3) - LinkListHeader	6.004	3.490	2.514	1.256	0,8670	0,1330	0,3528	0,4921	0,4110	-58,13%	-96,57%	6,68%	14,21%	-16,78%	1,28%
	OwnTitle	6.004	249	5.755	1.758	0,9161	0,0839	0,3384	0,4369	0,3814	-4,15%	-95,20%	12,72%	9,55%	-26,11%	-6,01%
	InTitle	6.004	1.130	4.874	2.434	0,9019	0,0981	0,3810	0,3797	0,3804	-18,82%	-93,35%	10,98%	23,34%	-35,79%	-6,26%
	OutTitle	6.004	2.497	3.507	2.492	0,8172	0,1828	0,2605	0,3507	0,2989	-41,59%	-93,20%	0,55%	-15,67%	-40,69%	-26,34%
	AllTitle	6.004	23	5.981	5.510	0,8332	0,1668	0,3004	0,3421	0,3199	-0,38%	-84,96%	2,52%	-2,75%	-42,14%	-21,17%
	WA 10 & (1)	6.004	1.023	4.981	9.478	0,8852	0,1148	0,3863	0,4553	0,4180	-17,04%	-74,12%	8,92%	25,06%	-23,00%	3,01%
	WA 300 & (1)	6.004	1.023	4.981	33.369	0,8068	0,1932	0,2263	0,2833	0,2516	-17,04%	-8,89%	-0,73%	-26,74%	-52,09%	-38,00%
	WA 10 & (1) & (2) & (3)	6.004	1.023	4.981	10.207	0,8546	0,1454	0,3433	0,4092	0,3734	-17,04%	-72,13%	5,16%	11,14%	-30,80%	-7,98%
WebKB (10 CV)	OwnText	6.004	0	6.004	36.624	0,8596	0,1404	0,4745	0,6782	0,5583						
	WA 10	**6.004**	**1.023**	**4.981**	**8.931**	**0,9184**	**0,0816**	**0,5396**	**0,6398**	**0,5854**	**-17,04%**	**-75,61%**	**6,84%**	**13,72%**	**-5,66%**	**4,85%**
	WA 300	6.004	1.023	4.981	33.298	0,8891	0,1109	0,5315	0,6121	0,5690	-17,04%	-9,08%	3,43%	12,01%	-9,75%	1,92%
	(1) - LinkDescription	6.004	1.156	4.848	2.385	0,9069	0,0931	0,5584	0,4395	0,4919	-19,25%	-93,49%	5,50%	17,68%	-35,20%	-11,89%
	LinkParagraph	6.004	4.882	1.122	3.497	0,8655	0,1345	0,4068	0,4417	0,4235	-81,31%	-90,45%	0,69%	-14,27%	-34,87%	-24,14%
	(2) - LinkHeader	6.004	1.405	4.599	2.937	0,9169	0,0831	0,5243	0,5450	0,5345	-23,40%	-91,98%	6,67%	10,50%	-19,64%	-4,26%
	(3) - LinkListHeader	6.004	3.490	2.514	1.256	0,9137	0,0863	0,5481	0,5522	0,5501	-58,13%	-96,57%	6,29%	15,51%	-18,58%	-1,47%
	OwnTitle	6.004	249	5.755	1.758	0,9061	0,0939	0,5281	0,4921	0,5094	-4,15%	-95,20%	5,41%	11,30%	-27,44%	-8,76%
	InTitle	6.004	1.130	4.874	2.434	0,9313	0,0687	0,5998	0,5641	0,5814	-18,82%	-93,35%	8,34%	26,41%	-16,82%	4,14%
	OutTitle	6.004	2.497	3.507	2.492	0,8437	0,1563	0,3445	0,3440	0,3443	-41,59%	-93,20%	-1,85%	-27,40%	-49,28%	-38,33%
	AllTitle	6.004	23	5.981	5.510	0,8755	0,1245	0,4276	0,4915	0,4574	-0,38%	-84,96%	1,85%	-9,88%	-27,53%	-18,07%
	WA 10 & (1)	6.004	1.023	4.981	9.478	0,9222	0,0778	0,5409	0,6341	0,5838	-17,04%	-74,12%	7,28%	13,99%	-6,50%	4,57%
	WA 300 & (1)	6.004	1.023	4.981	33.369	0,8911	0,1089	0,5291	0,6047	0,5644	-17,04%	-8,89%	3,66%	11,51%	-10,84%	1,09%
	WA 10 & (1) & (2) & (3)	6.004	1.023	4.981	10.207	0,9159	0,0841	0,5279	0,6502	0,5827	-17,04%	-72,13%	6,55%	11,25%	-4,13%	4,37%

