DIW Berlin

Institut für Wirtschaftsforschung

# STATLINK

# Wirtschaftsinformationen für Forschung und Lehre

ANHANG

## 1. DOKUMENTENPOOL DES STATLINK-PROJEKTS ..... 6

**1.1. DIW VGR 1960-1998 (ESVG79) ..... 6**
1.1.1. DIW VGR ab 1991 (ESVG95) ..... 6
1.1.2. DIW Bauvolumensrechnung 1960-91 ..... 6
1.1.3. Ifo Konjunkturtest ..... 7
1.1.4. Regionalstatistische Datenbanken (Widab-Brandenburg) ..... 11
1.1.5. EUROSTAT LINDA, STATISBUND und andere ..... 15

## 2. TECHNISCHE SYSTEM ÜBERSICHT ..... 16

**2.1. Einführung ..... 16**

**2.2. Datenbasis der Datenbank ..... 21**
2.2.1. DBMS ..... 21
2.2.2. Datenbankdesign ..... 22
2.2.3. Datenbanktabellen ..... 24

**2.3. Alphabetische Übersicht über die verwendeten Datenbanktabellen ..... 40**

**2.4. Dämon ..... 47**
2.4.1. Systemarchitektur ..... 47
2.4.2. Datenbankserver (Dämon) ..... 47
2.4.3. Sitzung (Session) ..... 49
2.4.4. Protokoll: Anfragen an den Server ..... 51
2.4.5. Protokoll: Antworten des Servers ..... 55
2.4.6. Technische Details des Dämons ..... 60
2.4.7. Erweiterung des Standard-Dämons ..... 60
2.4.8. Erweiterung des Standard-Dämons, Protokollerweiterung ..... 61

## 3. IR-SUBSYSTEM ..... 68

**3.1. Allgemeines ..... 68**

**3.2. Einführung in das Information Retrieval ..... 69**
4.3.3 Bewertungskriterien im Information Retrieval ..... 71
3.2.2. Konkrete Information-Retrieval-Systeme ..... 72
3.2.3. Forschungsaspekte im Umfeld von Information Retrieval ..... 72
3.2.4. Persistenz im Information Retrieval ..... 73
3.2.5. Einbindung mehrerer verteilter Information-Retrieval-Systeme ..... 73
3.2.6. Erweiterte Information-Retrieval-Anforderungen ..... 73
3.2.7. Zusammenfassung ..... 73

**3.3. IR Funktionen von STATLINK ..... 73**

**3.4. Terming, Parsing und Filtern ..... 76**
3.4.1. Synonyme ..... 76
3.4.2. Terme ..... 77
3.4.3. Stoppworte ..... 77

**3.5. Stemming ..... 78**
3.5.1. Schlagworte ..... 79

**3.6. Voting ..... 79**

3.6.1. Übersicht ... 79
3.6.2. Votingverfahren ... 80
3.6.3. Algoritmische Methoden ... 83

**3.7. Tagging (Markieren) ... 84**

**3.8. Implementierung der Methoden im Dämon ... 84**
3.8.1. Dämonprogramm irdaemon ... 84
3.8.2. Modul prefs ... 85
3.8.3. Modul util ... 85
3.8.4. Modul stem ... 85
3.8.5. Modul vote ... 85
3.8.6. Modul tag ... 86
3.8.7. Modul main ... 87

**3.9. Entwicklungsumgebung ... 87**
3.9.1. Web-basierte Entwicklungstools ... 87
3.9.2. Kommandozeilen-Tools ... 91

**3.10. Perl-Sourcen des erweiterten Dämons (IR-Dämon) ... 91**
3.10.1. Dämonprogramm irdaemon ... 91
3.10.2. Modul prefs ... 95
3.10.3. Modul util ... 99
3.10.4. Modul stem ... 102
3.10.5. Modul vote ... 103
3.10.6. Modul tag ... 120
3.10.7. Modul main ... 127

**3.11. Vollständiges Protokoll einer Dämonanfrage mit Debug-Informationen ... 129**

**3.12. Der STATFINDER in Aktion ... 158**

**3.13. Nutzung der STATFINDER-Technologie in weiteren Web-basierten Lösungen ... 163**
3.13.1. Die Teleforum-Marktzugangsinitiative ... 163
3.13.2. Das DIW als Partner für "InfoSchul 3"- Projekte ... 164

## 4. WEITERENTWICKLUNGEN DER INTERNETLÖSUNGEN: STATLINK ... 166

**4.1. Die STATLINK-Technik der kategorisierten Suche ... 166**

**4.2. Verbesserung der Retrievalmöglichkeiten durch Freitextsuche ... 171**

## 5. LITERATURNACHWEISE ... 179

## 6. INTERNETADRESSEN (URLS) ... 180

# 1. Dokumentenpool des STATLINK-Projekts

Folgende Bestände gehören zum Bestand der für das Projekt verwendeten Domäne:

## *1.1. DIW VGR 1960-1998 (ESVG79)*

Volkswirtschaftliche Gesamtrechnung
Zeitreihendokumente
2 Kategorien
Bestandssprache deutsch

Die vierteljährliche volkswirtschaftliche Gesamtrechnung (VGR) ist das wichtigste Analyse- und Prognose-Instrument des Konjunkturforschers. Das DIW stellt dafür schon seit vielen Jahren – in Ergänzung zu den Daten des Statistischen Bundesamtes (StaBu) – ein geschlossenes Kreislaufsystem zur Beobachtung von Entstehung, Verteilung, Verwendung und Umverteilung des gesamtwirtschaftlichen Einkommens (des Bruttoinlandsprodukts) in Form langer Zeitreihen auf vierteljährlicher Basis zur Verfügung. Die allgemeinen Ausführungen zum Konzept und zu den empirischen Grundlagen der volkswirtschaftlichen Gesamtrechnung sollen dem Nutzer helfen, sein Verständnis für die VGR zu verbessern und die Arbeit mit ihr zu erleichtern

### 1.1.1. DIW VGR ab 1991 (ESVG95)

Volkswirtschaftliche Gesamtrechnung
Zeitreihendokumente
2 Kategorien,
Bestandssprache deutsch

Seit April 1999 ist die VGR des StaBu - und damit auch die des DIW - vollständig auf das Europäische System Volkswirtschaftlicher Gesamtrechnungen (ESVG95) ausgerichtet.

Eine ausführliche Darstestellung der Konzepte und Umsetzungen findet sich in: Karin Müller-Krumholz: Methodische und empirische Grundlagen der vierteljährlichen volkswirtschaftlichen Gesamtrechnung des Deutschen Instituts für Wirtschaftsforschung. In: Diskussionspapier Nr. 206 des DIW Berlin. Berlin 2000, online verfügbar unter:

http://www.diw.de/deutsch/publikationen/diskussionspapiere/jahrgang2000/dp206.pdf

### 1.1.2. DIW Bauvolumensrechnung 1960-91

Die vierteljährliche Berechnung des Bauvolumens nach Baubereichen und Produzentengruppen wird vom DIW im Auftrage des Bundesministeriums für Wirtschaft durchgeführt. Die laufende Fortschreibung innerhalb des Berichtsjahres dient zur Beurteilung der konjunkturellen Situation im Bausektor. Unvermeidlich sind jedoch nachträgliche Korrekturen, in denen Ergebnisse aus den Jahreserhebungen für das Baugewerbe und andere am Bau beteiligte Sparten einbezogen werden.

Die Datenbank enthält Ergebnisse der Berechnungen für das frühere Bundesgebiet im Zeitraum 1960 bis 1991, wobei alle zwischenzeitlichen Revisionen berücksichtigt

worden sind. Allerdings wurde hier - anders als bei den neueren Veröffentlichungen für die Jahre nach 1991 - keine Umgruppierung entsprechend der neuen Wirtschaftszweig-Systematik (WZ 93) vorgenommen (sie betrifft vor allem Umgruppierungen zwischen den Bereichen Bauhaupt- und Ausbaugewerbe). Im Interesse der Vergleichbarkeit mit den neueren Daten für Deutschland (ab 1991) ist jedoch die Darstellung der realen Entwicklung auf die Preisbasis von 1995 umgestellt worden. (Vgl. hierzu auch: Zur Entwicklung der Bauwirtschaft in West- und Ostdeutschland. In: Wochenbericht 42/1 997 des DIW, Bearb.: B. Bartholmai.)
Einen ausführlichen Überblick über die Berechnungsgrundlagen und -methoden des Bauvolumens vermittelt die Veröffentlichung: Zeitreihen für das Bauvolumen in der Bundesrepublik Deutschland. DIW-Beiträge zur Strukturforschung, Heft 154/1994.

*Gegenstand und allgemeine Zielsetzung*

*Veröffentlichungen*
Einen ausführlichen Überblick über die Berechnungsgrundlagen und -methoden des Bauvolumens vermittelt die Veröffentlichung: Zeitreihen für das Bauvolumen in der Bundesrepublik Deutschland", DIW-Beiträge zur Strukturforschung, Heft 154/1994.

### 1.1.3. Ifo Konjunkturtest

Zeitreihendokumente
2 Kategorien
Bestandssprache deutsch

Die wichtigste Ergänzung zur VGR stellt der Konjunkturtest des ifo Instituts dar. Der Konjunkturtest enthält monatliche Angaben zu folgenden Themen:

Geschäftslage
- Beurteilung (Saldo)
- für 6 Monate erwartete Änderung (Saldo)
- dass., saisonbereinigt (Saldo)

Geschäftsklima:
- Beurteilung (Saldo)
- für 6 Monate erwartete Änderung (Saldo)
- dass., saisonbereinigt (Saldo)

Pläne und Erwartungen der Betriebe
- Produklion, für 3 Monate erwartete Änderung (Saldo)
- dass., saisonbereinigt(Saldo)
- Erzeugerpreise, für 3 Monate erwartete Änderung (Saldo)
- Exportgeschäft, für 3 Monate erwartete Änderung (Saldo)
- dass., saisonbereinigt (Saldo)
- Beschäftigte, für 3 Monate erwartete Änderung (Saldo)
- dass., saisonbereinigt (Saldo)

Auftragsbestand
- Veränderungen gegen Vormonat (Saldo)
- dass., saisonbereinigt; (Saldo)
- Beurteilung (Saldo)
- dass., saisonbereinigt; (Saldo)
- in Produktionsmonaten (Monate)
- dass., saisonbereinigt; (Saldo)

Nachfragesituation
    Veränderungen gegen Vormonat (Saldo)
    dass., saisonbereinigt; (Saldo)
Produktionstätigkeit
    Veränderungen gegen Vormonat (Saldo)
    dass., saisonbereinigt; (Saldo)
Kapazitätsauslastung
    % der betrieblichen Vollauslastung
    dass., saisonbereinigt; (Saldo)
Kapazitäten - Einschätzung
    für 12 Monate (Saldo)
    dass., saisonbereinigt; (Saldo)
Produktionsbehindernde Faktoren
    Produktionsbehinderung bei % der Betriebe (%)
    Auftragsmangel bei Prozent der Betriebe (%)
    Arbeitskräftemangel bei % der Betriebe (%)
    Materialknappheit bei % der Betriebe (%)
    Kapazitätsengpässe bei % der Betriebe (%)
Lagerhaltung

Der Branchenbaum für den ifo Konjunkturtest umfasst derzeit. folgende Wirtschaftszweige:

Verarbeitendes Gewerbe ohne Nahrungs- und Genußmittel
Vorprodukte (Gliederung nach dem Verwendungszweck)
Investitionsgüter (Gliederung nach dem Verwendungszweck)
Gebrauchsgüter (Gliederung nach dem Verwendungszweck)
Verbrauchsgüter (Gliederung nach dem Verwendungszweck)
Konsumgüter (Gliederung nach dem Verwendungszweck)
Grundstoff- und Produktionsgütergewerbe
Mineralölverarbeitung
Gewinnung und Verarbeitung von Steinen und Erden
H.v. Zement
Ziegelei
H.v. Betonerzeugnissen
H.v.Schleifmitteln
NE-Metallerzeugung, NE-Metallhalbzeugwerke
Gießerei
Eisen-, Stahl- und Tempergießerei
NE-Metallgießerei
Eisen-u.Stahlerzeug.,-bearbeitung (einschl. Gießerei)
NE-Metallindustrie (einschl. NE-Metallgießerei)
Chemische Industrie
Anstrichstoffe u. Verdünnungen
Bit.Dach-u.Dichtungsmat.,Bautenschutzm.u.ä.
Seifen,Wasch-,Reinigungs-u.Körperpflegemittel
H.v. Chemiefasern
Holzbearbeitung
Sägewerke und Hobelwerke
H.v. Halbwaren aus Holz
Zellstoff-, Holzschliff-, Papier- und Pappeerzeugung

