# KATI

## Endbericht des vom DFN-Verein mit Mitteln des BMBF geförderten Projektes TK 602 – VA/I 104


**Berlin, Oktober 2001**

**Verfasst von:**
Ivonne Kellner

**Mitarbeit:**
Christian Thieme
Dr. Thorsten Wichmann

BERLECON RESEARCH GmbH
Oranienburger Str. 32
10117 Berlin
Tel.: +49 30 285296-0
Fax: +49 30 285296-29
Web: http://www.berlecon.de
Email: info@berlecon.de

V 1.0 011031

# Inhaltsverzeichnis

# 1 Projektziele und -verlauf

Im Projekt *KATI* (Maschinelle Typenkategorisierung thematisch verwandter Internet-Dokumente aus dem Wissenschaftsbereich) sollten Verfahren zur automatischen Typen-Kategorisierung von wissenschaftlichen Internet-Dokumenten entwickelt und implementiert werden. In der Herangehensweise sollte sich *KATI* von den meisten anderen Klassifizierungsprojekten durch folgende Punkte unterscheiden:

- Die Kategorisierung sollte nach **Inhalts*typen*** (Publikationsliste, Jobangebot, statistische Daten) anstatt nach Inhalts*themen* (Arbeitslosigkeit, Wachstum, Spieltheorie) erfolgen.
- Zur Kategorisierung sollten nicht nur der Dokumenttext herangezogen werden, sondern auch die **Metainformationen** des Dokumentes, insbesondere in den HTML-Strukturen, die Anzahl der enthaltenen Links und das Vorkommen bestimmter Strings in der Dokument-URL.

In der konkreten Umsetzung sollte der bestehende Suchdienst *Inomics* (*www.inomics.com*) durch die Möglichkeit der Kategorienauswahl erweitert werden.

Praktisches Ziel war es, eine Internet-Recherche im wissenschaftlichen Bereich durch die Kategorienauswahl deutlich zu erleichtern, indem:

- die Suchanfrage auf einfache Art spezialisiert werden kann,
- der Suchraum auf Dokumente der gewünschten Kategorien eingeschränkt wird und
- die Ergebnismenge mit höherer Wahrscheinlichkeit relevante Dokumente enthält.

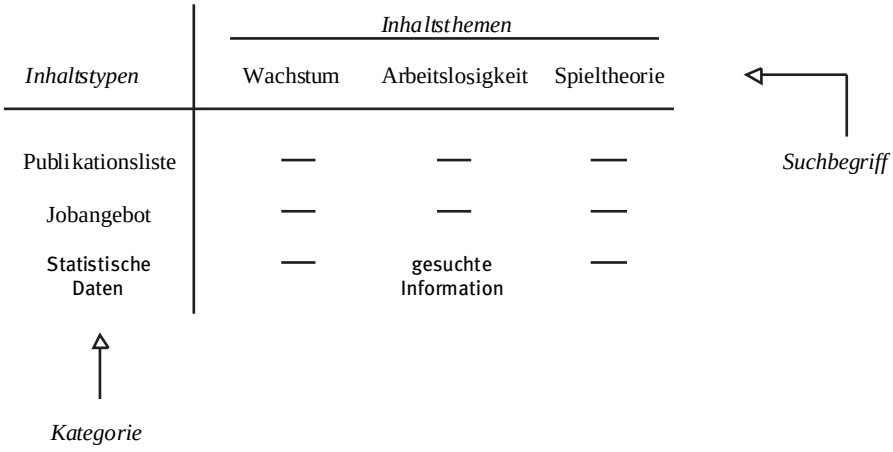



*Abb. 1–1 Rechercheziel-Matrix: Gesucht sind statistische Daten zur Arbeitslosigkeit*

## 1.1 Voraussetzungen

Die betrachtete Dokumentmenge wurde auf den Bereich der Wirtschaftswissenschaften eingeschränkt, aber die erarbeiteten Kategorien sollten nicht fachspezifisch und damit gut auf andere wissenschaftliche Bereiche übertragbar sein. Die ursprünglich geplante Übertragung auf ein zweites Fachgebiet wurde allerdings bei einer Revision des Projektplans Anfang 2001 fallen gelassen.

Eine weitere Einschränkung betraf die Sprache, in der ein Dokumenttext verfasst ist. Wir haben uns bei sprachabhängigen Merkmalen (Vorkommen bestimmter Phrasen) auf die deutsche und die englische Sprache konzentriert[1]. Eine Ausnahme bildet die Wortliste mit bekannten, ökonomischen Journalnamen, die für die Kategorie *Publikationsliste* erstellt wurde. Da wir aber auch sprachunabhängige Merkmale (z.B. Anzahl PDF-Links) verwendet haben, konnten bei anderssprachigen Dokumenten auch gute Ergebnisse erzielt werden, aber nicht für alle Kategorien.

Für das *KATI*-Projekt konnte auf verschiedene Vorarbeiten zurückgegriffen werden. Einmal wurde an die wissenschaftlichen und technischen Erfahrungen aus dem DFN-Projekt *Assoziativer Integrationsdienst* (AID) angeknüpft. Die auch in *KATI* verwendete Methode der Probit-Analyse ist dabei z.B. erprobt worden.

Auf der anderen Seite bildeten die Arbeiten zu dem seit 1998 bestehenden Suchdienst *Inomics* eine gute Grundlage für *KATI*. Die Protokolldateien gaben Hinweise auf die implizite Verwendung von Kategorienumschreibungen im Suchbegriff, über den *Inomics*-Newsletter konnte die Zielgruppe für eine Umfrage zu Kategorien erreicht werden, und Tests von frei verfügbaren Suchmaschinen-Tools und kommerziellen Softwarebibliotheken konnten auch für *KATI* genutzt werden. Nicht zuletzt stehen die Projektergebnisse sofort im Rahmen eines bewährten Suchdienstes einer schon angestammten Nutzergruppe zur Verfügung.

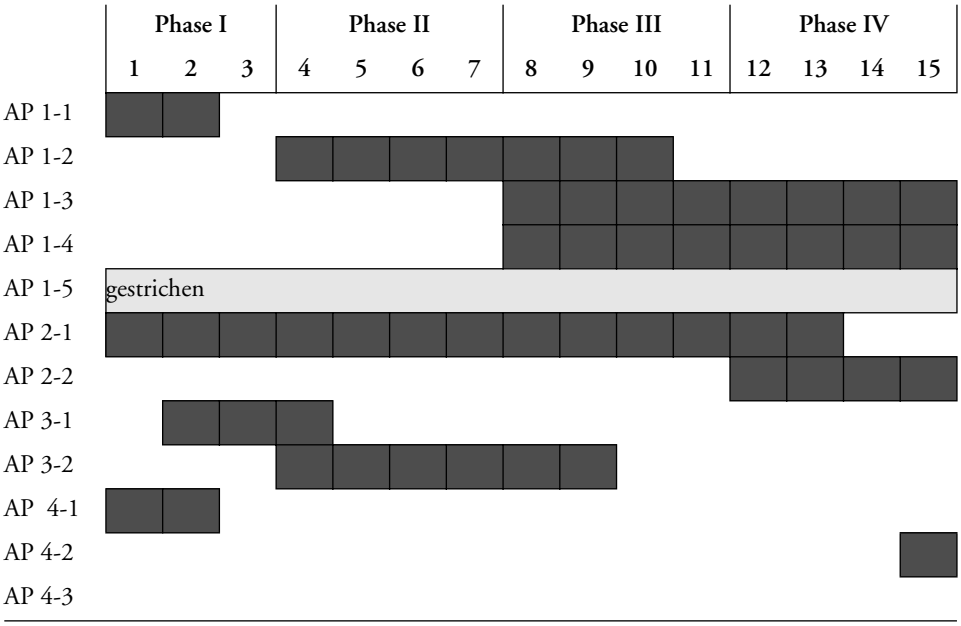
## 1.2 Planung und Ablauf des Projektes

Ursprünglich waren folgende **Arbeitspakete** mit einer Projekt-Laufzeit von 18 Monaten geplant:

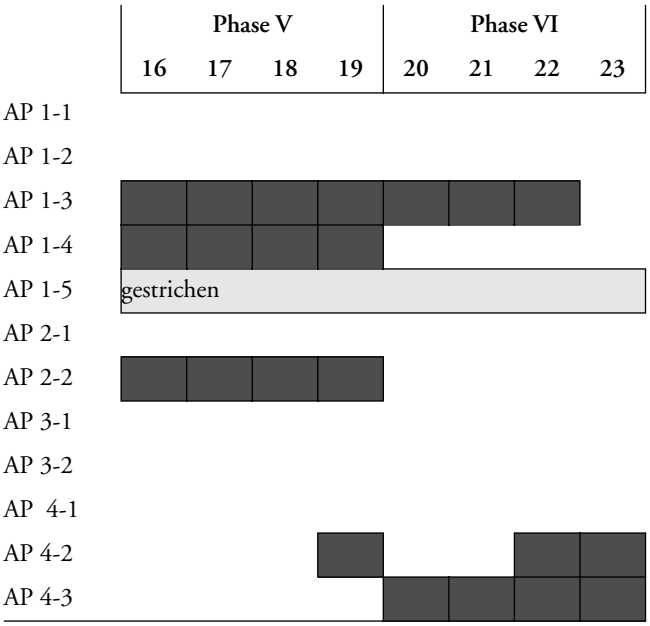
- ❑ AP 1-1 Auswertung existierender maschineller Verfahren
- ❑ AP 1-2 Konzeption einer Typenklassifikation für die Wirtschaftswissenschaften
- ❑ AP 1-3 Umsetzung, Test und Verfeinerung der Klassifikationsverfahren
- ❑ AP 1-4 Implementierung der Typenklassifikation für die Wirtschaftswissenschaften
- ❑ AP 1-5 Übertragung der Typenklassifikation auf ein zweites Themengebiet
- ❑ AP 2-1 Entwicklung eines Basisrobotersystems
- ❑ AP 2-2 Erweiterung des Robotersystems um die Kategorisierungssoftware
- ❑ AP 3-1 Konzipierung des Indexierungssystems
- ❑ AP 3-2 Implementierung des Indexierungssystems
- ❑ AP 4-1 Startkoordination
- ❑ AP 4-2 Entwicklung des Anwendungssystems
- ❑ AP 4-3 Pilotbetrieb

1. Siehe auch Kapitel “Stichpunkte zu weiteren Arbeiten” auf Seite 29. Englisch ist „die“ Wissenschaftssprache für Ökonomie, was die Einschränkung rechtfertigt.

Aufgrund der schwierigen Personalsituation im EDV-Bereich im Jahr 2000 und schließlich des Ausstiegs des Projektpartners TU Berlin wurde die Laufzeit bei Verringerung des Projektvolumens verlängert und *AP 1-5 Übertragung der Typenklassifikation auf ein zweites Themengebiet* gestrichen. Der Ablauf des Projektes bezüglich der Arbeitspakete ist in der folgenden Tabelle dargestellt:




*Tab. 1–1 Monatlicher Ablaufplan für die Arbeitspakete*




*Tab. 1–2 Arbeitspakete in der Verlängerung*

Im Projektverlauf hat sich herausgestellt, dass die Aufteilung Roboter, Kategorisierer, Indexierer bzw. Suchmaschine auf verschiedene Arbeitspakete ungünstig war, weil die Konzeption sowie die Implementierung der einzelnen Systemkomponenten sehr stark zusammenhängen. Deshalb haben wir das System und seine Architektur arbeitspaketübergreifend konzipiert und implementiert.

# 2 Projekt-Ergebnisse

Im ersten Teil dieses Kapitels werden die wissenschaftlichen Ergebnisse zur Kategorisierung dargestellt und ihre Auswirkungen in der Suchmaschine erläutert. Im zweiten Teil folgt der technische Bericht über das implementierte System.

## 2.1 Kategorisierung wissenschaftlicher Dokumente nach Inhaltstypen

### 2.1.1 Kategorien

Aufgrund unserer Umfrage-Ergebnisse aus der zweiten Projektphase haben wir folgende Kategorien ausgewählt:

- Publikation
- Statistische Daten
- Publikationsliste
- Link-Sammlung
- Institutions-Homepage
- Persönliche Homepage
- Curriculum Vitae
- Jobangebot
- Konferenz
- Kursangebot

Ein Bestand von ca. 2000 zufällig ausgewählten Dokumenten wurde von Hand in diese Kategorien eingeordnet, um als Trainingsmenge für statistische Verfahren zu dienen. Dann haben wir überprüft, welche Kategorien tatsächlich ins System aufgenommen werden sollen.

Publikationen haben wir aus mehreren Gründen zurückgestellt: Erstens stehen in *Inomics* die Arbeitspapiere und Journalartikel von *RePEc* zur Verfügung. Zweitens liegen sie oft im PDF-Format vor, während wir uns zuerst auf HTML konzentrieren wollten. Drittens findet man Publikationen leicht auch über die Kategorie Publikationsliste. Oft gehören Dokumente sogar zu beiden Kategorien.

Bei Linksammlungen haben wir feststellt, dass sie besser in die einzelnen Kategorien eingegliedert werden sollten, anstatt eine eigene zu bilden. Eine Linksammlung von Publikationen stellt natürlich eine Publikationsliste dar, ein Dokument mit Links zu Tabellen kann schon Statistischen Daten zugeordnet werden usw.

Bei Institutions-Homepages erschweren oft Frame-Sets und die sehr unterschiedliche Gestaltung die Erkennung. Persönliche Homepages und Curriculum Vitae treten oft zusammen mit einer Publikationsliste auf und sind deshalb von dieser schlecht abzugrenzen.

Da die Kategorien Konferenz und Kursangebot im Umfrage-Ranking als eher unwichtiger eingestuft wurden, haben wir sie aus Prioritätsgründen nur kurz betrachtet.

Publikationslisten, Statistische Daten und Jobangebote wurden als viel versprechend aufgenommen.

### Kategoriendefinition

Bei der Arbeit mit Kategorien hat sich gezeigt, dass von ihrer Definition die Qualität der Klassifikation entscheidend abhängt. Sie darf weder zu allgemein, noch zu spezifisch gewählt werden.

- Sind die Kategoriendefinitionen sehr allgemein, ist es schwer, die sehr unterschiedlichen Dokumente als zur selben Kategorie zugehörig zu erkennen und gleichzeitig die Fehlkategorisierung zu minimieren (Startseiten von Universitäten, Institutionen, Firmen, Organisationen, Staaten und Städten). Außerdem ist die Gefahr größer, dass sich mehrere Kategorien stark überschneiden oder einige Teilmengen von anderen darstellen, was die Abgrenzung der Kategorien erschweren kann (Linksammlung).
- Sind die Kategoriendefinitionen sehr spezifisch, sind sie gut abgrenzbar, aber oft künstlich und für den Benutzer nicht nachvollziehbar (Daten, die in einer Tabelle dargestellt sind, aber nicht Bestandteil einer Publikation sind). Außerdem werden sehr viele Dokumente aus diesem strengen Schema herausfallen und keiner Kategorie zugeordnet werden können.