Tabelle A.15: Ergebnisse der Link-Local-Methoden (Datenbanken WebKB)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
BankSearch1 (10 CV)	OwnText	2.176	0	2.176	43.690	0,9754	0,0246	0,9548	0,9473	0,9510						
	WA 10	2.176	265	1.911	4.819	0,9458	0,0542	0,8964	0,8997	0,8980	-12,18%	-88,97%	-3,03%	-6,12%	-5,02%	-5,57%
	WA 300	2.176	265	1.911	32.127	0,9945	0,0055	0,9908	0,9905	0,9906	-12,18%	-26,47%	1,96%	3,77%	4,56%	4,16%
	(1) - LinkDescription	2.176	353	1.823	1.521	0,7540	0,2460	0,6017	0,4737	0,5301	-16,22%	-96,52%	-22,70%	-36,98%	-49,99%	-44,26%
	LinkParagraph	2.176	1.548	628	3.708	0,8885	0,1115	0,7660	0,7569	0,7614	-71,14%	-91,51%	-8,91%	-19,77%	-20,10%	-19,94%
	(2) - LinkHeader	2.176	1.291	885	969	0,9305	0,0695	0,8911	0,7875	0,8361	-59,33%	-97,78%	-4,60%	-6,67%	-16,87%	-12,08%
	(3) - LinkListHeader	2.176	1.888	288	291	0,9028	0,0972	0,8888	0,8224	0,8543	-86,76%	-99,33%	-7,44%	-6,91%	-13,18%	-10,17%
	OwnTitle	2.176	128	2.048	773	0,8611	0,1389	0,8274	0,7169	0,7682	-5,88%	-98,23%	-11,72%	-13,34%	-24,32%	-19,22%
	InTitle	2.176	349	1.827	1.307	0,9381	0,0619	0,9045	0,8777	0,8909	-16,04%	-97,01%	-3,82%	-5,27%	-7,35%	-6,32%
	OutTitle	2.176	365	1.811	1.272	0,9324	0,0676	0,8858	0,8683	0,8770	-16,77%	-97,09%	-4,41%	-7,23%	-8,34%	-7,78%
	AllTitle	2.176	72	2.104	1.724	0,9579	0,0421	0,9240	0,9210	0,9225	-3,31%	-96,05%	-1,79%	-3,23%	-2,78%	-3,00%
	WA 10 & (1)	2.176	265	1.911	5.043	0,9521	0,0479	0,9080	0,9109	0,9095	-12,18%	-88,46%	-2,39%	-4,90%	-3,84%	-4,36%
	WA 300 & (1)	**2.176**	**265**	**1.911**	**32.158**	**0,9945**	**0,0055**	**0,9908**	**0,9905**	**0,9906**	**-12,18%**	**-26,40%**	**1,96%**	**3,77%**	**4,56%**	**4,16%**
	WA 10 & (1) & (2) & (3)	2.176	265	1.911	5.209	0,9652	0,0348	0,9314	0,9329	0,9321	-12,18%	-88,08%	-1,05%	-2,45%	-1,52%	-1,99%