Herstellung von Papier
Herstellung von Pappe
Gummiverarbeitung
Investitionsgüter produzierendes Gewerbe
Stahlverformung
H.v.Gesenk-,lei.Freiformschmiedest.,schw.Preßt.etc.
Stahlmetallbau und Leichtmetallbau
Stahl- und Leichtmetallkonstr. (ohne Grubenausbau)
Kessel- und Behälterbau
Bauelem. aus Stahl und Leichtmetall
Maschinenbau (einschließlich Ackerschlepper)
H.v.landwirtschaftlichen Maschinen,Ackerschleppern
H.v.landwirtschaftl.Maschinen ohne Ackerschlepper
H.v. Metallbearbeitungsm.,Maschinen- u.Präzisionswerkzeuge
H.v.Werkzeugmaschinen
Maschinen- und Präzisionswerkzeugeeuge
H.v. Textil- und Nähmaschinen
H.v.Masch.f.d.Nahr.u.Genußmittelind.,chem.u.verw.Ind.
H.v.Masch.f.d.Nahrungsmittel- u.verw.Ind.
H.v.Einzelapp.,Masch.u.Anl.f.d.chem.u.verw.Ind.
H.v.Masch.f.d.Verarbeitung v.Gummi u.Kunststoff
H.v.Hütten-,Walzwerkseinr.,Bergw.-,Gießereimaschinen,Hebezeuge..
Hütten- und Walzwerkseinrichtungen
Maschinen und Einrichtungen für den Bergbau
H.v.Hebezeugeeugen u.Fördermitteln
H.v. Bau-, Baustoff- und ähnlichen Maschinen
Herstellung v.Baumaschinen
Herstellung v.Baustoff- u.ähnl.Maschinen
H.v. Zahnrädern,Getrieben,Lagern,Antriebselementen
Herstellung v.Zahnrädern und Getrieben
Herstellung v.Wälzlagern
H.v.Maschinen f.d.Holz-u.Papierindustrie
Herstellung v.Holzbe-u.-verarbeitungsmaschinen
Herstellung v.Papier- und Druckereimaschinen
Sonstiger Maschinenbau
Herstellung v.Dieselmotoren
Herstellung v.Flüssigkeitspumpen
Herstellung v.ölhydr.Anl.u.pneumat. Steuerungen
Herstellung v.Lufttechn.u.Klimaeinrichtungen
Herstellung v.Trocknungsanlagen
Herstellung v.Armaturen
Straßenfahrzeugbau (ohne Ackerschlepper)
H.v.Kraftwagen und deren Teilen
H.v. Kraftwagen und Kraftwagenmotoren
H.v.Teilen für Kraftwagen und Kraftwagenmotoren
H.v.Personenkraftwagen einschl.Kombiwagen
H.v.Nutzkraftwagen
Schiffbau
Elektrotechnik
H.v.Ger.u.Einr.f.d.Elektrizitätserz.,-umw.u.-vert.
H.v.Geräten u.Einricht.f.d.Elektrizitätserzeugung

H.v.Geräten u.Einricht.f.d.Elektrizitätsverteilung
H.v. elektrischen Leuchten und Lampen
Herstellung von Elektrohaushaltsgeräten
H.v.Fernmelde-,Meß- u.Regel-,elektromedizin.Ger,Bauel.
Nachrichtentechnik
Elektrische Meß-, Prüf-, Regel-u.Steuerungsgeräte
H.v.Elektromedizin.Ger.u.Einrichungen
Aktive und passive Bauelemente
H.v.Rundf.-,Fernseh-,phonotechn.Ger.etc
Elektrotechnik, vorwiegend Investitionsgüter
Elektrotechnik, vorwiegend Verbrauchsgüter
Feinmechanik, Optik, Herstellung von Uhren
Optik (ohne Augenoptik, Foto- und Kinotechnik)
Augenoptik
H.v.Foto-, Projektions- und Kinogeräten
Feinmechanik (ohne Medizin- und Orthopädiemechanik)
H.v.Medizin- und orthopädiemechan. Erzeugnissen
H.v. Uhren
H.v. Eisen-, Blech- und Metallwaren
H.v. Werkzeugen
H.v.Heizgeräten und Kochgeräten
H.v.Haushalts-u.Küchengeschirr a.Metall
H.v.Stahlblecherzeugn.f.d.Zentralheizungenbed.
H.v. Möbeln aus Metall
H.v. Feinstblechpackungen
H.v. Schlössern und Beschlägen
H.v. Schneidwaren und Bestecke
H.v.Büro- und Schreibgeräten
H.v. Metallkurzw.,leichten Preß-,Zieh-,u.Stanzteilen
Büro- u. Informationstechnik
Verbrauchsgüter produzierendes Gewerbe
Spielwaren
Feinkeramik
Herstellung und Verarbeitung von Glas
H.v. Hohlglas, technischem Glas
Holzverarbeitung
H.v.Bauelementen aus Holz (ohne Fertigteileeilbauten)
H.v.Holzmöbeln
H.v.Holzmöbeln (Oh.Polstermöbelöbel)
H.v.Zimmer- u.Küchenmöbeln
H.v.Büro- u.Schulmöbeln
H.v.Laden- u. Betriebseinrichtungen
H.v.Polstermöbelöbeln
H.v. Holzverpackungsmitteln und -lagerbehältern
H.v.Pinseln, Besen u.Bürsten
Papierverarbeitung und Pappeverarbeitung
H.v.Tapeten,Verpackungsmitteln a.Papier,Pappe
H.v.Verpackungsmitteln a.Papier und Pappe
H.v. Schreibwaren,Bürobed.a.Papier u.Pappe
Druckerei, Vervielfältigung
Druck v.Geschäftspap.,Verpack.u.Werbemat.

Druck v.Büchern, Zeitungen, Zeitschriften
H.v. Kunststoffwaren
H.v.Halbzeug aus Kunststoff
H.v.Einzelteilen aus Kunststoff
H.v.Bauelementen aus Kunststoff
H.v.Verpackungsm.,Lager-,Transportbeh.a.Kunststoff
Ledererzeugung
Lederverarbeitung
Lederverarbeitung (ohne Schuhe)
H.v. Schuhen
Textilgewerbe
Spinnerei und Weberei
Wollspinnerei
Baumwollspinnerei
Texturiererei
Wollweberei a.n.g.
Baumwollweberei a.n.g.
H.v. Möbelstoff und Dekorationsstoff
H.v. Teppichen
Wirkerei, Strickerei
H.v.gewirkter und gestrickter Bekleidung
Veredlung von Textilien
Bekleidungsgewerbe
H.v. Herren-, Damen- und Kinderwäsche
H.v. Miederwaren
Herren- und Knabenoberbekleidung
Damen- und Mädchenoberbekleidung
Nahrungs- und Genußmittelgewerbe
Ernährungsgewerbe
Mahlmühlen und Schälmühlen
H.v. Backwaren (ohne Dauerbackwaren)
Obstverarbeitung und Gemüseverarbeitung
Fischverarbeitung
Brauerei
H.v. Spirituosen
Mineralbrunnen, Limonaden
Milch und Milcherzeugnisse
Schlachthäuser (o.komm.), Fleischverarbeitung
Tabakverarbeitung

### 1.1.4. Regionalstatistische Datenbanken (Widab-Brandenburg)

Ein Projekt im Auftrag des Ministeriums für Wirtschaft, Mittelstand und Technologie Brandenburg.

#### 1.1.4.1. Industriestatistik nach Bundesländern

Zeitreihendokumente
3 Kategorien (Regionen, Themen, Branchen)
Bestandssprache deutsch

Die Industriestatistik enthält die Ergebnisse aus den monatlichen Erhebungen für Betriebe des Bereiches Bergbau, Gewinnung von Steinen und Erden sowie des verarbeitenden Gewerbes nach der seit 1995 gültigen (seit 1996 eingesetzten) "Klassifikation der Wirtschaftszweige, Ausgabe 1993 (WZ 93)". Diese Klassifikation ersetzt die bisherige "Systematik der Wirtschaftszweige, Fassung für die Statistik im Produzierenden Gewerbe (SYPRO). Sie wurde aufgrund einer EU-Verordnung eingeführt und bildet die nationale Fassung der europäischen Wirtschaftszweigklassifikation NACE Rev. 1. Eine Vergleichbarkeit der statistischen Ergebnisse der Jahre vor 1995 mit denen der Klassifikation nach NACE ist nicht gegeben.

Die monatlichen Berichte für Betriebe im verarbeitenden Gewerbe (sowie Bergbau und Gewinnung von Steine und Erden) ist Teil der Statistiken im verarbeitenden Gewerbe. In ihnen werden Daten erhoben, die primär für kurzfristige Konjunkturanalysen genutzt werden. Darüber hinaus dienen die Daten jedoch auch der Untersuchung von längerfristigen strukturellen Veränderungen sowohl innerhalb des verarbeitenden Gewerbes als auch zwischen den Sektoren der gesamten Volkswirtschaft. Weiterhin fließen die Ergebnisse des Monatsberichts in die Berechnung von Konjunkturindikatoren (z. B. Auftragseingangsindizes) ein. Die Statistik liefert somit vor allem Informationen für wirtschafts- und arbeitsmarktpolitische Entscheidungen.

Zum Berichtskreis gehören die Betriebe des verarbeitenden Gewerbes von Unternehmen des produzierenden Gewerbes mit im allgemeinen 20 und mehr Beschäftigten sowie Betriebe des verarbeitenden Gewerbes mit im allgemeinen 20 und mehr Beschäftigten von Unternehmen der übrigen Wirtschaftsbereiche. Die Berichterstattung schließt verarbeitende Betriebe des Handwerks ein.

Aus Gründen der besseren Repräsentation ist bei folgenden Branchen der Steine- und Erdenindustrie sowie des Ernährungsgewerbes die Erfassungsgrenze auf 10 und mehr tätige Personen begrenzt worden:

14.11 Gewinnung von Naturwerksteinen und Natursteinen a. n. g.
14.21 Gewinnung von Kies und Sand
15.20 Fischverarbeitung
15.31 Verarbeitung von Kartoffeln
15.32 Herstellung von Frucht- und Gemüsesäften
15.33 Verarbeitung von Obst und Gemüse a. n. g.
15.71 Herstellung von Futtermitteln für Nutztiere
15.72 Herstellung von Futtermitteln für sonstige Tiere
15.91 Herstellung von Spirituosen
15.92 Alkoholbrennerei
15.97 Herstellung von Malz
15.98 Mineralbrunnen, Herstellung von Erfrischungsgetränken
26.63 Herstellung von Transportbeton

Für den Wirtschaftszweig 20.10 Säge-, Hobel- und Holzimprägnierwerke gilt als Erfassungsgrenze ein Jahreseinschnitt von mindestens 5000
Kubikmeter Rohholz (im Festmaß).

Grundsätzlich nicht einbezogen werden im Rahmen dieser Statistik die öffentlichen Versorgungsbetriebe sowie die Betriebe des Baugewerbes.

#### 1.1.4.2. Baugewerbe nach Bundesländern

Zeitreihendokumente
3 Kategorien (Regionen, Themen, Branchen)
Bestandssprache deutsch

Die monatliche Berichterstattung im Bereich Vorbereitende Baustellenarbeiten, Hoch- und Tiefbau einschließlich der vierteljährlichen Statistik über den Auftragsbestand sowie die vierteljährliche Statistik im Bereich Bauinstallation, sonstiges Baugewerbe und Vermietung von Baumaschinen und -geräten mit Bedienungspersonal dient vor allem der kurzfristigen Beurteilung der konjunkturellen Lage des Baumarktes. Sie liefert wichtige Unterlagen für die Arbeit der gesetzgebenden Körperschaften, der Bundes- und Landesregierung, der Bau- und Handwerksverbände sowie der Kammern und ist somit eine unentbehrliche Grundlage für zahlreiche Entscheidungen auf dem Gebiet der gesamten Wirtschaftspolitik, insbesondere im Bereich der Bauwirtschaft.