Also müssen Prioritäten gesetzt werden, da sich nie völlig vermeiden lässt, dass Dokumente zu mehreren Kategorien gehören oder aus der Definition herausfallen.

In diesem Projekt haben wir uns am Ende auf drei Kategorien konzentriert, die wenig Überschneidungen aufweisen, d.h. ein Dokument gehört selten gleichzeitig zu mehreren dieser Kategorien. Die Dokumentmengen von Publikationsliste, Jobangebot und Statistischen Daten sind also theoretisch disjunkt. Damit sind sie gut voneinander abgrenzbar, aber wir konnten gleichzeitig die großzügigen Kategoriendefinitionen beibehalten, nach denen wir auch unsere Trainingsdokumente handkategorisiert hatten. Unsere Hoffnung war, auch sehr untypische Vertreter einer Kategorie zu erkennen, ohne gleichzeitig zu viele Fehler zu provozieren.

In der folgenden Aufstellung wird aufgezählt, welche Dokumente zum Beispiel als zur Kategorie *Publikationsliste* zugehörig eingeordnet werden sollen:

### Beispiel-Definition *Publikationsliste*

1. Publikationsliste einer Person (z.B. auf ihrer Home-Page),
2. Veröffentlichungen einer Institution (z.B. Working-Paper-Liste oder Report-Reihe),
3. Quellenverzeichnisse oder Referenzangaben in Publikationen,
4. Literaturlisten (reading lists), z.B. bei Kursangeboten,
5. Auflistung von Beiträgen auf einer Konferenzseite,
6. Publikationssammlung (z.B. zu einem bestimmten Thema),
7. Inhaltsverzeichnisse von Journalen oder anderen Publikationen,
8. Liste von Neuerwerbungen, z.B. von Bibliotheken,
9. Linksammlung zu Publikationslisten.

### 2.1.2 Merkmale

Sind die Kategorien und ihre Definitionen festgelegt, müssen als nächstes die Merkmale definiert werden, die zur Trennung zwischen den Kategorien beitragen sollen. Diese signifikanten Merkmale zusammen bestimmen den sogenannten Klassifikator, ein Entscheidungskriterium, nach dem ein Objekt (hier ein Internetdokument) einer Kategorie (Dokumenttyp) zugeordnet werden kann.

Auf die Sichtung von Beispiel-Dokumenten zu einer Kategorie folgte eine Auswahl vermutlich signifikanter Merkmale, die zunächst umgangssprachlich beschrieben wurden. Dann musste überprüft werden, wie gut sie sich zur maschinellen Erkennung eignen. Dabei haben wir zuerst die Merkmale ausgewählt, die sich leicht umsetzen lassen, und wollten die komplizierteren später hinzufügen. Es stellte sich aber heraus, dass komplexere Merkmale meist sehr kategorien- und fachgebietspezifisch sind, was ungünstig für die Übertragbarkeit ist. Daher lohnt sich der höhere Implementierungsaufwand komplexer Merkmale meist nicht. Wir haben uns daher für ein einfaches Konzept der **Merkmalsgenerierung** entschieden, das im folgenden Abschnitt dargestellt werden soll.

Zunächst behandelten wir die Merkmale, die sich auf Phrasen im Dokumenttext beziehen. Diese Schlüsselbegriffe wurden in **Wortlisten** gruppiert, um die Merkmalsmenge nicht zu groß werden zu lassen. Zu einer Liste wurden Phrasen zusammengefasst, die entweder Synonyme[2] darstellen oder vermutlich eine ähnliche Trennfähigkeit besitzen. Außerdem sind in manchen Fällen Singular und Plural einer Phrase in einer Liste, wenn ihr Erklärungsgehalt vermutlich derselbe ist. In anderen Fällen wird bewusst zwischen Singular und Plural getrennt. Als Beispiel für eine Wortliste soll *publ* dienen, die Synonyme für „Publikationsliste“ enthält:

*Abb. 2–1 Phrasenliste publ*

```
bibliography
inhaltsverzeichnis
list of publications
publication list
publications
publikationen
publikationsliste
literature
recent publications
references
selected publications
table of contents
veröffentlichungen
```

Eine Besonderheit in der HTML-Struktur stellen Adressen von Links (HREF) und Bildern (SRC) dar. In ihren Pfaden kann man auch nach signifikanten Zeichenketten suchen, die aber oft keine ganzen Wörter oder Phrasen darstellen (z.B. „wp“ als Verzeichnis für Working-Paper). Deshalb werden sie in **String-Listen** zusammengefaßt.

Folgendes Beispiel enthält Wörter, die oft in Pfadnamen von Home-Pages auftauchen (z.B. www.tu-berlin.de/institut/staff/mueller/home.html).

*Abb. 2–2 Stringliste staf*

```
staff
people
member
team
```

2. Dazu zählen auch die Übersetzungen von Phrasen ins Englische bzw. Deutsche, d.h. *paper* und *papier* sind in einer Wortliste.

Der **Merkmals-Parser** führt die Erkennung der Merkmale mit Hilfe der Wortlisten durch. Er zählt das Vorkommen der Phrasen in den verschiedenen Teilen der HTML-Struktur eines Dokumentes:

- im Dokument insgesamt,
- im „sichtbaren" Text[3],
- im Meta-Tag,
- im Titel,
- in Überschriften,
- im ausgezeichneten Text[4],
- im nicht-ausgezeichneten Text,
- in Tabellen und Listen,
- in alternativen Bildtexten,
- in Linktexten.

So stehen zu einer Phrasenliste zehn potentielle Merkmale zur Verfügung, aus denen die für die betrachtete Kategorie signifikanten ausgewählt werden müssen.

*Abb. 2–3 Merkmale zur Phrasenliste publ*

| Merkmal | Anzahl der Phrasen |
|---|---|
| publalle | insgesamt im Dokument |
| publtext | im „sichtbaren" Text |
| publmeta | in den Meta-Daten |
| publtitl | im Titel |
| publhead | in einer Überschrift |
| publemph | in hervorgehobenem Text |
| publnorm | im normal formatierten Text |
| publtabl | in einer Tabelle |
| publlist | in einer Listenstruktur |
| publimga | in einem Bildalternativtext |
| publlink | in einem Linktext |

Für die Phrasen aus *publ* wurde z.B. vermutet, dass ihr Vorkommen im Titel, in Überschriften und in ausgezeichnetem Text signifikante Trennfähigkeit besitzt. Außerdem erhofften wir uns eine negative[5] Signifikanz vom Vorkommen dieser Phrasen in einem Linktext, weil das eher auf das Parent-Dokument als auf eine Publikationsliste selbst hinweist.

Der Merkmals-Parser zählt die Strings in der Adresse:

- des Dokumentes (seiner URL),
- von enthaltenen Links,
- von Bildern.