BankSearch2 (10 CV)	OwnText	6.076	0	6.076	90.736	0,9813	0,0187	0,9034	0,8925	0,8979						
	WA 10	6.076	951	5.125	10.061	0,9576	0,0424	0,7735	0,7637	0,7686	-15,65%	-88,91%	-2,42%	-14,38%	-14,43%	-14,40%
	WA 300	6.076	951	5.125	63.842	0,9890	0,0110	0,9464	0,9386	0,9425	-15,65%	-29,64%	0,78%	4,76%	5,17%	4,97%
	(1) - LinkDescription	6.076	1.451	4.625	3.622	0,8817	0,1183	0,4421	0,3347	0,3810	-23,88%	-96,01%	-10,15%	-51,06%	-62,50%	-57,57%
	LinkParagraph	6.076	4.237	1.839	7.353	0,9323	0,0677	0,6674	0,6164	0,6409	-69,73%	-91,90%	-4,99%	-26,12%	-30,94%	-28,62%
	(2) - LinkHeader	6.076	4.066	2.010	1.938	0,9397	0,0603	0,7180	0,6288	0,6705	-66,92%	-97,86%	-4,24%	-20,52%	-29,55%	-25,33%
	(3) - LinkListHeader	6.076	5.467	609	454	0,8601	0,1399	0,5005	0,3236	0,3931	-89,98%	-99,50%	-12,35%	-44,60%	-63,74%	-56,22%
	OwnTitle	6.076	398	5.678	1.870	0,9341	0,0659	0,7764	0,6380	0,7005	-6,55%	-97,94%	-4,81%	-14,06%	-28,52%	-21,98%
	InTitle	6.076	1.215	4.861	2.982	0,9560	0,0440	0,7994	0,7601	0,7793	-20,00%	-96,71%	-2,58%	-11,51%	-14,83%	-13,21%
	OutTitle	6.076	995	5.081	2.770	0,9566	0,0434	0,7913	0,7606	0,7757	-16,38%	-96,95%	-2,52%	-12,41%	-14,78%	-13,61%
	AllTitle	6.076	135	5.941	3.839	0,9696	0,0304	0,8422	0,8342	0,8382	-2,22%	-95,77%	-1,19%	-6,77%	-6,53%	-6,65%
	WA 10 & (1)	6.076	951	5.125	10.653	0,9595	0,0405	0,7817	0,7741	0,7779	-15,65%	-88,26%	-2,22%	-13,47%	-13,27%	-13,36%
	WA 300 & (1)	**6.076**	**951**	**5.125**	**63.920**	**0,9892**	**0,0108**	**0,9469**	**0,9395**	**0,9432**	**-15,65%**	**-29,55%**	**0,81%**	**4,82%**	**5,27%**	**5,05%**
	WA 10 & (1) & (2) & (3)	6.076	951	5.125	10.911	0,9665	0,0335	0,8156	0,8093	0,8124	-15,65%	-87,98%	-1,51%	-9,72%	-9,32%	-9,52%