Die im Baugewerbe (Bauhauptgewerbe) monatlich erhobenen Daten beziehen sich zunächst auf die meldepflichtigen Betriebe (Melde-Betriebe) der jeweiligen Bundesländer, erfaßt durch die Statistischen Landesämter. Auf Grundlage einer Totalerhebung (Durchführungszeitpunkt im Juni jedes Jahres) werden Faktoren ermittelt, die für die im Monatsbericht des Baugewerbes erhobenen Kennziffern eine Aufschätzung auf die Ergebnisse für alle Betriebe ermöglichen (davon ausgeschlossen sind der Auftragseingang und der Auftragsbestand). Aus den Ergebnissen der Totalerhebung werden die zum Monatsbericht meldepflichtigen Einheiten für ein Jahr neu festgelegt. Die Festlegung des neuen Berichtskreises der Betriebe von Unternehmen mit 20 und mehr Beschäftigten erfolgt ab Berichtsmonat Oktober eines jeden Jahres. Diese Totalerhebung im Baugewerbe erfaßt alle bauhauptgewerblichen Betriebe von Unternehmen des Baugewerbes, des üblichen Produzierenden Gewerbes und von Unternehmen anderer Wirtschaftsbereiche sowie die Arbeitsgemeinschaften des Baugewerbes, soweit sich die Tätigkeit auf das Inland bezieht.

*Ausbaugewerbe*
Die vierteljährliche Erhebung im Aubaugewerbe dient ebenfalls der Beurteilung der konjunkturellen Lage des Baumarktes, außerdem liefert sie Informationen über die Struktur dieses Wirtschaftsbereiches. Sie stellt Unterlagen für die Arbeit der gesetzgebenden Körperschaften, der Bundesregierung sowie der Landesregierungen, der Bau- und Handwerksverbände sowie der Kammern zur Verfügung und ist eine Grundlage für zahlreiche wirtschaftspolitische Entscheidungen, insbesondere im Bereich der Bauwirtschaft.

Diese Erhebung umfaßt in der Regel die ausbaugewerblichen Betriebe mit 20 und mehr Beschäftigten des Ausbaugewerbes sowie die ausbaugewerblichen Betriebe mit 20 und mehr Beschäftigten anderer Wirtschaftsbereiche.

Die Unterabteilung "*Ausbaugewerbe*" des Produzierenden Gewerbes umfaßt 14 Wirtschaftszweige, deren Tätigkeit vor allem darin besteht, Ausbauarbeiten und

entsprechende Reparatur- und Unterhaltungsarbeiten vorzunehmen. In diesen Zweigen können auch Institutionen vorkommen, die nicht oder nicht überwiegend an Bauten tätig sind, z. B. Autoglaser und Schiffsmaler. Für die Zuordnung der Betriebe zu denWirtschaftszweigen ist die überwiegend ausgeübte Tätigkeit maßgebend.

Sozialwesens; für Freizeit, Erholung, Sport; der Ver- und Entsorgung sowie des Verkehrs- und Nachrichtenwesens.

(Entnommen aus den statistischen Berichten EII1 / EIII1 - m 2/98, FII2 - m 02/98 und FII1 - j/96 / FII3 - j/96 des Statistischen Landesamtes Berlin.)

#### 1.1.4.3. Arbeitsmarkt nach Bundesländern

Zeitreihendokumente
2 Kategorien (Regionen, Arbeitsmarkt)
Bestandssprache deutsch

Die vorliegenden Daten zum Arbeitsmarkt wurden aus den Informationen der Landesarbeitsämter zusammengestellt. Die Maßnahmen zur produktiven Arbeitsförderung (MpA) wurden in Berlin und Brandenburg durch die Förderung von Strukturanpassungsmaßnahmen (SAM) abgelöst.

Definitionen der in der Arbeitsmarktstatistik verwendeten Begriffe:

(Entnommen aus dem Bericht des Statistischen Landesamtes Brandenburg AVI 7-m07/98.)

#### 1.1.4.4. Erwerbstätigkeit nach Wirtschaftsbereichen

Zeitreihendokumente
2 Kategorien (Regionen, Erwerbstätigkeit nach Wirtschaftsbereichen)
Bestandssprache deutsch

Auf Landesebene wird die Erwerbstätigkeit arbeitsteilig von den Statistischen Landesämtern berechnet. Die Erstellung der Kreisergebnisse erfolgt nach einem gemeinsamen methodischen Konzept und wird in jedem Landesamt separat durchgeführt. Die Hauptquellen zur Berechnung der Erwerbstätigen sind die aus dem Meldeverfahren der Sozialversicherung gewonnenen Zahlen über sozialversicherungspflichtig beschäftigte Arbeitnehmer, die auf Meldungen von Unternehmen und Betrieben beruhenden Ergebnisse der Bereichsstatistiken sowie die Angaben des als Haushaltsbefragung angelegten Mikrozensus.
Länder- und Kreisergebnisse aus dem Meldeverfahren zur Sozialversicherung werden von der Bundesanstalt für Arbeit vierteljährlich in wirtschaftsfachlicher Gliederung jeweils etwa mit sechsmonatigem Abstand zum Berichtsstichtag ermittelt. Für die neuen Bundesländer stehtdiese Datenquelle seit dem Berichtsjahr 1992 zur Verfügung und geht in die Berechnung ein.

Ergebnisse nach Ländern und Kreisen aus amtlichen Statistiken sind nur für das Produzierende Gewerbe und den öffentlichen Dienst verfügbar. Das Produzierende Gewerbe wird zu einem großen Teil durch monatliche Erhebungen abgedeckt, wobei die großen Unternehmen vollständig einbezogen sind. Für die regionale Arbeitsmarktbeobachtung erfolgt die Zuordnung eines Erwerbstätigen zu einem Wirtschaftszweig nach dem wirtschaftlichen Schwerpunkt des Betriebes, in dem er tätig ist. Den vorliegenden Berechnungen liegt das Inlandskonzept zugrunde, das bedeutet, daß alle Erwerbstätigen, die bei inländischen Wirtschaftseinheiten beschäftigt sind, berücksichtigt werden.

Für die neuen Bundesländer erfolgt mit der Darstellung der Entwicklung der Erwerbstätigkeit auch eine Abbildung des sich in den letzten Jahren vollziehenden Strukturwandels der Volkswirtschaft, der in besonderem Maße durch die Privatisierungen der ehemals staatlichen Unternehmen und den völligen Umbau der Wirtschaftsstruktur gekennzeichnet ist. Dieser Strukturwandel ist eine der Hauptursachen für die starken Schwankungen in der Erwerbstätigenentwicklung, die sich erst allmählich abschwächen wird. Bei den Erwerbstätigenzahlen für 1995, 1996 und 1997 ist zu beachten, daß die Bundes- und Länderrechnung derzeit nicht übereinstimmt. Bei der Berechnung der Produktivitätskennziffern sind die durch den „Arbeitskreis VGR der Länder" auf die Bundeswerte abgestimmten Daten bevorzugt worden. Die Zahl der Erwerbstätigen ist für wirtschafts- und sozialpolitische Zwecke eine wichtige Größe.

#### 1.1.4.5. Wertschöpfung nach Wirtschaftsbereichen

Zeitreihendokumente
2 Kategorien (Regionen, Wertschöpfung nach Wirtschaftsbereichen, Bruttoinlandsprodukt BIP)
Bestandssprache deutsch

Erläuterungen zur Wertschöpfung
In der Volkswirtschaftlichen Gesamtrechnung (VGR) werden die Entstehung und Verwendung des Inlandsprodukts, die Verteilung, Umverteilung und Verwendung der Einkommen (Drei-Seiten-Rechnung) nachgewiesen.
Die Gliederung der Ergebnisse erfolgt anhand der Systematik der Wirtschaftszweige, Ausgabe 1979, Fassung für Volkswirtschaftliche Gesamtrechnungen, Juni 1985.

Die Berechnung der Ergebnisse der VGR erfolgt vierteljährlich und in tieferer Untergliederung jährlich, wobei sich durch die regelmäßig notwendigen Revisionen Abweichungen der Angaben zu vorangegangenen Veröffentlichungen ergeben können.

(Nach den statistischen Berichten PI 2 - j 94, PI 1 - j 94 des Statistischen Landesamtes Brandenburg sowie des Statistischen Landesamtes Berlin.)

### 1.1.5. EUROSTAT LINDA, STATISBUND und andere

EUROSTAT-Linda, die europäische Produktionssstatistik, ist tief gegliedert nach Regionen, Branchen und Themen. Es handelt sich bei der benutzten Version um einen älteren, nicht aktueller Referenzbestand, hauptsächlich verwendet zum Testen

der multilingualen Fähigkeiten der Teilmodule und zur Erweiterung des Schlagwortkatalogs für die Erprobung der Technik der Freitextsuchanfrage.
STATISBUND ist die Zeitreihendatenbank des Statistischen Bundesamtes. Im Rahmen dieses Projekts wurde der Subset der Industriestatistik nach WZ93 für Erprobungszwecke sowie zur Erweiterung der Regionenhierarchie für die regionalstatistische Datenbank "Industriestatistik" (hier wurde Deutschland, Deutschland Ost und West als Ergänzung der vorhandenen Industriestatistiken der Bundesländer mit integriert).verwendet.
OECD-MEI sind die Main Economic Indicators der OECD (Organisation for Economic Co-operation and Development), eine Datenbank mit Monatsstatistiken für die OECD Länder und Ländergruppen wie z.B.die Europäische Gemeinschaft, branchengegliedert mit den wichtigsten Indikatoren der Wirtschaftsentwicklung.

# 2. Technische System Übersicht

## *2.1. Einführung*

Der Kern des STATLINK Projekts ist eine Zeitreihendatenbank mit Wirtschaftsdaten. Alle Werkzeuge und Clients dieses Projektes dienen dem Zugriff auf diese Datenbank.

Das Projektergebnis beinhaltet Programme, Skripte, Dämonen, Datenbanktabellen und Rohdaten- Dateien.

Zur einfacheren Orientierung folgt ein grafischer Überblick über alle im STATLINK-Projekt benutzten Komponenten.

Im oberen Teil der folgenden Grafik werden die Clients aufgeführt, die als Ausgangspunkt für das STATLINK-Projekt dienten, oder die Neuentwicklungen innerhalb des STATLINK-Projekt sind. Es gibt drei grundsätzlich verschiedene Arten von Clients:

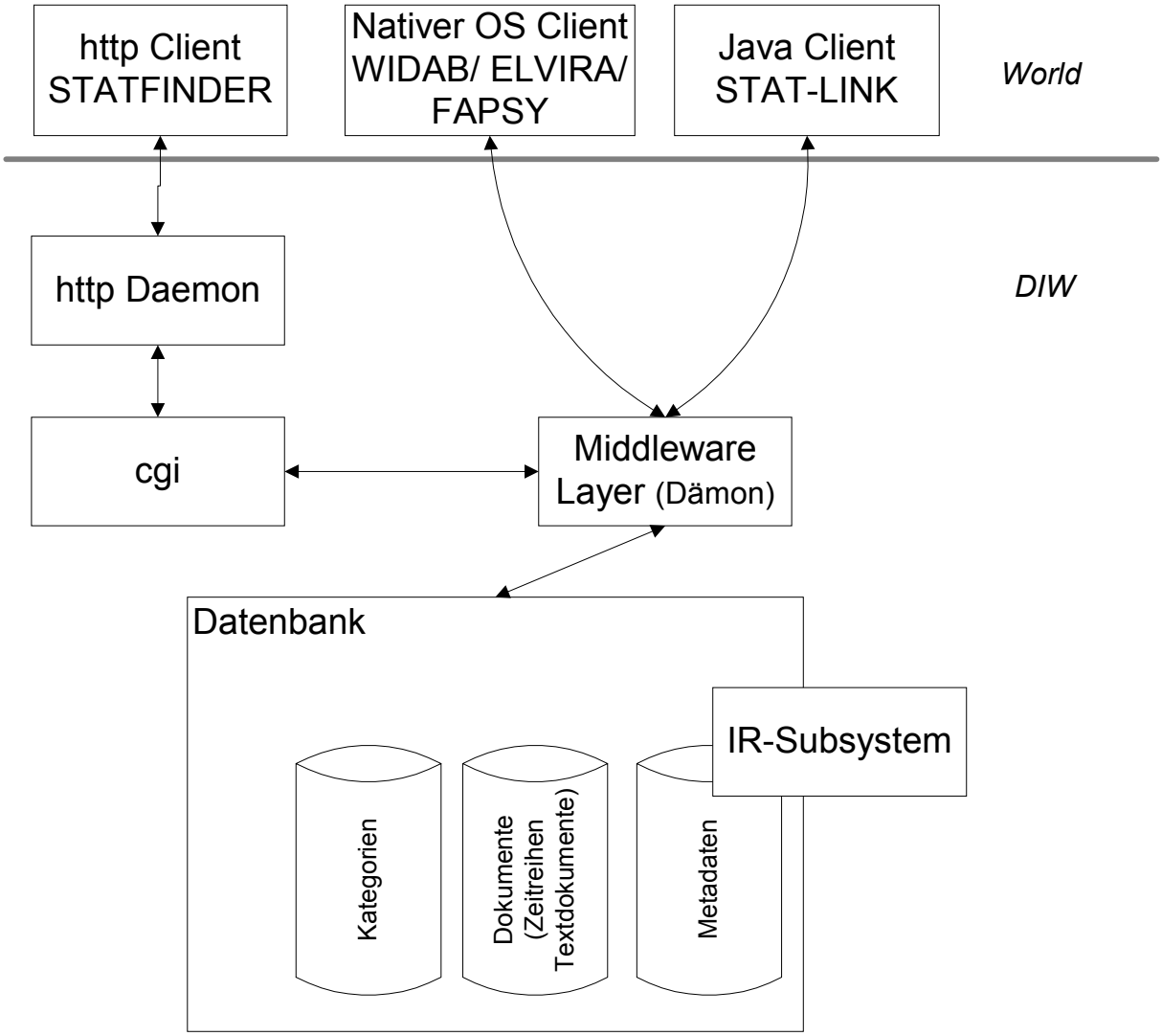



**Abbildung 1: Überblick über die Kommuniktionswege**

Die Clients kommunizieren mit den serverseitigen Komponenten, um Daten aus der Datenbank abzurufen. Zu diesem Zweck wurde eine standardisierte http-Kommunikation eingerichtet (reine Webbrowser-Version) und eine weitere, TCP/IP-basierte Kommunikation mit einem eigenen Protokoll.