Aus einer Stringliste werden also drei Merkmale generiert:

3. Unter „sichtbarem" Dokumenttext verstehen wir alle Phrasen, die über den Browser gesehen werden können. „Unsichtbarer" Dokumenttext kann dagegen nur im Quellcode der HTML-Seite angesehen werden, z.B. in Meta-Tags, Link- und Bildadressen (HREF, SRC) oder der Alternativtext zu einem Bild (ALT). Beide Textarten zusammen bilden den Gesamttext des Dokumentes.
4. Als *ausgezeichnet* gilt eine Phrase, wenn sie mit einem der folgenden Tags formatiert wurde: <b>, <big>, <blink>, , <i>, <u>, , .
5. Ein *negativ* signifikantes Merkmal weist darauf hin, dass ein Dokument *nicht* zu der Kategorie gehört.

| Merkmal | Anzahl der Phrasen |
|---|---|
| url_staf | in der Dokument-URL |
| lin_staf | in der Adresse eines Links |
| img_staf | in der Adresse eines Bildes |

*Abb. 2–4 Merkmale zu der Stringliste staf*

Schließlich gibt es aber auch Merkmale, die unabhängig von den Wortlisten sind:

| Merkmal | Anzahl |
|---|---|
| all | aller Links |
| html | Links zu HTML-Dateien |
| pdf | Links zu PDF-Dateien |
| email | Links zu E-Mail-Adressen |
| ps | Links zu Postscript-Dateien |
| spread | Links zu Spreadsheet-Dateien |
| compress | Links zu komprimierten Dateien |
| ftp | Links zu FTP-Servern |

*Abb. 2–5 Linkzähler*

| Merkmal | Anzahl der |
|---|---|
| _co_img | Bilder |
| _co_word | Wörter |
| _co_numb | Zahlen |
| _co_tabl | Tabellen |
| _co_dl | Definitionslisten |
| _co_ul | ungeordneten Listen |
| _co_pre | vorformatierten Textabschnitte (preformat) |

*Abb. 2–6 Weitere Merkmale*

Außerdem können die Merkmale durch einfache Verknüpfungen kombiniert werden, z.B. der Quotient aus den Anzahlen von Wörtern und Zahlen im Dokument.

## Klassifikator

Am Beispiel der Kategorie *Publikationsliste* soll in diesem Abschnitt verdeutlicht werden, wie wir beim Erstellen des Klassifikators vorgegangen sind.

Für die handkategorisierte Trainingsmenge wurde mit Hilfe des **Merkmals-Parsers** eine Datei mit Merkmalsvektoren für jedes Dokument erstellt. Welche Merkmale in die Vektoren eingetragen werden, hängt von den erstellten Wortlisten ab. Als Beispiel-Wortlisten sollen die schon im vorigen Abschnitt genannten betrachtet werden: die Phrasenliste *publ*[6] und die Stringliste *staf*[7].

Die Rechteckdatei mit einem Merkmalsvektor für jedes Dokument dient als Grundlage für die **Probit-Analyse** (mit dem Statistikprogramm *Stata*), bei der unsere Merkmale als Variablen behandelt werden. Betrachtet man nun die Ergebnisse der Probit-Analyse in der folgenden Tabelle, sind zwei Spalten besonders interessant.

Die **Signifikanz** einer Variable (4. und 5. Spalte) bzw. ihre Trennfähigkeit bzgl. der Kategorie war ausschlaggebend dafür, ob ein Merkmal in die Schätzgleichung aufgenommen werden soll. Je näher der P-Wert (5. Spalte) an Null, desto besser.

Der **Variablen-Koeffizient** (2. Spalte) drückt den Beitrag dieser Variable - in Abhängigkeit von ihrer konkreten Ausprägung für ein bestimmtes Dokument – z.B. 13 enthaltene PDF-Links – zu der Wahrscheinlichkeit aus, dass es sich bei diesem Dokument um eine Publikationsliste handelt.

6. Siehe Abbildung 2–1, „Phrasenliste publ," auf Seite 13.
7. Siehe Abbildung 2–2, „Stringliste staf," auf Seite 13.

*Abb. 2–7 Ergebnisse der Probit-Analyse für das Merkmal Publikationsliste*

| publ_lis | Coef. | Std. Err. | z | P>\|z\| | [ 95% Conf. | Interval ] |
|---|---|---|---|---|---|---|
| all | .0050003 | .0014175 | 3.528 | 0.000 | .0022221 | .0077784 |
| email | -.0366063 | .0176054 | -2.079 | 0.038 | -.0711123 | -.0021003 |
| compress | .4392393 | .1403374 | 3.130 | 0.002 | .164183 | .7142956 |
| pdf | .3732807 | .0566665 | 6.587 | 0.000 | .2622165 | .484345 |
| acrotext | -.2094519 | .0449567 | -4.659 | 0.000 | -.2975654 | -.1213384 |
| biblist | -.1173049 | .0213653 | -5.490 | 0.000 | -.1591802 | -.0754297 |
| bibtext | .1254032 | .0172728 | 7.260 | 0.000 | .0915491 | .1592574 |
| confnorm | -.0888486 | .0421326 | -2.109 | 0.035 | -.1714269 | -.0062703 |
| cvlink | .6102011 | .268329 | 2.274 | 0.023 | .0842859 | 1.136116 |
| homitext | -.0434763 | .0141808 | -3.066 | 0.002 | -.0712702 | -.0156825 |
| hompnorm | .2027657 | .0954128 | 2.125 | 0.034 | .0157601 | .3897713 |
| jourtext | .0794057 | .0278395 | 2.852 | 0.004 | .0248413 | .1339701 |
| jonohead | .2152838 | .0544773 | 3.952 | 0.000 | .1085103 | .3220574 |
| jookalle | .1146653 | .0459928 | 2.493 | 0.013 | .024521 | .2048095 |
| monaemph | .0251894 | .0098294 | 2.563 | 0.010 | .0059241 | .0444547 |
| pubemph | -.0847083 | .0339005 | -2.499 | 0.012 | -.151152 | -.0182646 |
| pubnorm | .0697152 | .0232338 | 3.001 | 0.003 | .0241778 | .1152526 |
| pubtitl | .8386823 | .2037242 | 4.117 | 0.000 | .4393903 | 1.237974 |
| publhead | 1.44827 | .2381544 | 6.081 | 0.000 | .9814957 | 1.915044 |
| publemph | .5613481 | .0982678 | 5.712 | 0.000 | .3687469 | .7539494 |
| publlink | -.2673178 | .0780273 | -3.426 | 0.001 | -.4202484 | -.1143872 |
| publtitl | .9629669 | .3620196 | 2.660 | 0.008 | .2534214 | 1.672512 |
| pubsemph | .2038782 | .0571182 | 3.569 | 0.000 | .0919286 | .3158277 |
| pubshead | .5714395 | .1488849 | 3.838 | 0.000 | .2796305 | .8632485 |
| staftabl | -.5089569 | .0983373 | -5.176 | 0.000 | -.7016945 | -.3162193 |
| verltext | .7131606 | .1502 | 4.748 | 0.000 | .418774 | 1.007547 |
| verllink | -1.177585 | .4173535 | -2.822 | 0.005 | -1.995582 | -.3595866 |
| url_mon | .3828344 | .1323241 | 2.893 | 0.004 | .123484 | .6421848 |
| url_publ | .2922405 | .104461 | 2.798 | 0.005 | .0875006 | .4969803 |
| url_staf | .6066788 | .1648182 | 3.681 | 0.000 | .2836411 | .9297165 |
| _co_ul | .0217716 | .0091435 | 2.381 | 0.017 | .0038507 | .0396926 |
| _cons | -1.575814 | .0584042 | -26.981 | 0.000 | -1.690284 | -1.461344 |

Wie zu sehen ist, haben sich unsere Vermutungen bezüglich *publ* bestätigt: *publtitl*, *publhead*, *publemph* und *publlink* sind sehr signifikant, und *publlink* ist wie erwartet negativ.