Tabelle A.16: Ergebnisse der Link-Local-Methoden (Datenbanken BankSearch)

		Messwerte									Abweichungen in % (zu Own Text)					
DB	Methode	Docs	NoCoverage	Coverage	Features	Accuracy	Error	Precision	Recall	F1	Coverage	Features	Accuracy	Precision	Recall	F1
ODP100 (10 CV)	OwnText	95	1	94	3.791	0,8681	0,1319	0,6801	0,6766	0,6783						
	WA 10	95	8	87	2.105	0,9816	0,0184	0,9500	0,9600	0,9550	-7,45%	-44,47%	13,07%	39,69%	41,89%	40,79%
	WA 300	95	8	87	22.489	0,9770	0,0230	0,9390	0,9500	0,9445	-7,45%	493,22%	12,54%	38,07%	40,41%	39,25%
	(1) - LinkDescription	95	8	87	354	0,9034	0,0966	0,7898	0,7564	0,7727	-7,45%	-90,66%	4,07%	16,13%	11,79%	13,92%
	LinkParagraph	95	9	86	1.853	0,9581	0,0419	0,9086	0,9015	0,9050	-8,51%	-51,12%	10,37%	33,60%	33,24%	33,42%
	(2) - LinkHeader	95	8	87	307	0,9770	0,0230	0,9429	0,9500	0,9465	-7,45%	-91,90%	12,54%	38,64%	40,41%	39,54%
	(3) - LinkListHeader	95	10	85	128	0,9812	0,0188	0,9510	0,9579	0,9545	-9,57%	-96,62%	13,03%	39,83%	41,58%	40,72%
	OwnTitle	95	5	90	47	0,7556	0,2444	0,4521	0,3760	0,4105	-4,26%	-98,76%	-12,96%	-33,52%	-44,43%	-39,48%
	InTitle	95	8	87	638	0,9816	0,0184	0,9529	0,9600	0,9565	-7,45%	-83,17%	13,07%	40,11%	41,89%	41,01%
	OutTitle	95	49	46	574	0,8087	0,1913	0,5004	0,5044	0,5024	-51,06%	-84,86%	-6,84%	-26,42%	-25,45%	-25,93%
	AllTitle	95	0	95	1.136	0,9789	0,0211	0,9481	0,9489	0,9485	1,06%	-70,03%	12,76%	39,41%	40,25%	39,83%
	WA 10 & (1)	95	8	87	2.167	0,9816	0,0184	0,9497	0,9600	0,9548	-7,45%	-42,84%	13,07%	39,64%	41,89%	40,76%
	WA 300 & (1)	95	8	87	22.518	0,9770	0,0230	0,9390	0,9500	0,9445	-7,45%	493,99%	12,54%	38,07%	40,41%	39,25%
	WA 10 & (1) & (2) & (3)	**95**	**8**	**87**	**2.451**	**0,9862**	**0,0138**	**0,9625**	**0,9700**	**0,9662**	**-7,45%**	**-35,35%**	**13,60%**	**41,52%**	**43,36%**	**42,44%**
ODP500 (10 CV)	OwnText	483	8	475	12.857	0,8880	0,1120	0,7400	0,7132	0,7264						
	WA 10	483	33	450	7.651	0,9902	0,0098	0,9750	0,9746	0,9748	-5,26%	-40,49%	11,51%	31,76%	36,65%	34,20%
	WA 300	483	33	450	70.373	0,9929	0,0071	0,9828	0,9816	0,9822	-5,26%	447,35%	11,81%	32,81%	37,63%	35,21%
	(1) - LinkDescription	483	33	450	1.658	0,8969	0,1031	0,7682	0,7401	0,7539	-5,26%	-87,10%	1,00%	3,81%	3,77%	3,79%
	LinkParagraph	483	34	449	15.433	0,9367	0,0633	0,8585	0,8359	0,8471	-5,47%	20,04%	5,48%	16,01%	17,20%	16,62%
	(2) - LinkHeader	483	33	450	1.390	0,9964	0,0036	0,9912	0,9914	0,9913	-5,26%	-89,19%	12,21%	33,95%	39,01%	36,47%
	(3) - LinkListHeader	483	37	446	478	0,9435	0,0565	0,8943	0,8538	0,8736	-6,11%	-96,28%	6,25%	20,85%	19,71%	20,26%