Die Datenbank enthält verschiedene Informationseinheiten, wie Kategorien, Dokumente (hauptsächlich Zeitreihen, als ein kleinerer Bestand Texte) und Meta-Daten.

Die Datenbank wurde um ein komplexes Information-Retrieval (IR) System ergänzt.

- Http-Clients, die eine Webbrowser-basierte Anwendung darstellen (STATFINDER)
- Native Clients für Windows NT/2000 und Windows 9x (WIDAB, ELVIRA, FAPSY)
- Java Clients (eine komplette Java Applikation die mit Hilfe der Java-Runtime-Umgebung (JRE), Version 1.3 funktioniert (STATLINK).

Die nachfolgende Abbildung zeigt eine Übersicht der verschiedenen Kommunikationsmöglichkeiten der Komponenten des STATLINK Projekts bei Verwendung verschiedener Clients (Frontends).

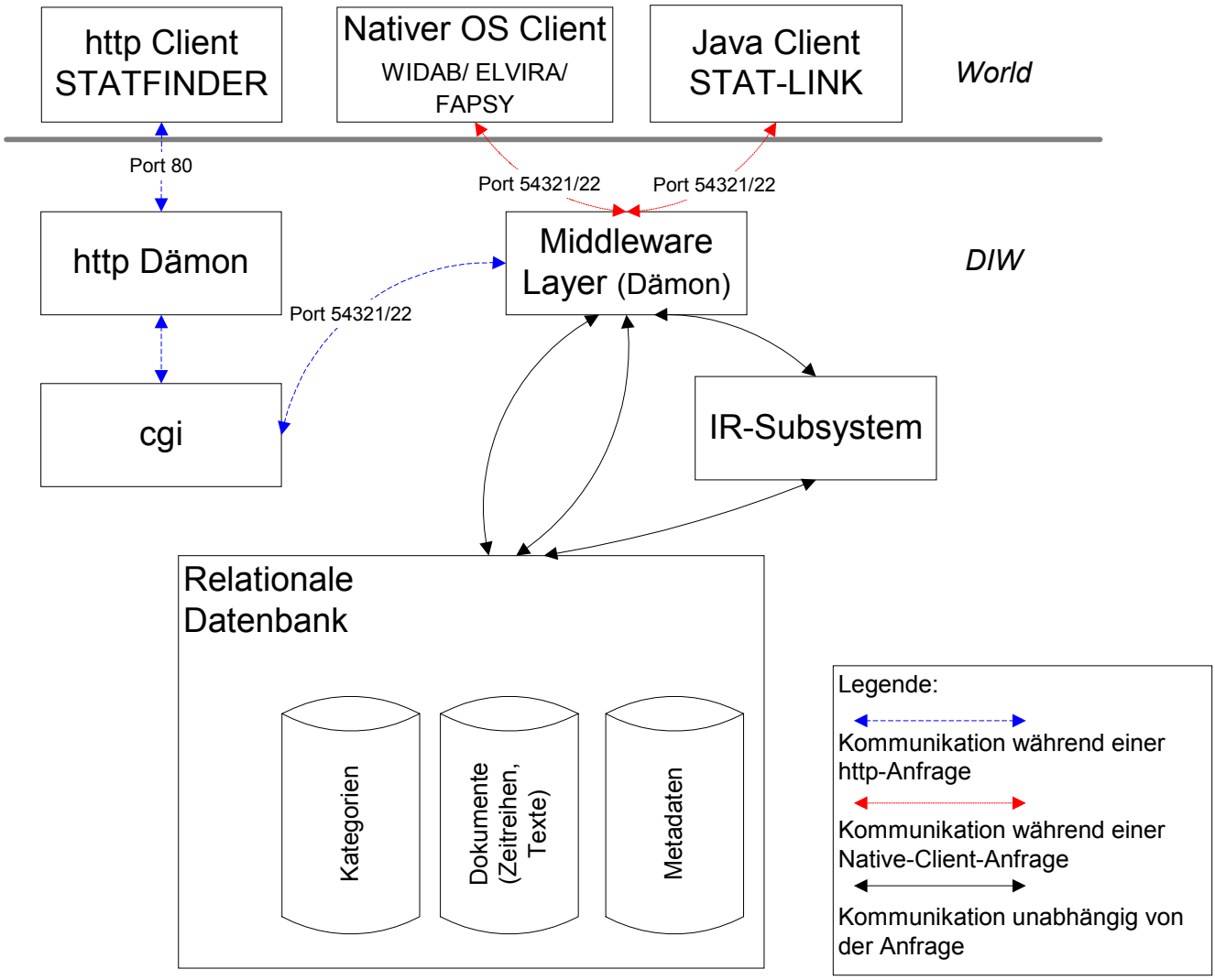



**Abbildung 2: Übersicht über die Kommunikationswege bei Anfragen verschiedener Clients**

Im Falle einer HTTP-Anfrage wird diese über den Standardport 80 und das HTTP-Protokoll abgewickelt. Jeder beliebige Webserver (am DIW getestet mit dem Netscape Enterprise 3.6 und dem IPlanet 4.1 Server) kann die Kommunikation zwischen dem Browser und der Datenbank abwickeln. CGI-Skripte und die nativen OS-Clients kommunizieren über unterschiedliche Ports mit dem IR Subsystem - abhängig davon, ob IR relevante Daten benötigt werden oder nicht. Im STATLINK gibt es zwei Dämonen für diesen Zweck: einer handelt die Anfragen mit, der andere erledigt Anfragen ohne IR-Inhalte.

## *2.2. Datenbasis der Datenbank*

### 2.2.1. DBMS

Das Data Base Management System ist OPEN INGRES Ver. 1.2 / 2.0.
Es läuft unter SCO UNIX Ver. 5.05. Das Entwicklungssystem läuft auf einem Dell Poweredge, das Produktionssystem auf einer IBM RS6000-F50 unter AIX 4.3.1.

### 2.2.2. Datenbankdesign

Das für das Projekt benutzte Datenbankdesign wurde bereits in anderen Projekten erprobt und ist entsprechend ausgereift. Unsere Zeitreihendatenbank ist für grosse numerische Datenmengen konzipiert worden. Die Datenbank muss einerseits mit einem grossen Datenpool umgehen, andererseits muss sie leicht benutzbar und wartbar sein. Und natürlich muss sie Daten in annehmbar kurzer Antwortzeit liefern können. Dies führte zu einer durchsatzorientierten Redundanz des Datenbankkonzepts.
Das Datenbankmanagementsystem ist ein relationales; Daten werden in Tabellen gespeichert. In der Datenbank werden hauptsächlich einfache Datentypen wie Real, Integer und String benutzt. Allerdings werden bereits die erste grossen abstrakten Datentypen (z.B. für sehr lange Textdokumente) benutzt.

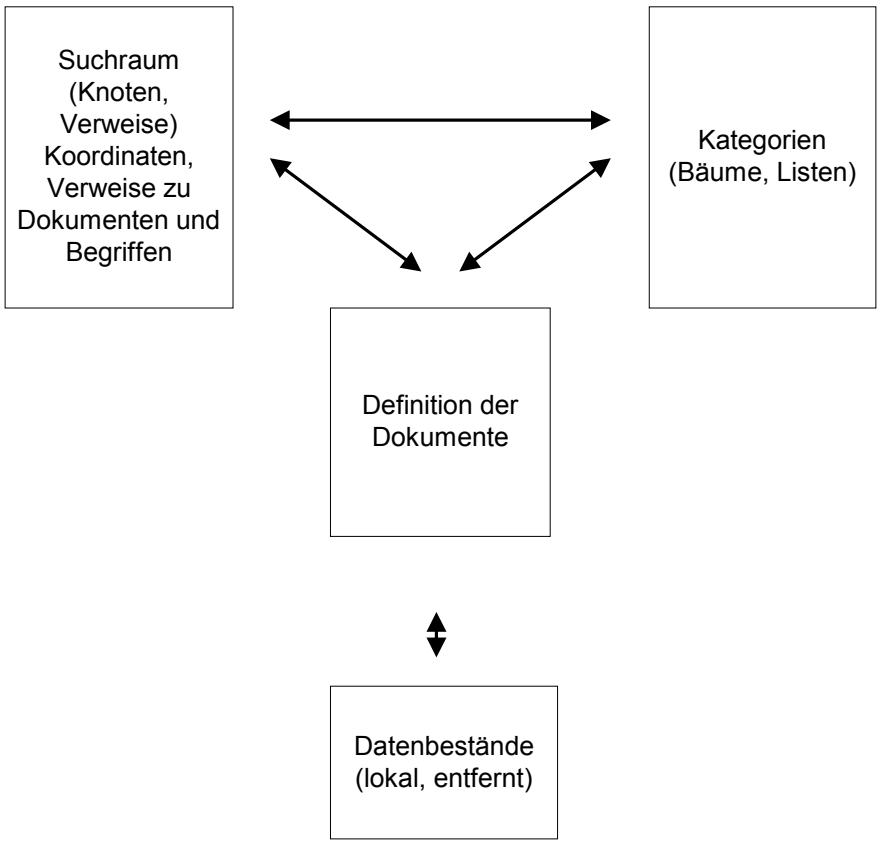



**Abbildung 3: Tabellenkonzept (Tabellengruppen) der Datenbank**

Die Datenbanktabellen lassen sich in vier verschiedene Gruppen unterteilen, wie auch in der obenstehenden Abbildung gezeigt. Interne Tabellen für die Administration (Autorisierung, Benutzerkennungen usw.) sind nicht abgebildet.

#### 2.2.2.1. Definition von Dokumenten

Diese wichtige Tabellengruppe enthält beschreibende Informationen über jedes einzelne Dokument des Systems. Unter anderem sind dies:

- Name des Dokuments
- eine Kurzbeschreibung
- eine detaillierte Beschreibung; sie wird häufig aus Informationen automatisch generiert, die in der Rohdatenquelle vorhanden sind oder aus intellektuell, von Hand erstellten zusätzlichen Rohdaten.

- zeitreihenspezifische Informationen wie Periodizität, Startdatum und Dimension der Masseinheit (TDM, Kubikmeter) usw.
- Verweise zu Autoren, Quellen etc.

Diese Tabellengruppe enthält also Metainformationen der einzelnen Dokumente innerhalb des gesamten Nutzdatenraums.
Ein einzelnes Dokument kann fast unbeschränkt vielen verschiedenen Kategorien zugeordnet sein. Das bedeutet, dass ein einzelnes Dokument innerhalb verschiedener Zusammenhänge verortet sein kann. Außerdem können Dokumente verschiedener Bestände unter einer Ergebnismenge angezeigt werden.