Bei den Link-Merkmalen stellt sich nicht nur das Vorkommen von PDF-Links als bedeutsam heraus, sondern auch das Vorkommen von Links auf komprimierte Dateien, da Veröffentlichungen auch oft „gezipt“ zum Download zur Verfügung stehen.

Interessant war das Verhalten des Merkmals *url_staf*, bei dem es um Strings geht, die

oft in Pfadnamen von Home-Pages auftauchen. Es ist auch signifikant für Publikationslisten, weil sie im Pfad meist unter einer Home-Page oder im selben Verzeichnis liegen.

Nicht unseren Erwartungen entsprach z.B. die Bedeutung von Tabellen für Publikationslisten. Die Anzahl von Tabellen scheint nicht signifikant zu sein, da die TABLE-Tags eine in fast jedem Internet-Dokument verwendete Struktur darstellen.

Mit den Koeffizienten aller für eine Kategorie signifikanten Variablen ist eine **Schätzgleichung** aufgestellt worden. Sie stellt den Klassifikator für eine Kategorie dar und wurde im Kategorisierer implementiert. Um nun ein neues Dokument zu kategorisieren, muß der Parser die konkreten Variablen-Ausprägungen erkennen, mit denen dann der Wahrscheinlichkeitswert für die Kategorienzugehörigkeit des Dokumentes errechnet wird.

## 2.2 Die Suchmaschine

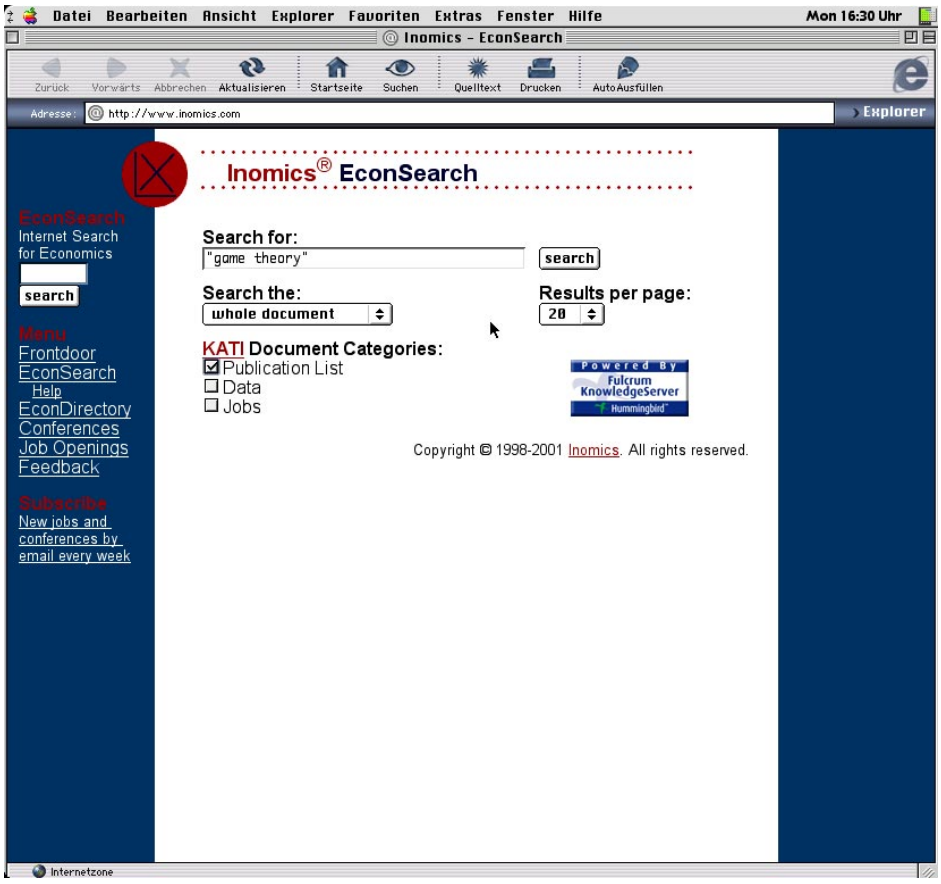
Wir haben die Projektergebnisse in den Suchdienst *Inomics* integriert. Die gewohnte Bedienungsoberfläche von *Inomics* wurde weitgehend beibehalten und durch die Kategorienauswahl erweitert (siehe Abb. 2–8). Eine Internet-Recherche kann durch Anklicken der gewünschten Kategorie(n) zusätzlich zur Eingabe der Suchbegriffe vereinfacht werden. Der Suchraum wird eingeschränkt und die Ergebnismenge überschaubarer, so dass relevante Dokumente schneller gefunden werden können.

Die Ergebnisauflistung wurde in mehren Punkten verbessert (siehe Abb. 2–9). Zusätzlich zu dem Titel des Dokumentes und seiner Beschreibung (bzw. zu den ersten Zeilen, wenn keine Beschreibung vorhanden ist) wird nun auch seine URL, seine Kategorienzugehörigkeit und die Relevanz bzgl. des Suchbegriffes ausgegeben. Für Letztere wird einfach das Vorkommen aller Phrasen des Suchbegriffes im Dokument gezählt[8].

Die Sortierung der Ergebnis-Dokumente erfolgt zuerst nach der **Kategorienzugehörigkeit** und dann nach der **Suchbegriffrelevanz**. Außerdem wurde ein Schwellwert für die Zugehörigkeit eines Dokumentes zu einer Kategorie bestimmt, nach dessen Überschreiten ein Dokument in die Ergebnismenge aufgenommen wird. In diesem Projekt war es uns wichtig, dass die von der Suchmaschine gefundenen Dokumente mit großer Wahrscheinlichkeit zur gewählten Kategorie gehören. Dafür nehmen wir in Kauf, dass Dokumente, die schwieriger zu erkennen sind, nicht in der Ergebnismenge enthalten sind, bzw. nur mit sehr niedriger Relevanz-Einstufung. Also wurde der Schwellwert für die Kategorien (100 - 1000) sehr hoch angesetzt (900). Er kann für jede Kategorie individuell eingestellt werden (aber nicht vom Suchmaschinen-Benutzer).

8. Komplexere Relevanzfunktionen lassen sich leicht beim SearchServer einstellen, siehe dazu Kapitel “Stichpunkte zu weiteren Arbeiten” auf Seite 29.

*Abb. 2–8 Suche mit Kategorienauswahl*




Gibt man als Beispiel die Suchanfrage „game theory“ ein und wählt die Kategorie *Publication List* aus, fallen mehrere Punkte an der Ergebnisliste auf:

- viele Dokumente führen weder den Suchbegriff noch eine Kategorienumschreibung im Titel trotz hoher Suchbegriffrelevanz;
- auch Dokumente mit sinnlosem oder wenig aussagekräftigem Titel (z.B. „Untitled“ oder „bib“), aber relevantem Inhalt wurden gefunden;
- neben den klassischen Publikationslisten wird die ganze Variationsbreite der Kategoriendefinition[9] abgedeckt, es gibt z.B. auch Konferenzprogramme mit Links zu den Beiträgen oder Neuerwerbungslisten von Bibliotheken usw.