	OwnTitle	483	16	467	312	0,8030	0,1970	0,6352	0,5049	0,5626	-1,68%	-97,57%	-9,57%	-14,16%	-29,21%	-22,55%
	InTitle	**483**	**33**	**450**	**2.529**	**0,9973**	**0,0027**	**0,9936**	**0,9933**	**0,9934**	**-5,26%**	**-80,33%**	**12,31%**	**34,27%**	**39,27%**	**36,76%**
	OutTitle	483	255	228	2.149	0,8123	0,1877	0,5746	0,5307	0,5517	-52,00%	-83,29%	-8,52%	-22,35%	-25,59%	-24,05%
	AllTitle	483	1	482	4.292	0,9917	0,0083	0,9794	0,9793	0,9793	1,47%	-66,62%	11,68%	32,35%	37,31%	34,82%
	WA 10 & (1)	483	33	450	7.950	0,9876	0,0124	0,9687	0,9677	0,9682	-5,26%	-38,17%	11,22%	30,91%	35,68%	33,29%
	WA 300 & (1)	483	33	450	70.470	0,9929	0,0071	0,9828	0,9816	0,9822	-5,26%	448,11%	11,81%	32,81%	37,63%	35,21%
	WA 10 & (1) & (2) & (3)	483	33	450	8.827	0,9947	0,0053	0,9862	0,9863	0,9862	-5,26%	-31,34%	12,02%	33,27%	38,29%	35,77%
ODP5000 (10 CV)	OwnText	5.257	62	5.195	71.268	0,9081	0,0919	0,7788	0,7701	0,7744						
	WA 10	5.257	316	4.941	51.484	0,9845	0,0155	0,9619	0,9611	0,9615	-4,89%	-27,76%	8,41%	23,51%	24,80%	24,16%
	WA 300	5.257	316	4.941	415.232	0,9918	0,0082	0,9798	0,9798	0,9798	-4,89%	482,63%	9,22%	25,81%	27,23%	26,52%
	(1) - LinkDescription	5.257	316	4.941	14.418	0,9114	0,0886	0,7823	0,7782	0,7803	-4,89%	-79,77%	0,36%	0,45%	1,05%	0,76%
	LinkParagraph	5.257	368	4.889	70.023	0,9417	0,0583	0,8603	0,8553	0,8578	-5,89%	-1,75%	3,70%	10,46%	11,06%	10,77%
	(2) - LinkHeader	5.257	322	4.935	7.939	0,9805	0,0195	0,9515	0,9511	0,9513	-5,00%	-88,86%	7,97%	22,18%	23,50%	22,84%
	(3) - LinkListHeader	5.257	393	4.864	2.968	0,9386	0,0614	0,8519	0,8465	0,8492	-6,37%	-95,84%	3,36%	9,39%	9,92%	9,66%
	OwnTitle	5.257	197	5.060	2.425	0,8334	0,1666	0,6469	0,5828	0,6132	-2,60%	-96,60%	-8,23%	-16,94%	-24,32%	-20,82%
	InTitle	5.257	316	4.941	14.771	0,9937	0,0063	0,9846	0,9843	0,9844	-4,89%	-79,27%	9,43%	26,43%	27,81%	27,12%
	OutTitle	5.257	2.785	2.472	15.440	0,8602	0,1398	0,7121	0,6406	0,6745	-52,42%	-78,34%	-5,27%	-8,56%	-16,82%	-12,90%
	AllTitle	5.257	13	5.244	25.784	0,9867	0,0133	0,9668	0,9667	0,9667	0,94%	-63,82%	8,66%	24,14%	25,53%	24,83%
	WA 10 & (1)	5.257	316	4.941	54.445	0,9850	0,0150	0,9629	0,9622	0,9626	-4,89%	-23,61%	8,47%	23,64%	24,94%	24,30%
	WA 300 & (1)	**5.257**	**316**	**4.941**	**416.076**	**0,9920**	**0,0080**	**0,9802**	**0,9802**	**0,9802**	**-4,89%**	**483,82%**	**9,24%**	**25,86%**	**27,28%**	**26,58%**
	WA 10 & (1) & (2) & (3)	5.257	316	4.941	58.015	0,9902	0,0098	0,9754	0,9753	0,9754	-4,89%	-18,60%	9,04%	25,24%	26,65%	25,96%