#### 2.2.2.2. Kategorien und Begriffe

In dieser Gruppe fnden sich die dem Benutzer auf der Clientseite zur Auswahl stehenden Kategorien und Begriffshierarchien wieder. Diese sind abhängig von ihrem Inhalt als Listen oder Bäume hinterlegt. Dazu kommen noch die nötigen Verweistabellen zu den Definitionen, um die benötigten m:n Relationen darzustellen.
Da die Struktur bei allen Kategorietabellen (Bäumen) gleich bleibt, lassen sich relativ leicht Abfragen und Operationen für diese Tabellen entwickeln. Das Hinzufügen neuer Begriffe ändert dabei nichts am generellen Vorgehen.

#### 2.2.2.3. Datenraum

Ein Datenraum wird durch eine Tabelle (Verweistabelle oder Linktabelle) mit Koordinaten gebildet, die für bestimmte Kombinationen von Kategorieelementen gilt. Dieser Datenraum nennt sich Bestand.

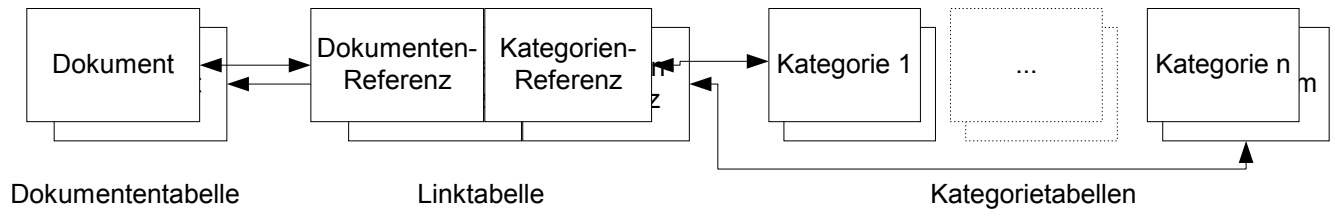



**Abbildung 4: Datenraum der Datenbank, Beschreibung via Kategorien, Linktabelle, Dokument**

Zusätzlich wird der Datenraum durch eine sogenannte Knotentabelle redundant definiert. Diese Knotentabelle beschleunigt allerdings den Zugriff auf den Datenraum. Nicht alle möglichen Permutationen werden in dieser Tabelle gespeichert, sondern nur die, die tatsächlich Dokumente enthalten. Sobald sich ein Datenbestand ändert, wird die Knotentabelle neu erzeugt.

**Abbildung 5: Datenraum der Datenbank, Beschreibung via Knotentabelle, Dokument**

Eine Datenbankanfrage (Query) sucht die Schlüssel (Satznummern) für die Kategorien in der Tabelle und liefert die Variablennamen mit Hilfe der Linktabelle (Verweistabelle).
Eine alternative Suchanfrage kann direkt mit Hilfe der Knotentabelle formuliert werden. Das setzt voraus Diese Art der Suchanfrage ist deutlich schneller als eine Anfrage über die Linktabelle, allerdings setzt dies die das Vorhandensein der Knotentabelle beim Client voraus sowie deren korrekte Benutzung. In einigen Fällen ist dies nicht praktikabel, weil die Knotentabellen außerordentlich groß sind und lokal vom Client gehalten werden können. Jede Änderung in der Knotentabelle würde außerdem einen langwierigen Aktualisierungsprozess beim Client erfordern.
Textdokumente werden auf dieselbe Art gespeichert. Hierbei ist allerdings die definierte Variable ein BLOb (Binary Large Object), für die im Projekt benutzte Ingres-Datenbank ein Long Var Char Datentyp.

#### 2.2.2.4. Datenbestände, Nutzdaten

Zeitreihen werden als Variable-Zeit-Wert-Tripel gespeichert. Die führt einerseits zu einem erwähnenswerten Overhead und Redundanz, andererseits erlaubt die Speicherung von Zeitreihen verschiedener Periodizität und Länge. Der Zugriff auf die so strukturierten Daten ist allerdings sehr schnell, einfach und variabel (verschiedene Zeitreihen in einer Anfrage). Die Darstellung der Zeitreihen(-werte) ist eine der Aufgaben des Clients.

### 2.2.3. Datenbanktabellen

Die Struktur der Datenbank ist hierarchisch organisiert.
Von einem Top-Down-Standpunkt können vier vierschiedene Tabellentypen definiert werden, die in der folgenden Tabelle dargestellt sind.

| | |
|---|---|
| **Domäne** | Dies ist der übergeordnete Name der gesamten Datenbank (z.B. Statlink)<br>Die Metatabellen einer Domäne gelten für die gesamten Daten. Einer Domäne sind ein oder mehrere ... |
| **Bestände** | zugeordnet. Sie sind in diesem Modell einzelne Gruppen von Daten, wie z.B. die Volkswirtschaftliche Gesamtrechnung und der ifo Konjunkturtest. |
| **Kategorien** | Eine oder mehrere Kategorien beschreiben den begrifflichen Zugang zu einem vorher definierten Bestand. Es ist möglich, mehrere Sätze von Kategorien innerhalb eine Bestandes zu definieren und so verschiedene Ansichten und auch Unteransichten auf Datenbestand zu ermöglichen. |
| **Tabellen** | Tabellen sind die kleinste Einheit dieses Modells. Jeder Bestand hat Kategorien, dazugehörige Verweise und Knotentabellen sowie die Zeitreihentabellen. |

**Tabelle 1: Hierarchische Tabellentypen der DB**

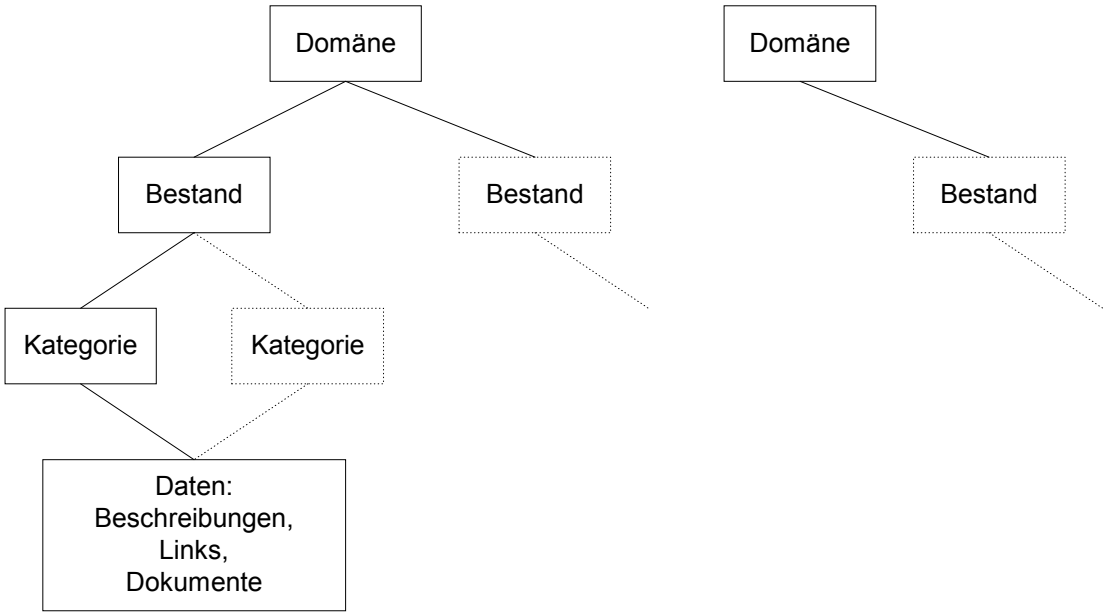



**Abbildung 6: Hierarchische Ansicht des Datenmodells**

Eine strikte inhaltliche Trennung zwischen den hier dargestellten Domänen ist nicht zwingend, sie dient hier nur zu besseren Anschauung. Es kann durchaus sinnvoll sein, einer Domäne Bestände aus einer anderen Domäne zuzuordnen. Beispielsweise wurde für das Projekt STATLINK die Domäne „Statlink“ erzeugt, die alle Daten enthält, die nach außen hin (aufgrund von Lizenzbestimmungen) zugänglich sein dürfen. Intern und zu Test- und Entwicklungszwecken wurde eine erweiterte Statlink-Domäne kreiert, die noch zusätzliche Bestände (Bundesbank) enthält. Auf diese Art ist es auch möglich, temporäre und auf einen bestimmten Zweck zugeschnittene Domänen zu definieren und anzubieten. Selbstverständlich sind die einzelnen Domänen mit einem Zugriffsschutz versehbar, so daß ein systematischer und kontrollierter Zugriff möglich ist.

### 2.2.3.1. Meta Daten

Allgemeine Metadaten-Tabellen speichern Informationen über alle Bestände einer Domäne, ihre Beziehung und detailliertere Beschreibungen einiger Datenbankattribute. Die Definition dieser Metadaten-Sets erweist sich insofern als ein schwieriges Unterfangen, als daß sie häufig Änderungen unterworfen sind. Neue Anforderungen erzwingen Redifinition beispielsweise der Benutzerinformationen oder der Definition von „fremden“ oder von völlig überarbeiteten Beständen, deren Ursprung u.U. nicht in unserem Hause liegt. So wurde auch die im DIW gepflegte Volkswirtschaftliche Gesamtrechnung (nach ESVG 1979) einer neuen Schematisierung unterworfen, so daß eine sinnvolle Fortführung als ein fortlaufender Bestand nicht mehr möglich war: die neue Volkswirtschaftliche Gesamtrechnung (nach ESVG 95) wurde als neuer Bestand implementiert.
Metatabellen werden zur Neuimplementierung eines Bestandes entworfen. Diese werden dann im Prozess der Erzeugung, des Ladens und des Updates des Bestandes gefüllt. Einige Attribute können mit Standardwerten gefüllt werden, so z.B. das Zeitreihenart-Attibut oder das Werteart-Attribut.
Wie in Abbildung 1 dargestellt, gibt es allgemeine Metatabellen sowie IR-bezogene Metatabellen. Diese Tabellen enthalten zum einen die bestandsdefinierenden Linkinformationen und zum anderen die IR-bezogenen Metadaten.

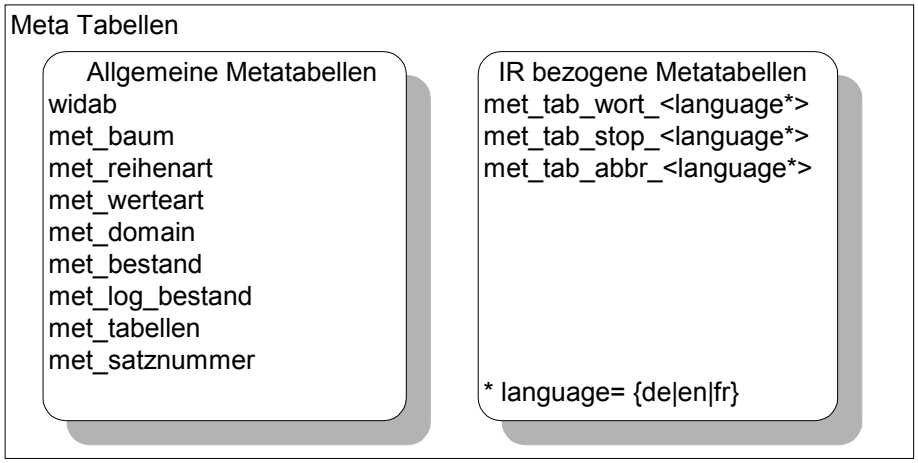



**Abbildung 7: Allgemeine und IR-bezogene Metatabellen**

Die IR-bezogenen Metatabellen erschliessen den gesamten Suchraum durch eine alternative Methode. Die Koordinaten des Suchraums werden für den wortinhaltlichen Zugriff durch IR-Methoden so definiert, daß eine neue, inhaltsbezogene Suche nach Stich- oder Schlagwörtern möglich wird. Es wird hierdurch möglich, freie, textuelle Anfragen zu formulieren, die dann durch Wortgrenzenerkennung, Wortstammbildung und Bewertungsverfahren vorhandenen Inhalten der DB zugeordnet werden können.

Dies wurde durch eine fast vollständige Neuindizierung der betroffenen Domäne erreicht..

Für weitere Details hierzu sehen Sie bitte in Kapitel 2.2.3.1.2. nach.

#### 2.2.3.1.1. Allgemeine Metadaten-Tabellen

Die folgende Abbildung 8 zeigt einen Überblick über die vorhandenen allgemeinen Metadaten-Tabellen. Sie werden im folgenden eingeführt.