9. Siehe “Beispiel-Definition Publikationsliste” auf Seite 12.

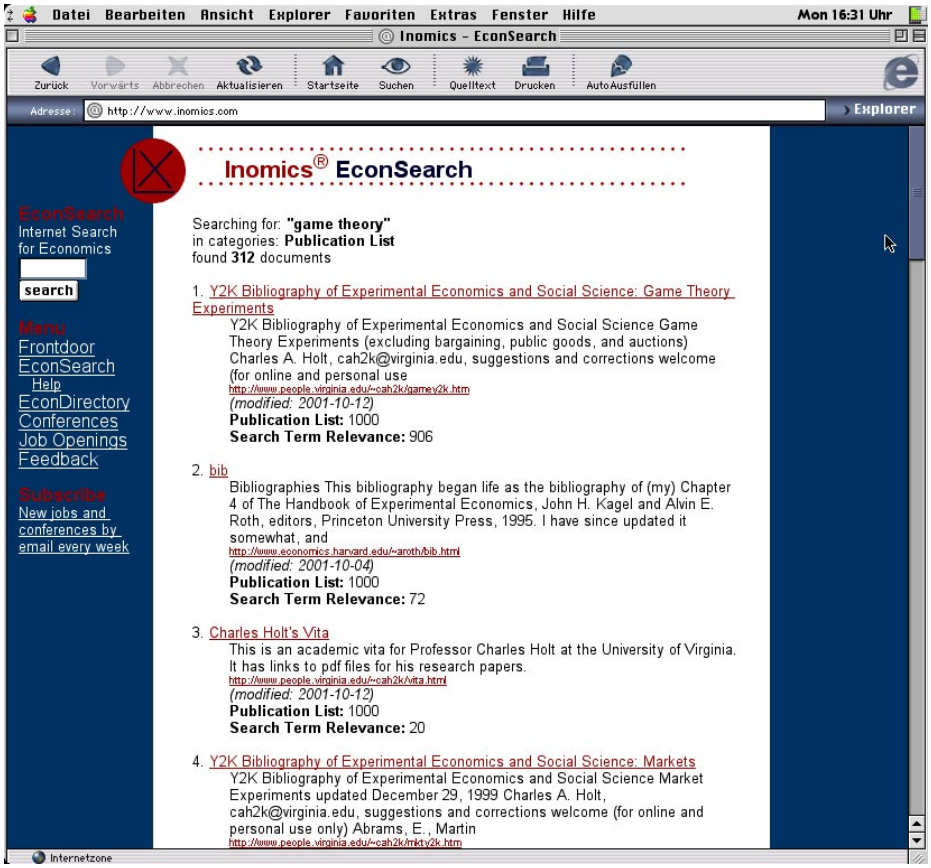



*Abb. 2–9*
*Ergebnisliste*

Zusammenfassend läßt sich sagen, dass *KATI* im Vergleich zu herkömmlichen Suchmaschinen auch unkonventionelle Dokumente findet. Außerdem werden durch die Spezialisierung auf ökonomische Dokumente seltener unrelevante Dokumente anderer Wissensgebiete gefunden.

# 3 Technischer Bericht

## 3.1 Erläuterung des Datenmodells

Ein Internet-Dokument hat verschiedene Eigenschaften, die sich wie folgt einteilen lassen:

- **Eigenschaften des Servers**, der ein Dokument bereitstellt:
  - Übertragungsprotokoll (z.B. HTTP, FTP),
  - Hostname (seine Adresse, z.B. *www.berlecon.de*),
  - Serviceport[10].

Durch diese drei Eigenschaften wird der Server, auf dem ein Dokument zu finden ist, eindeutig festgelegt. Sie finden sich in der Tabelle 'hosts' der ***KATI*-MySQL-Datenbank** wieder. In dieser Tabelle wird jedem Server eine eindeutige numerische ID zugeordnet.

- **Eigenschaften des Dokumentes** auf diesem Server:
  - Pfadname (z.B.: '/presse/index.html'),
  - Inhaltsformat (z.B. HTML),
  - Datum der letzten Modifikation,
  - Inhalt des Dokumentes in seiner ursprünglichen (Source) und in reiner Textform.

Diese Eigenschaften werden in *KATI* in verschiedener Form gespeichert. Der Pfadname und das Inhaltsformat werden zusammen mit weiteren Angaben in zusätzlichen MySQL-Tabellen gespeichert. Aufgrund der großen Anzahl der zu verarbeitenden Dokumente existieren hierfür mehrere Tabellen[11]. Jedem Dokument wird eine in der jeweiligen Tabelle eindeutige ID zugeordnet.

Die **Dokument-Source** wird unter einem Dateinamen gespeichert, der sich aus ServerID und DokumentID ergibt. Dateisysteme sind in der Lage, für jede Datei das Datum der Erstellung sowie das Datum der letzten Modifikation zu vermerken. Diese Eigenschaft wird genutzt, um das **Modifikationsdatum** des Dokumentes zu erhalten.

Das Ziel einer Suchmaschine ist es, Dokumente für Anwender durchsuchbar zu machen. Für diese Zwecke wurde das Softwarepaket *Fulcrum Searchserver* erworben. Es bietet Werkzeuge, um aus verschiedenen Dateiformaten den Text zu extrahieren und erstellt einen **Index** über alle Dokumente.

10. Jeder IP-Adresse können verschiedene Portnummern zugeordnet werden, dies bietet die Möglichkeit, unter einer Adresse verschiedene Dienste anzubieten. Für gewöhnlich benutzt z.B. HTTP den Port 80, HTTPS den Port 443 und FTP Port 21.
11. Die Zuordnung eines Dokumentes zu einer Tabelle ergibt sich aus der Operation „ServerID *modulo* Tabellenanzahl“.

## 3.2 Der Weg eines Dokumentes in KATI

Zuerst wird die URL eines Dokumentes in die Datenbank eingefügt. Die Dokumente eines Servers werden zusammen bearbeitet. Wenn der Server das nächste Mal besucht wird, versucht der **Roboter** (*crawler*) das Dokument zu laden. Ist dies erfolgreich geschehen, wird das Dokument vom **Kategorisierer** und danach vom **Indexierer** verarbeitet.

In einem vom Benutzer festzulegenden Intervall wird der Dokumentbestand aufgefrischt, d.h. der Roboter versucht, modifizierte Dokumente neu zu laden. Treten bei der Übertragung Fehler auf (z.B. wenn das Dokument nicht mehr unter der bekannten Adresse existiert), werden die Dokumentinformationen (nach mehreren erfolglosen Versuchen) aus der Datenbank entfernt.

## 3.3 Aufbau der Software

Für den Anwender sind drei Scripte (im *bin*-Verzeichnis) entscheidend:

- *crawler* hat die Funktion, Dokumente aus dem Internet zu laden und aus HTML-Dateien weiterführende Links zu extrahieren,
- *kategorisierer* ordnet den Dokumenten ihre Kategorienzugehörigkeiten zu,
- *indexer* fügt Dokumente dem Index hinzu oder löscht sie aus diesem. Außerdem entfernt er Dokumente, die Fehler aufweisen, aus dem System.

Alle drei Scripte werden als Daemon gestartet. Der *crawler* arbeitet so lange, bis er vom Benutzer beendet wird. Kategorisierer und Indexierer terminieren selber, wenn keine Dokumente mehr zu bearbeiten sind.

Außerdem können die Scripte durch ein INT- bzw. TERM-Signal beendet werden. Hierzu steht das Script 'stop' im *bin*-Verzeichnis zur Verfügung, das allen Prozessen, deren lock file im *lock*-Verzeichnis auftaucht, das TERM-Signal sendet.