Tabelle A.17: Ergebnisse der Link-Local-Methoden (Datenbanken ODP)

Anhang B: Der Inhalt der beiliegenden DVD

Auf der beiliegenden DVD befinden sich die JAVA-Sourcen des implementierten Test-Frameworks (Abbildung B.1). Diese befinden sich in den Verzeichnissen unter: `/src`.

Weiterhin sind auf der DVD die archivierten[10] Daten der verwendeten Datenbanken (Abbildung B.2) enthalten. Diese Archive liegen im Verzeichnis: `/data`.

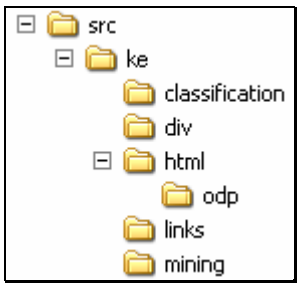

Abbildung B.1: Verzeichnisse der JAVA-Sourcen (/src)

Name	Größe
banksearch1.zip	70.575 KB
banksearch2.zip	199.294 KB
odp100.zip	20.656 KB
odp500.zip	89.806 KB
odp5000.zip	971.731 KB
odpXMLData.zip	381.763 KB
webkb.zip	52.839 KB

Abbildung B.2: Verzeichnis der Datenbanken (/data)

B.1 Java-Klassen des Test-Frameworks

Tabelle B.1 beschreibt die im Rahmen dieser Diplomarbeit implementierten JAVA-Klassen des Test-Frameworks.

[10] Durch die Verwendung eines geeigneten Programms zur Entkomprimierung, wie z. B. „gunzip“, können diese Daten entpackt werden.

Klasse	Beschreibung
StartHTC	Hauptprogramm: Daten-Konvertierung & Test der Methoden zur Hypertext-Klassifikation
ke.classification.Analyse	Aufbau von Feature-Listen, welche die Häufigkeiten von Wörtern in einem StringBuffer beinhalten
ke.classification.Classification	Durchführung von Text-Klassifikationen
ke.classification.NaiveBayes	Implementierung eines Naive-Bayes-Klassifizierers
ke.classification.StartClassification	Starten des Feature-Mining und der Klassifikation von Texten
ke.classification.SVMAccess	Zugriff auf einen SVM-Klassifizierer (SVM-Light)
ke.div.Convert2XML	Konvertierung von HTML-Dateien in XHTML (XML-Format)
ke.div.FileIO	Ein-/Ausgabe von Texten in Dateien
ke.div.PrepareFiles	Entfernen von Headern, Kopieren von Datei-Sammlungen und Generierung von gültigen Dateinamen
ke.div.Utils	Hilfsfunktionen für die Verarbeitung von Dateien, Aufruf von externen Programmen
ke.div.XPathQuery	Zugriff auf XML-Dateien mit XPath
ke.html.AltaVistLinks	Bestimmung der In-Links eines Dokumnetes über den parametrisierten Aufruf einer AltaVista-Seite
ke.html.HTMLUtils	Methoden für den Zugriff auf HTML-Dateien
ke.html.odp.Retrieval	Abfrage von HTML-Seiten (über den ODP-Webkatalog)
ke.html.odp.Topics	Bestimmung der Klasse eines Dokumentes über einen Eintrag im ODP-Katalog
ke.links.LinkList	Liste der ausgehenden und ankommenden Verweise (HTML-Links) von HTML-Dateien
ke.links.Links	Vorbereitung von (HTML-)Dateien für eine Klassifikation (Link-Listen etc.)
ke.mining.Extraction	Methoden für die Extraktion von Features aus Text-Dokumenten
ke.mining.Mining	Startes des Mining von Features

Tabelle B.1: Klassen des Test-Frameworks

B.2 Archive der Testdaten

Alle Testdaten befinden sich jeweils in einem komprimierten Archiv (Abbildung B.2). Die Inhalte dieser Archive (Dokumente, Links und Klassen) werden in Tabelle B.2 beschrieben.