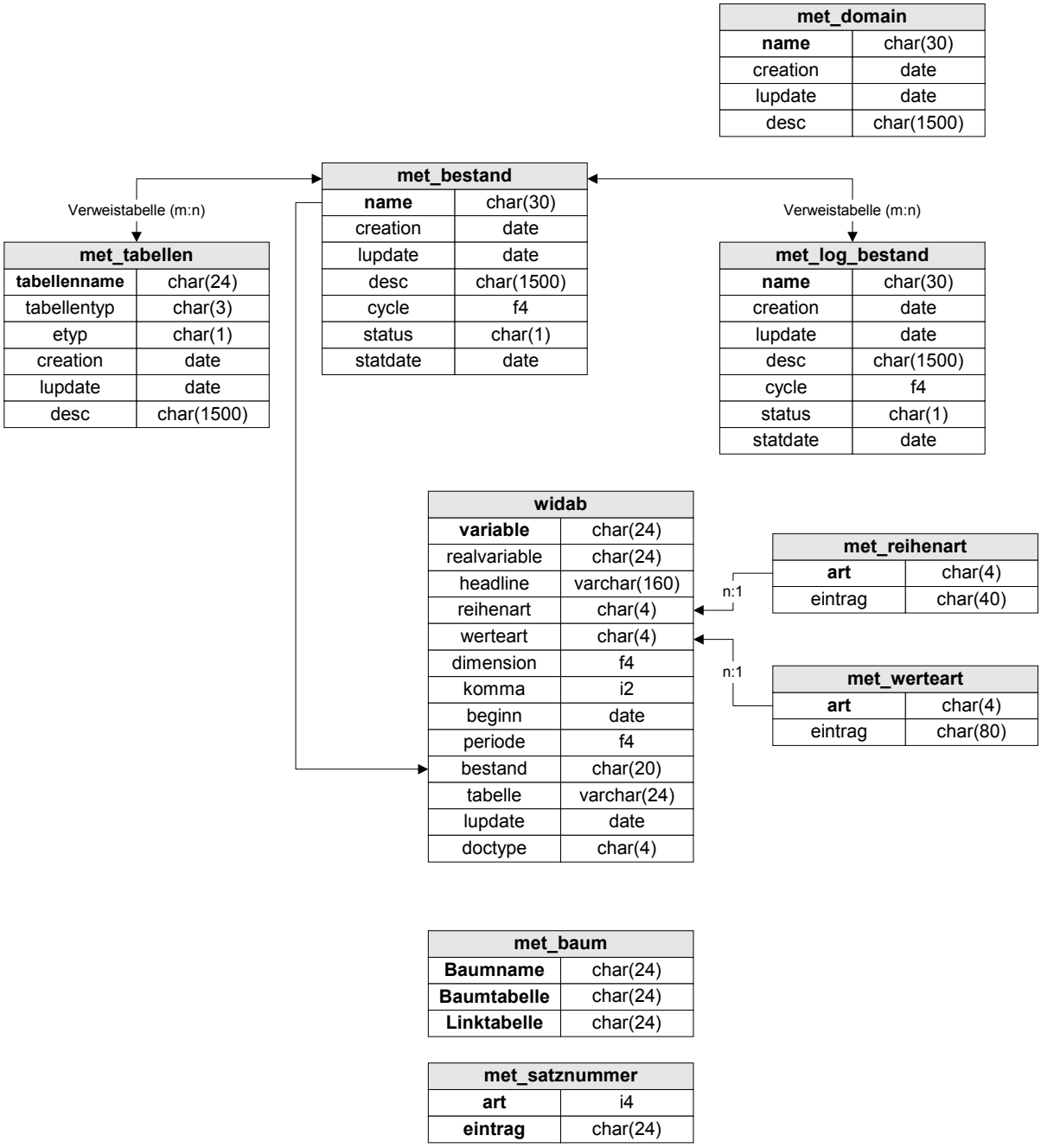



**Abbildung 8: Übersicht über die allgemeinen Metadaten-Tabellen**

Metatabellen sind bestandsübergreifend innerhalb der Domäne definiert.

*2.2.3.1.1.1. WIDAB*

Diese Tabelle ist die zentrale Tabelle der IR-Metadaten. Die enthält die Beschreibung jedes Dokuments der Domäne. Fast alle Anfragen als auch die Laderoutinen benutzen diese Tabelle.

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| variable | Char | 24 | no | no | 1 |
| realvariable | Char | 24 | yes | no | |
| headline | Varchar | 160 | no | no | |
| reihenart | Char | 4 | no | no | |
| werteart | Char | 4 | yes | no | |
| dimension | Integer4 | | yes | no | |
| bezeichnung | Char | 60 | yes | no | |
| komma | Integer2 | | yes | | |

| | | | | |
|---|---|---|---|---|
| Beginn | Date | | no | no |
| Periode | Float4 | | no | no |
| Bestand | Char | 20 | no | no |
| Tabelle | Varchar | 24 | no | no |
| Lupdate | Date | | no | no |
| Doctype | Char | 4 | no | no |

**Tabelle 2: Allgemeine Metatabelle WIDAB, Definition der Felder**

| **Attribute** | **Erläuterung** |
|---|---|
| variable | Ein optionaler Name für ein Dokument, z.B. für mehrfache Abbildung eines Dokument |
| realvariable | Kurzbeschreibung eines Dokuments |
| reihenart (Art der Zeitreihe) | Wird mit der Tabelle MET_REIHENART verknüpft; Reihenarten sind zur Zeit Bestandsgröße, Stomgröße, Quote und Index. Diese Angaben werden zur Weiterverarbeitung benötigt (.z.B. für die Durchschnittsbildung). |
| werteart (Art der Werte) | Wird mit der Tabelle MET_WERTEART verknüpft; die Werteart gibt z.B. an, ob die Werte Originalwerte oder saisonbereinigte Werte sind. Diese Angaben werden zur Weiterverarbeitung benötig (z.B. sollte man saisonbereinigte Werte nicht noch nochmals saisonbereinigen) |
| dimension | Größenangaben der Werte wie 1, 10, 100, 1000 etc.; die Angabe kann zur Weiterverarbeitung verwendet werden (z.B. für die graphische Darstellung) |
| bezeichnug | Arten von Einheiten wie Währung, Personenanzahl, tkm, Stunden etc. |
| komma (Dezimalstellen) | Anuahl der relevanten Kommastellen,. In einer Zeitreihentabelle kommen unterschiedlichste Reihen vor (z.B. Indices und Angaben in Mill. DM). Die Speicherung erfolgt immer als Gleitkommazahl. Die Ausgabe der Werte erfolgt aber formatiert nach der Anzahl der Kommastellen. |
| beginn (Datum) | Startdatum, im 'dd.mm.yyyy'-Format. Es obliegt dem Clientprogramm, vom Datum die nötigen Angaben zu extrahieren, um daraus Angaben wie 1. Quartal o.ä. zu erzeugen. Generell gilt aber für die Monats- und Quartalswerte, daß das Erhebungsdatum wichtig ist. |
| periode | Anzahl der Beobachtungen pro Jahr: 1, 2, 4, 12 entsprechen hier Jahres-, Hablsjahres-, Vierteljahres- und Monatswerten. 0,5 entspricht einem Zweijahreswert. |
| bestand | Name des Bestandes. Wir mit der Tabelle MET_BESTAND verknüpft. Dieses Attribut dient auch sehr oft für Wartungsarbeiten an einem Bestand. Sie ist unerläßlich beim Entfernen eines Bestandes. Typische Namen sind 'EuroStat-Linda', 'ifo-KT' etc. |
| tabelle | Name der Zeitreihentabelle; in dieser Tabelle stehen die zur Zeitreihe gehörenden Werte |
| lupdate | Datum der letzten Änderung dieser Variable |
| Doctype | Typ des Dokuments; in einer Erweiterung der Datenbank werden nicht nur lokal vorliegende Zeitreihen als Ergebnismenge gelifert, sondern auch Texte und andere Objekte/ Dokumente. Mögliche Werte sind 'LTS' (Local Time Series) und 'ASC' (ASCII-text). |

**Tabelle 3: Allgemeine Metatabelle WIDAB, Erläuterung der Felder**

#### *2.2.3.1.1.2. MET_BAUM*

Diese Tabelle speichert die Namen aller Kategorien (auch Bäume oder Listen) mit ihren zugehörigen Linktabellen. Sie wird von der baumbezogenen Anfrage und für die Erzeugung von Verweistabellen benutzt.

| **Name** | **Type** | **Length** | **Nulls** | **Defaults** | **Seq** |
|---|---|---|---|---|---|
| Baumname | char | 24 | no | no | 1 |
| Baumtabelle | char | 24 | no | no | 2 |

| | | | | | |
|---|---|---|---|---|---|
| Linktabelle | char | 24 | no | no | 3 |

**Tabelle 4: Allgemeine Metatabelle MET_BAUM, Definition der Felder**

| **Attribute** | **Erläuterung** |
|---|---|
| BAUMNAME | Der logische, vom Client sichtbare Name einer Kategorientabelle oder Liste |
| BAUMTABELLE | Der reale Name der Tabelle in der Datenbank. Die Idee hierbei ist, daß analog zu Views auf die gleiche Tabelle mit verschiedenen Namen zugegriffen werden kann. Es wären auch verschiedene Baumtabellen denkbar, die die gleiche Verweistabelle benutzten |
| LINKTABELLE | Die zugehörige Linktabelle, sie verweist auf das Variablenattribut in der zentralen WIDAB-Tabelle. |

**Tabelle 5: Allgemeine Metatabelle MET_BAUM, Erläuterung der Felder**

#### *2.2.3.1.1.3. MET_REIHENART*

Hier sind alle Attribute für Reihenart aufgeführt.

| **Name** | **Type** | **Length** | **Nulls** | **Defaults** | **Seq** |
|---|---|---|---|---|---|
| Art | char | 4 | no | no | 1 |
| Eintrag | char | 40 | no | no | |

**Tabelle 6: Allgemeine Metatabelle MET_REIHENART, Definition der Felder**

| **Attribute** | **Erläuterung** |
|---|---|
| art | Dieses Attribute korrespondiert mit dem Reihenattribut der WIDAB-Tabelle |
| eintrag | Beschreibung dieser Reihenart |

**Tabelle 7: Allgemeine Metatabelle MET_REIHENART, Erläuterung der Felder**

#### *2.2.3.1.1.4. MET_WERTEART*

Hier sind alle Attribute für Werteart aufgeführt.

| **Name** | **Type** | **Length** | **Nulls** | **Defaults** | **Seq** |
|---|---|---|---|---|---|
| art | Char | 4 | no | no | 1 |
| eintrag | Char | 80 | no | no | |

**Tabelle 8: Allgemeine Metatabelle MET_WERTEART, Definition der Felder**

| **Attribute** | **Erläuterung** |
|---|---|
| art | Dieses Attribut korrespondiert mit dem Werteartattribut der WIDAB-Tabelle |
| eintrag | Beschreibung dieser Werteart |

**Tabelle 9: Allgemeine Metatabelle MET_WERTEART, Erläuterung der Felder**

#### *2.2.3.1.1.5. MET_DOMAIN*

Diese Tabelle enthält detaillierte Angaben zur Domäne.

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| name | Char | 30 | no | no | 1 |
| creation | Date | | no | no | |
| lupdate | Date | | no | no | |
| desc | Char | 1500 | yes | no | |

**Tabelle 10: Allgemeine Metatabelle MET_DOMAIN, Definition der Felder**

| **Attribute** | **Erläuterung** |
|---|---|
| name | name of the domain |
| creation | creation date |
| lupdate | last updated; date when new data came in. |
| desc | detailed description of the domain. |

**Tabelle 11: Allgemeine Metatabelle MET_DOMAIN, Erläuterung der Felder**

### *2.2.3.1.1.6. MET_LOG_DOMAIN*

Diese Tabelle enthält einige Angaben zur logischen Domäne.

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| name | Char | 30 | no | No | 1 |
| creation | Date | | no | No | |
| lupdate | Date | | no | No | |
| desc | Char | 1500 | yes | No | |

**Tabelle 12: Allgemeine Metatabelle MET_LOG_DOMAIN, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| name | name of the logical domain or stock |
| creation | creation date; date when new data came in. |
| lupdate | last updated; |
| desc | detailed description of the domain. |

**Tabelle 13: Allgemeine Metatabelle MET_LOG_DOMAIN, Erläuterung der Felder**

### *2.2.3.1.1.7. MET_BESTAND*

Diese Tabelle enthält detailierte Angaben zum Bestand.