Für den Betrieb des Systems können deshalb ein oder mehrere *crawler* permanent laufen. Kategorisierer und Indexierer sollten per Cron-Job, z.B. nachts, hintereinander gestartet werden.

Ein Dokument duchläuft folgende Bearbeitungsschritte:

Ist auf dem Server eine neuere Version als lokal vorhanden, so wird diese ins *source*-Verzeichnis übertragen, danach kann der Kategorisierer das Dokument bearbeiten. Ist dies erfolgreich abgeschlossen, kann der Indexierer die neuen Daten in den Index eintragen.

Der Benutzer muss im Modul *KATI* ein Intervall angeben, in dem die Dokumente besucht werden sollen.

### Erläuterung der Dokument- und Hostklassen

Die Hostklassen stellen die Funktionalitäten bereit, um Hosts in der Datenbank zu verarbeiten.

- *Host* ist die namensgebende Oberklasse,
- *Host::Base* enthält die Datenfelder und die zugehörigen Zugriffsmethoden,
- *Host::Database* stellt das Interface für die *MySQL*-Tabellen bereit sowie Methoden zum Lesen und Schreiben von Datensätzen.

Die Dokumentklassen stellen Methoden bereit, um Metadaten einzelener Dokumente zu lesen, zu bearbeiten und zu speichern.

- *Document::Base* enthält die Basisdatenfelder und Zugriffsmethoden,
- *Document::Database* erweitert die Documentklasse um Methoden zum MySQL-Datenbankzugriff,
- *Document::LocalFile* stellt Methoden bereit, um Dateien zu lesen und schreiben,
- *Document::Remote* enthält Methoden für den *crawler*, um Dokumente von entfernten Servern zu holen,
- *Document::SimpleHTML* stellt HTML-Parser-Methoden bereit und wird vom *crawler* zum Linkextrahieren benutzt,
- Document::Category ist die Dokumentklasse für den Kategorisierer,
- Document::Index ist die Dokumentklasse für den Indexierer.

### Erläuterung der zusätzlichen Klassen

- *KATI* enthält benutzerdefinierte Variablen,
- *Daemon* macht aus einem Prozess einen Daemon,
- *LogFunc* exportiert Funktionen, um auf STDOUT und STDERR in einem einheitlichen Format zu schreiben,
- *MyUserAgent* lädt die Klasse *LWP::UserAgent* und überschreibt einige Methoden,
- *Parser* enthält die Parser-Funktionalität für den Kategorisierer,
- *MyDBI* emuliert einige Methoden der DBI-Klasse für den *Searchserver*.

### Software die nicht weitergegeben werden kann:

- *ssdrv.so* ist der Treiber für *MyDBI* <=> *Searchserver*.

Dieser Treiber wurde unter Zuhilfenahme des Tools *SWIG* als Perl-Extension in der Sprache C programmiert und wird vom Modul *MyDBI* mittels *DynaLoader* (Perl-Standardbibliothek) eingebunden.

Der Treiber darf aus lizenzrechtlichen Gründen nicht weiterverbreitet werden, deshalb soll hier seine Funktionalität kurz erläutert werden:

Folgende Funktionen stellt der Treiber bereit:

- *bool setup_connect()* - initialisiert die Datenbankverbindung (databasehandle) mittels der von *Searchserver* benötigten Umgebungsvariablen, bei Erfolg wird true zurückgegeben. Dieses Handle ist im Treiber *modulglobal* definiert, deshalb kann für jeden Prozess nur ein Database-Handle bestehen.
- *bool prepare_stmt ()* - initialisiert ein Statement-Handle, gibt true bei Erfolg zurück, kann nur nach erfolgreichem *setup_connect* ausgeführt werden, genau wie das Database-Handle *modulglobal*, d.h. es existiert jederzeit nur ein Statement-Handle.
- *int simple_stmt(char *sqlstr)* - führt ein einfaches SQL-Statement wie z.B. insert, update oder delete aus, kann nach einem erfolgreichen *prepare_stmt* ausgeführt werden, gibt bei einem insert die ID des neuen Datensatzes, sonst die Anzahl der modifizierten Datensätze zurück.
- *bool execute_stmt(char *sqlstr)* - führt ein SQL-select-Statement aus, kann nach einem erfolgreichen *prepare_stmt* ausgeführt werden, gibt true bei Erfolg zu-

rück
- *char **fetch_result_row()* - gibt die aktuelle Zeile eines Resultates zurück und ein leeres Array, wenn keine weiteren Zeilen mehr existieren; kann nach einem erfolgreichen *execute_stmt* ausgeführt werden
- *void finish_stmt()* - das Statement-Handle wird auf die NULL-Werte zurückgesetzt, Speicher wird freigegeben
- *char *errstr()* - kann nach dem Auftreten eines Fehlers aufgerufen werden, um den Fehlertext zu erhalten
- *int get_numcols()* - liefert nach der erfolgreichen Ausführung eines select-Statements die Anzahl der Spalten des Ergebnisses

Füer den Betrieb des Indexierers ist entweder ein solcher Treiber bereitzustellen oder der Indexierer ist den Bedürfnissen anzupassen. Wird ein anderes Indizierungssystem verwendet, ist auch die Klasse *Document::Index* an dieses System anzupassen.

## 3.4 Installations- und Bedienungsanleitung

- Ein **Verzeichnis** für die Programmdateien, z.B. /usr/local/kati/, anlegen und die Dateien dorthin kopieren,
- ein Source-Verzeichnis für die heruntergeladenen Dateien erstellen,
- in der Datei lib/**Kati.pm** die Variablen anpassen,
- darauf achten, dass für das Lock-Verzeichnis, die Logdateien und das Source-Verzeichnis Schreibrechte bestehen,
- die **MySQL-Tabellen** anlegen:
  - Hierfür existiert im Verzeichnis *support* die Datei *kati.mysql,* die mit dem Befehl „mysql [-u mysqladmin] < kati.mysql“ die Datenbank *kati* sowie die benötigten Tabellen erzeugt.
  - Möchte man zur Dokumentverwaltung mehr oder weniger Tabellen als per Default verwenden, kann diese Datei angepasst werden. Außerdem muss in diesem Fall in der Datei *Kati.pm* die Variable *$Kati::num_tables* angepasst werden.
- **StartURLs** in die Datenbank einlesen, für ökonomische Dokumente sind bereits einige Dokumente in der Datenbank enthalten,
- Die Politik für die **Link-Weiterverfolgung** kann für jeden Host eingestellt werden. Dafür wird die Summe aus 4 Werten gebildet und in der 'hosts'-Tabelle im Feld 'follow' eingetragen[12]:
  - sollen URLs nur im gleichen oder in darunterliegenden Verzeichnissen wie das Dokument verfolgt werden, so ist der Summe 1 hinzuzufügen,
  - sollen alle URLs auf diesem Server verfolgt werden, so ist der Summe 2 hinzuzufügen,
  - sollen cgi-URLs verfolgt werden, muss 4 addiert werden,
  - sollen auch URLs, die auf andere Server verweisen, verfolgt werden, muss man 8 addieren.

12. Per Default wird dort eine 1 eingetragen (von jedem Dokument dieses Servers werden nur URLs ins gleiche Verzeichnis verfolgt).