			Im Archiv vorhanden (x) bzw. nicht vorhanden (-)					
Typ	**Dokumentart**	**Verzeichnis**	**BankSearch1**	**BankSeach2**	**WebKB**	**ODP100**	**ODP500**	**ODP5000**
Dokumente	Orginaldaten (von einer externen Datenquelle)	orig	x	x	x	-	-	-
	Orginaldaten, um evtl. vorhandene Header bereinigt	html	x	x	x	-	-	-
	Test-Dokumente	core	x	x	x	x	x	x
	Dokumente der In-Links	xml	x	x	x	x	x	x
	Dokumente der Out-Links	xmlout	in xml enthalten	in xml enthalten	in xml enthalten	x	x	x
		Datei						
Links	Tabellen der In-Links	inlinks	x	x	x	x	x	x
	Tabellen der Out-Links	outlinks	x	x	x	x	x	x
	Tabellen der Co-Links (IO-Bridges)	colinks	x	x	x	x	x	x
Klassen	Klassen der Test-Dokumente	owntopics	x	x	x	x	x	x
	Klassen der In-Links	intopics	x	x	x	x	x	x
	Klassen der Out-Links	outtopics	x	x	x	x	x	x

Tabelle B.2: Archive der Testdaten

a) Struktur der Linklisten (inlinks, outlinks und colinks)

Die Linklisten beinhalten die Vorgängerseiten, Nachfolgerseiten und Seiten der Ko-Zitierung im XML-Format. In den Linklisten wird jede Datei innerhalb des XML-Tags <*zeile*> beschrieben. Innerhalb dieses Tags befinden sich die folgenden Datenfelder:

<*to*> Zieldatei mit dem Dateinamen <*filename*>

<*from*> Eine Liste aller Quelldateien mit dem Dateinamen (<*filename*>) und dem HTML-Link, wie er im HTML-Code der Quelldatei verwendet wird (<*link*>).

Ein Beispiel für eine Linkliste wird in Abbildung B.3 dargestellt.

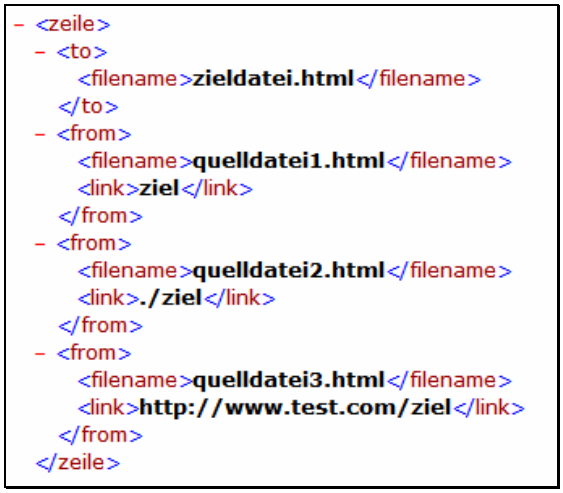

Abbildung B.3: Beispiel einer Linkliste (inlinks)

In diesem Beispiel besitzt die Datei *zieldatei.html* drei verschiedene Vorgängerseiten (*quelldatei1.html*, *quelldatei2.html* und *quelldatei3.html*).

b) Struktur der Klassenlisten (owntopics, intopics und outtopics)

Die Klassenlisten beinhalten die Klassenzuordnungen einer Datei im Textformat und verwenden für jede Datei eine Zeile in einer Liste. Innerhalb einer Zeile wird der Dateiname von dem Klassennamen durch einen Doppelpunkt getrennt. Abbildung B.4 zeigt ein Beispiel aus einer Klassenliste.

```
...
a-instruments.com:Top/Science/Instruments_and_Supplies
ablemedia.com/ctcweb:Top/Arts/Classical_Studies
artificialbrains.com:Top/Computers/Artificial_Intelligence
chichissalsa.com:Top/Business/Food_and_Related_Products
chinosport64.tripod.com/handball.html:Top/Sports/Handball
...
```

Abbildung B.4: Beispiel einer Klassenliste (intopics)

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit „Vergleich von Methoden zur Hypertext-Klassifikation“ selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum	Unterschrift