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| name | Char | 30 | no | no | 1 |
| creation | date | | no | no | |
| lupdate | date | | no | no | |
| desc | char | 1500 | yes | no | |
| cycle | float4 | | yes | no | |
| status | char | 1 | yes | no | |
| statdate | date | | yes | no | |

**Tabelle 14: Allgemeine Metatabelle MET_BESTAND, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| NAME | Name des Bestandes |
| CREATION | Erstellungsdatum |
| LUPDATE | Letzte Änderung |
| DESC | Detaillierte Beschreibung des Bestandes |
| CYCLE | Dieses Feld gibt das Intervall an, in dem der Bestand aktualisiert wird. |
| STATUS | Zustand des Bestandes. Dieses Attribut ist noch experimentell und gibt darüber Auskunft, ob ein Bestand eingefroren (unveränderlich) oder beschädigt ist. |
| STATDATE | Zeitpunkt der letzten Statusveränderung |

**Tabelle 15: Allgemeine Metatabelle MET_BESTAND, Erläuterung der Felder**

### *2.2.3.1.1.8. MET_LOG_BESTAND*

Diese Tabelle enthält detailierte Angaben zum logischen Bestand.

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| name | char | 30 | no | no | 1 |
| creation | date | | no | no | |
| lupdate | date | | no | no | |
| desc | char | 1500 | yes | no | |
| cycle | float4 | | yes | no | |
| status | char | 1 | yes | no | |
| statdate | date | | yes | no | |

**Tabelle 16: Allgemeine Metatabelle MET_LOG_BESTAND, Definition der Felder**

| Attribute | Erläuterung |
|---|---|

| | |
|---|---|
| NAME | Name des logischen Bestandes |
| CREATION | Erstellungsdatum |
| LUPDATE | Letzte Änderung |
| DESC | Detaillierte Beschreibung des logischen Bestandes |
| CYCLE | Dieses Feld gibt das Intervall an, in dem der logische Bestand aktualisiert wird. |
| STATUS | Zustand des logischen Bestandes. Dieses Attribut ist noch experimentell und gibt darüber Auskunft, ob ein logischer Bestand eingefroren (unveränderlich) oder beschädigt ist. |
| STATDATE | Zeitpunkt der letzten Statusveränderung |

**Tabelle 17: Allgemeine Metatabelle MET_LOG_BESTAND, Erläuterung der Felder**

#### *2.2.3.1.1.9. MET_TABELLEN*

Hier wird die kleinste Einheit der Datenbank, das Dokument beschrieben..

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| Tabellenname | char | 24 | no | no | 1 |
| Tabellentyp | char | 3 | no | no | |
| Etyp | char | 1 | no | no | |
| Creation | date | | no | no | |
| Lupdate | date | | no | no | |
| Desc | char | 1500 | yes | no | |

**Tabelle 18: Allgemeine Metatabelle MET_TABELLEN, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| Tabellenname | Der reale Name der Tabelle |
| Tabellentyp | Entsprechen zur Namenskonvention der Tabellen werden hier die Tabellentypen eingetragen |
| Etyp | Typ der Kategorie. Hier werden clientspezifische Angaben für einzelne Tabellen gemacht |
| Creation | Erzeugungsdatum |
| Lupdate | Letzte Änderung |
| Desc | Ausführliche Beschreibung der Tabelle (wenn nötig) |

**Tabelle 19: Allgemeine Metatabelle MET_TABELLEN, Erläuterung der Felder**

#### *2.2.3.1.1.10. MET_SATZNUMMER*

Diese Tabelle funktioniert als Nummernspeicher für die gesamte Domain. Diese Tabelle wird nur bei Datenbankaktualisierungen (Änderungen und Löschungen) gebraucht.

| Name | Type | Length | Nulls | Defaults | Seq |
|---|---|---|---|---|---|
| satznummer | integer4 | | no | no | 2 |
| tabellenname | char | 24 | no | no | 1 |

**Tabelle 20: Allgemeine Metatabelle MET_SATZNUMMER, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| Tabellenname | Ein Schlüssel |
| Tabellentyp | Die Tabelle, der der angegebene Schlüssel zugeordnet ist. |

**Tabelle 21: Allgemeine Metatabelle MET_SATZNUMMER, Erläuterung der Felder**

Während des Löschvorgang können Satznummern hier zwischengespeichert werden. Andere Datenbankfunktionen nutzen diese Tabelle, um die erste freie Satznummer zu finden.

### 2.2.3.1.2. IR-bezogene Metadaten

IR-bezogene Metadaten (IR= Information Retrieval) erschließen den Suchraum auf eine neue Art. Anders als beim systematischen, bestandsorientierten und kategorisierten Zugang erlauben IR-Metadaten den direkten wortinhaltlichen Zugang zu Dokumenten.
Verschiedene Bestände befinden sich anfangs inhaltlich getrennt und unabhängig im Suchraum. Dies veranschaulicht untenstehende Abbildung 9.

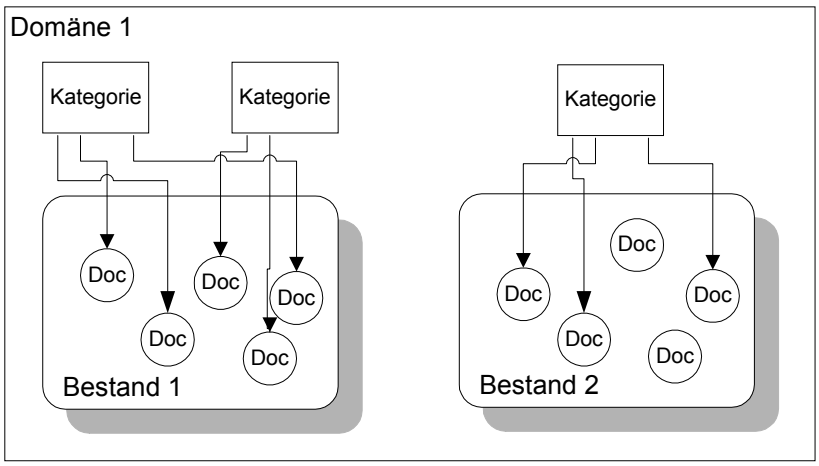



**Abbildung 9: Bestandsbezogener, herkömmlicher Dokumentenzugriff**

Deutlich ist hier erkennbar, daß die Dokumente der einzelnen Bestände jeweils nur durch die gültigen Kategorien eines Bestandes erschlossen werden können und somit auch der inhaltlich-strukturellen Vorgabe der Bestandsdefinition folgen müssen. Deutlicher Nachteil dieser Dokumentenreferenzierung ist, daß der Suchende sich mit der Systematisierung und der Definition des Bestandes auskennen bzw. einarbeiten muß. Dies kann durch spezielle Begriffsverwendung innerhalb von (logischen) Beständen erschwert werden. Bespielsweise werden die Begriffe Industriezweig und Branche synonym verwendet, was die inhaltliche Orientierung eines Nutzers erschweren kann.
In unserem neuen Ansatz werden die Dokumente dieser Bestände nun zusätzlich mit Hilfe von bestandsübergreifenden Schlüssel- oder Schlagwörter in neue Beziehungen gesetzt, wie in der folgenden Abbildung 10 gezeigt.

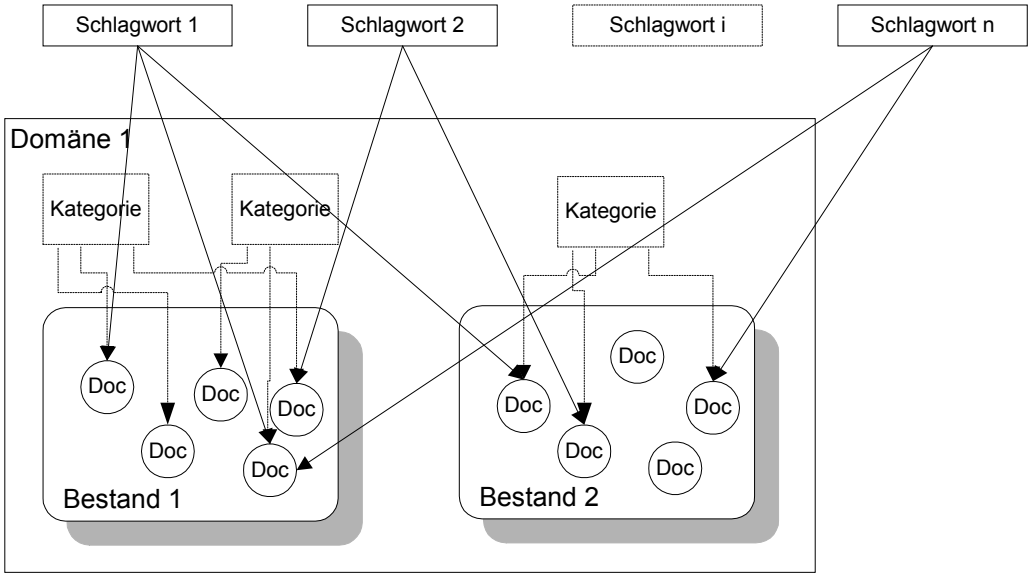



**Abbildung 10: Bestandsloser, inhaltlicher Zugriffs auf Dokumente über Schlagwörter**

Diese inhaltliche Neuerschließung der Bestände ermöglich eine Sicht auf den Suchraum, der durch die Addition der einzelnen Bestände entsteht. Eine formulierte Suchanfrage über die IR-bezogenen Metadaten liefert also eine Ergebnismenge aus der Gesamtheit der Domäne. Aufgrund der inhaltlichen Verschlagwortung vermindert sich für den Benutzer die Einarbeitungszeit in die Bestandsdefinition, da er inhaltliche Zusammenhänge bestandsübergreifend erkennen kann. Durch die im Suchergebnis nun parallel erscheinenden Bestände erschließen sich u.U. neue Einsichten für den Nutzer.
Es wurden gleichbedeutende Wörter, sogenannte Synonyme in die IR-Funktionalität mit aufgenommen. Synonyme werden wie normale Schlagwörter behandelt. Nähere Erläuterungen finden Sie unter 2.2.3.1.2.1 „MET_TAB_WORT_“.
Die folgenden Tabellen sind für die IR-bezogenen Daten definiert worden.

### *2.2.3.1.2.1. MET_TAB_WORT_*

Diese Tabelle ist die zentrale IR-Metadatentabelle. Sie speichert den Thesaurus aller generierten Schlagwörter.
Die Tabelle wurde in den Sprachen deutsch, englisch und französisch angelegt, um den multilingualen Ansatz zu verwirklichen.

| Name | Type | Length | Nulls | Defaults |
|---|---|---|---|---|
| wkey | integer4 | | No | no |
| eintrag | varchar | 100 | No | no |
| stemeintrag | varchar | 100 | Yes | no |

**Tabelle 22: IR- Metatabelle MET_TAB_WORT_, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| wkey | |
| eintrag | |
| stemeintrag | |

**Tabelle 23: IR- Metatabelle MET_TAB_WORT_, Erläuterung der Felder**

Die Tabelle enthält nicht nur die durch intellektuelle Indexierung gewonnenen Schlagwörter, sondern auch die Synonyme, die im Laufe der Indexierung in den Beständen identifiziert wurden. Diese liegen in einem Rohdatenformat und werden beim Laden der MET_TAB_WORT_ mit denselben Referenzen in MET_LNK_WORT

versehen. Synonyme erzeugen einfach wie die originalen Schlagwörter dieselben Einträge in der Verweistabelle.

#### *2.2.3.1.2.2. MET_TAB_STOPP_*

Diese Tabelle enthält alle Stoppwörter.

Die Tabelle wurde in den Sprachen deutsch, englisch und französisch angelegt, um den multilingualen Ansatz zu verwirklichen.

| Name | Type | Length | Nulls | Defaults |
|---|---|---|---|---|
| eintrag | varchar | 100 | no | no |

**Tabelle 24: IR- Metatabelle MET_TAB_STOPP_, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| eintrag | Dieses Feld enthält das Stoppwort. |

**Tabelle 25: IR- Metatabelle MET_TAB_STOPP_, Erläuterung der Felder**

Die Stoppwortliste ist so konzipiert, daß in einem Text möglichst direkt die irrelevanten Wörter erkannt und eleminiert werden können. Das bedeutet für die Stoppwortliste eine hohe Redundanz in Bezug auf die grammatisch/ lexikalischen Ausprägungen bestimmter Wörter. Die Stoppwortliste enthält nämlich alle Deklinationen bzw. Konjugationen eines Wortes. Diese Redundanz soll verhindern, daß ein größerer algorithmischer Aufwand und entsprechende Rechenleistung entsteht, wenn alle lexikalisch-grammatischen Änderungen dynamisch ausgeführt werden müssen.
Der Vorteil unserer Datenbankkonzeption, der schnelle Zugriff und entsprechend flinke Vergleichsoperationen, können durch diese Methode genutzt werden.