### Hinzufügen neuer Dokumentkategorien:

Sollen neue Kategorien hinzugefügt werden, so ist zuerst für jede Kategorie in allen doc-Tabellen das entsprechende Feld einzufügen. Empfohlen wird das Format: 'smallint unsigned not null'. Dann ist im Verzeichnis *support/coeff* eine gleichnamige Datei zu erstellen, welche die zum Beispiel mit *Stata* errechneten Koeffizienten enthält. Dabei muss darauf geachtet werden, dass die in dieser Datei angegebenen Merkmale existieren.

### Bildung von Merkmalsnamen:

Für jede im Verzeichnis *support/phrasen* angelegte **Phrasenliste**[13] werden mehrere Merkmale erzeugt. Der Merkmalsname ergibt sich, indem die unten genannten Schlüssel an den Dateinamen <*name*> angehängt werden:

| | |
|---|---|
| in sichtbarem Text | text |
| in normalem Text | norm |
| im Titel | titl |
| in Überschriften | head |
| in hervorgehobenem Text | emph |
| im Text von Links | link |
| in Listen | list |
| in Meta-Tags | meta |
| im ALT-Attribut von IMG | imga |
| in Tabellen | tabl |
| alle Vorkommen | alle |

Die Stringlisten in den Dateien[14] im Verzeichnis *support/strings* werden in folgenden Feldern gesucht:

| | |
|---|---|
| in der URL des Dokumentes | url_<*name*> |
| in Linkadressen (HREF) | lin_<*name*> |
| in Bildadressen (SRC) | img_<*name*> |

13. Der Dateiname darf nicht länger als 4 Zeichen sein.
14. Auch dieser Dateiname darf nicht länger als 4 Zeichen sein.

# 4 Übertragung auf ein neues Fachgebiet

Die Übertragung auf ein weiteres Fachgebiet wurde zwar im Rahmen einer Revision des Projektplanes gestrichen, trotzdem soll im folgenden Abschnitt kurz umrissen werden, wie dabei vorzugehen bzw. was zu beachten wäre.

Folgendes Vorgehen hat sich im KATI-Projekt bewährt:

- **Kategorien** eventuell anpassen,
- **Trainingsdokumente** mit Hilfs-Tool von Hand kategorisieren,
- Zyklus:
  - Merkmale: **Wort- und String-Listen** für jede Kategorie erstellen (eventuell neue Link- und Tag-Zähler),
  - Parser erstellt **Merkmalsvektoren** für Trainingsdokumente (Rechteckdatei),
  - Probit-Analyse erzeugt **Schätzgleichungs-Koeffizienten** für den Kategorisierer,
  - Internet-Dokumente laden und kategorisieren sowie den **Index** erstellen, und schließlich
  - das Ergebnis mit **Test**-Suchanfragen überprüfen.

Folgende Stichpunkte fassen einige Problempunkte zusammen, die bei der Kategorisierung von Internet-Dokumenten beachtet werden sollten:

*Kategorien*

- Kategorien dürfen weder zu allgemein noch zu spezifisch definiert werden.
- Internet-Dokumente gehören meist zu mehreren Kategorien gleichzeitig. Das kann die Abgrenzung der Kategorien erschweren.

*Merkmale*

- Komplexere Merkmale sind meist sehr kategorien- und fachgebietspezifisch. Das ist ungünstig für die Übertragbarkeit, daher lohnt der höhere Implementierungsaufwand meist nicht.
- Bei der Beschränkung auf einfache Merkmale kann die Merkmalsmenge sehr groß werden.

*HTML-Struktur*

- Tags werden in den HTML-Dokumenten gelegentlich nicht oder falsch benutzt (z.B. kein Titel bzw. kein sinnvoller Eintrag; für Überschrift <b> + <big> anstatt <h1> ).
- Standards haben sich noch nicht durchgesetzt (Meta-Tag).
- Formatierungen werden sehr unterschiedlich benutzt (Tabellen auch mit <pre>, als Liste oder ohne Tags; <table> wird auch für Navigationsleisten benutzt) .

*Sonstige technische Probleme*

- Dokumente verlieren schnell ihre Aktualität, aber neues Indizieren ist sehr zeit- und rechenintensiv.
- Umleitungen und vorübergehende Nichterreichbarkeit von Internet-Servern müssen behandelt werden.

- Frame-Sets bestehen aus mehreren Dokumenten, die verfolgt werden müssen.
- Die Verwendung von Bildern in Internet-Dokumenten nimmt immer mehr zu, aber sie können nicht so einfach analysiert werden wie Text.

# 5 Stichpunkte zu weiteren Arbeiten

Im Laufe des Projektes sind mehrere interessante und wünschenswerte Erweiterungen aufgetaucht, die möglicherweise in thematisch verwandten DFN-Projekten oder in Folgeprojekten bearbeitet werden könnten. Diese sollen hier als Anregung für Dritte aufgeführt werden.

Vielversprechend erscheint beispielsweise der Einbau von Merkmalen, die sich auf die Kategorien selbst und/oder das Parent-Dokument beziehen:

- *Frühere Kategorisierung eines Dokumentes*: Interessant, da sich häufig nur der Inhalt eines Dokumentes ändert, aber nicht sein Typ (neue Einträge in einer Publikationsliste) oder zu der bestehenden Kategorie kommt eine neue hinzu (eine Home-Page wird durch eine Publikationsliste ergänzt).
- *Kategorien bzw. Merkmale des Parent-Dokumentes*: Auch diese Variante ist interessant, da z.B. das Parent-Dokument von einer Publikationsliste meisteine Home-Page ist. Das zeigt sich beispielsweise in der Phrase im Text des Links, der zu aktuellem Dokument führt. Schwierigkeit dabei ist, dass ein Dokument mehrere Parents mit verschiedenen Linktexten besitzen kann.
- *Einordnung in andere Kategorien*: Dies kann sowohl als positives als auch als negatives Merkmal bezüglich einer Kategorie betrachtet werden. So ist eine Home-Page oft gleichzeitig eine Publikationsliste, während sich aber Jobs und Daten sich eher ausschließen. Schwierigkeit dabei sind zyklische Abhängigkeiten.

Außerdem könnte die Erkennung weiterer Kategorien verbessert werden (z.B. Home-Page, Konferenz- und Studieninfos). Dazu müßten allerdings die Definitionen der jeweiligen Kategorie in Hinblick auf die gemachten Erfahrungen[15] überarbeitet werden und Trainingsdokumente entsprechend neu handkategorisiert werden.

Sicherlich wäre es auch interessant, neben HTML weitere Dokumentformate zu indizieren und kategorisieren (z.B. PDF, RTF). Der Searchserver hält dazu auch noch sogenannte Textreader für verschiedene Formate bereit.

Die Übertragung auf ein weiteres Fachgebiet wurde zwar in einer späteren Projekt-Phase gestrichen, trotzdem wurde dieses Ziel nie ganz aus den Augen verloren und spielte bei vielen Designentscheidungen weiterhin eine Rolle. Diese Vorbereitungen zusammen mit den in den vorigen Kapiteln geschilderten Projekt-Erfahrungen[16], müßten eine Übertragung auf ein neues Fachgebiet vereinfachen.

Die erfolgreiche Kategorisierung von Dokumenten in einer weiteren Sprache, z.B Spanisch oder Französisch, könnten wahrscheinlich durch Ergänzen der Wortlisten mit den entsprechenden Phrasen der jeweiligen Sprache erreicht werden.

15. Siehe “Kategoriendefinition” auf Seite 12.
16. Siehe Kapitel “Übertragung auf ein neues Fachgebiet” auf Seite 27.