#### *2.2.3.1.2.3. MET_LNK_WORT_*

Diese Tabelle ist die zentrale IR- Verweistabelle für die IR-Metadaten. Sie referenziert die Schlagwörter mit den den Kategorien der einzelnen logischen Bestände.
Die Tabelle wurde in den Sprachen deutsch, englisch und französisch angelegt, um den multilingualen Ansatz zu verwirklichen.

| Name | Type | Length | Nulls | Defaults |
|---|---|---|---|---|
| wkey | integer4 | | no | no |
| kategorie | char | 40 | no | no |
| satznummer | integer4 | | no | no |

**Tabelle 26: IR- Metatabelle MET_ LNK_WORT_, Definition der Felder**

| Attribute | Erläuterung |
|---|---|
| wkey | Schlagwortschlüssel |
| kategorie | Name der Kategorie-Tabelle (bam_*) |
| satznummer | Satznummer in der bam_*-Tabelle |

**Tabelle 27: IR- Metatabelle MET_ LNK_WORT_, Erläuterung der Felder**

Der wkey ist der frei vergebene Schlüssel, um das Schlagwort aus met_tab_wort zu referenzieren.
Wie oben bereits erwähnt, werden die Referenzen synonymer Wörter in dieser Verweistabelle identisch mit denen der Ausgangs- oder Originalwörter eingetragen.

### 2.2.3.1.3. Gebrauch der IR-Metatabellen

In der folgenden Abbildung 11 sind die IR-bezogenen Metadatentabellen zu sehen, die für das Beispiel der VGR-Konten und des IFO-Konjunkturtests benutzt wurden.

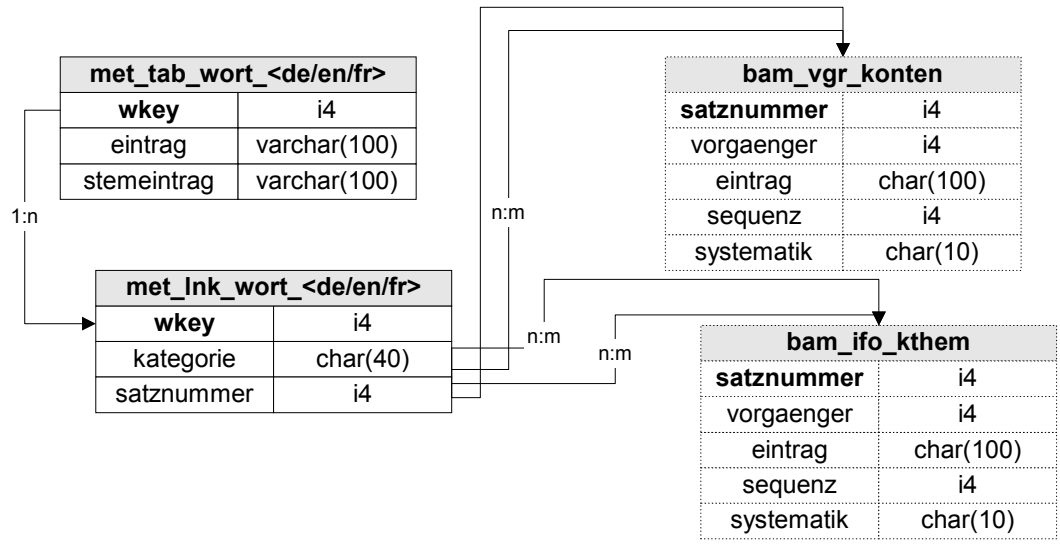



**Abbildung 11: IR-bezogene Metadaten-Tabellen am Beispiel VGR/ KT**

Durch die Einführung eines Schlüssel- oder Schlagwortes in die Tabelle met_tab_wort lassen sich nun frei Beziehungen zu BEIDEN hier beispielhaft genannten Beständen knüpfen.
In der schematischen Darstellung der Abbildung 10 wurde aus Gründen der Anschaulichkeit darauf verzichtet, die Beziehung von Schlagwort und „Dokument" genauer aufzuzeigen; in Abbildung 11 allerdings ist die Art der Referenz deutlich sichtbar: sie ist so definiert, daß ein Schlagwort sowohl die relevanten Einträge der Kategorien als auch, über die existierenden Verweistabellen, die zugeordneten Dokumente referenziert.
Der Thesaurus der Schlagwörter wurde von einem Experten der Wirtschaftswissenschaften durch manuelle, intellektuelle Indexierung angelegt. Er wurde so konzipiert, daß ein verlässliches und kategorisiertes Verzeichnis als Grundlage der IR-Methoden dienen kann. Mehrsprachigkeit wurde beim Entwurf der IR-bezogenen Methoden berücksichtigt. Zu diesem Zweck sind, wie in Abbildung 11 zu sehen, sprachbezogene Metatabellen durch ein internationales Länderkürzel versehen.
In dem Kapitel 3.4 „Terming, Parsing und Filtern" auf Seite 69 sind die übrigen Metatabellen (Stoppworte, Synonyme) aufgeführt und erklärt.

### 2.2.3.2. Dokumente

Grundsätzlich können die Dokumentendatentabellen alle Arten von Textdatenformaten aufnehmen, z.B. sehr große Textdateien (Text-Daten), aber keine binären Daten. Intern werden sie allerdings als sog. BLOBs (Binary Large Object) gespeichert. Letztere können sehr große Texte enthalten (bis 2GB). Die direkte Einbringung von pdf-, Word- oder Postscript-Dokumenten in die Datenbank ist bislang nicht vorgesehen; diese Dokumententypen müssen vorher in ein Textformat konvertiert werden.
Bisher haben können wir in unserer Datenbank folgende Datentypen speichern. Die Folgende Tabelle gibt einen Überblick:

| **Dokumententyp** | **Zweck** |
|---|---|
| Zeitreihe | speichert statische Zeitreihen |
| Textdokument | speichert ASCII-Texte |
| URL | speichert WWW-Referenzen auf Dokumente |

**Tabelle 28: Dokumententypen**

Als ein Beispiel für Dokumenten-Daten soll hier die Volkswirtschaftliche Gesamtrechnung (ESVG87) erklärt werden. Dieser Bestand wird vom DIW entwickelt, berechnet und gepflegt.
Der Bestand besteht aus zwei logischen Beständen,

- Der VGR nach Konten und
- Der VGR nach Sektoren

Jeder der logischen Bestände besteht wiederum aus zwei Kategorien, jeweils einer Regionen-Kategorie und einer Themen- bzw. Konten-Kategorie.
Die Rohdaten werden aus hausinternen Beständen (Excel-Tabellen, Berechnungsprogrammen) herausgeschrieben, konvertiert und dann in die Datenbank geladen.
Für diesen Bestand sind verschiedene Tabellentypen relevant. Als erstes ist die WIDAB-Tabelle zu nennen, die die Kategorien darstellenden „bam_“ Tabellen, die verknüpfenden Verweistabellen „lnk_“, die Nummernspeichertabellen „num_“ und die Dokumententabelle „tab_“. Dieses hier für die VGR dargestellte Schema gilt für alle Bestände aller Domänen.
Gemeinsam sind ebenfalls die Operationen, die auf ihnen ausgeführt werden können. Tabellen werden erzeugt, geladen, aktualisiert oder anderweitig von speziellen Programmen manipuliert.
Wie in Abbildung 12 und Abbildung 13 zu erkennen, können Dokumente innerhalb der allgemeinen Metatabellenstruktur bereits auf zwei unterschiedliche Arten referenziert werden.
Die folgende Abbildung Abbildung 12 zeigt das Referenzieren der Dokumenten-Tabelle mit Hilfe sogenannter Verweistabellen (lnk_vgr_konten, lnk_vgr_region).

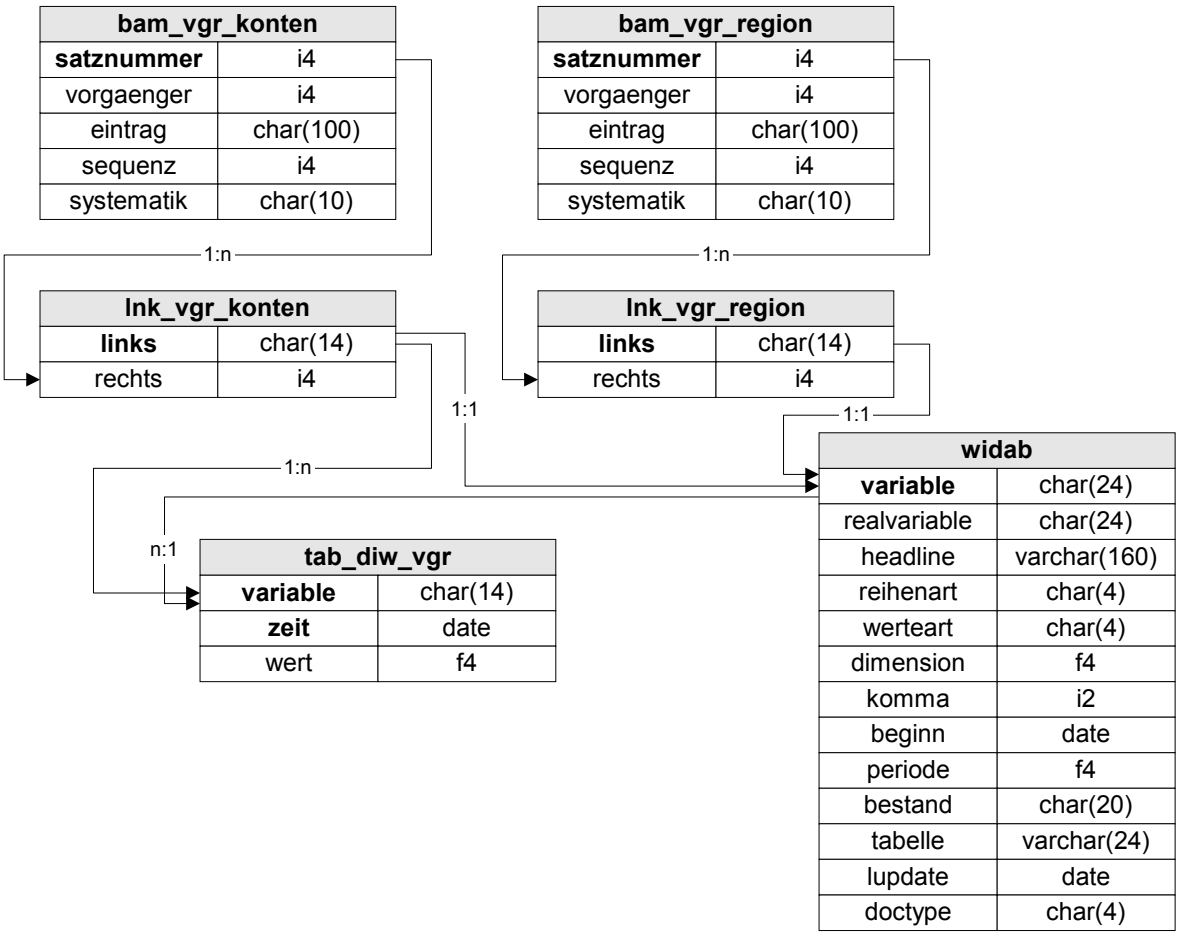



**Abbildung 12: Dokumenten-Tabellen, direkte Referenzierung über Verweistabellen**

In der Abbildung werden die Tabellen für den logischen Bestand „VGR nach Konten“ aufgezeigt. Der logische Bestand „VGR nach Sektoren“ benutzt einfach die Tabellen bam_vgr_*sectors* und lnk_vgr_*sectors* anstelle von *_konten* um die betreffenden Dokumente innerhalb dieses Bestandes zu referenzieren.

Innerhalb des Bestandes „VGR“ können Dokumente alternativ mit Hilfe von sogenannten Knotentabellen referenziert werden, wie die folgende Abbildung 13 zeigt.

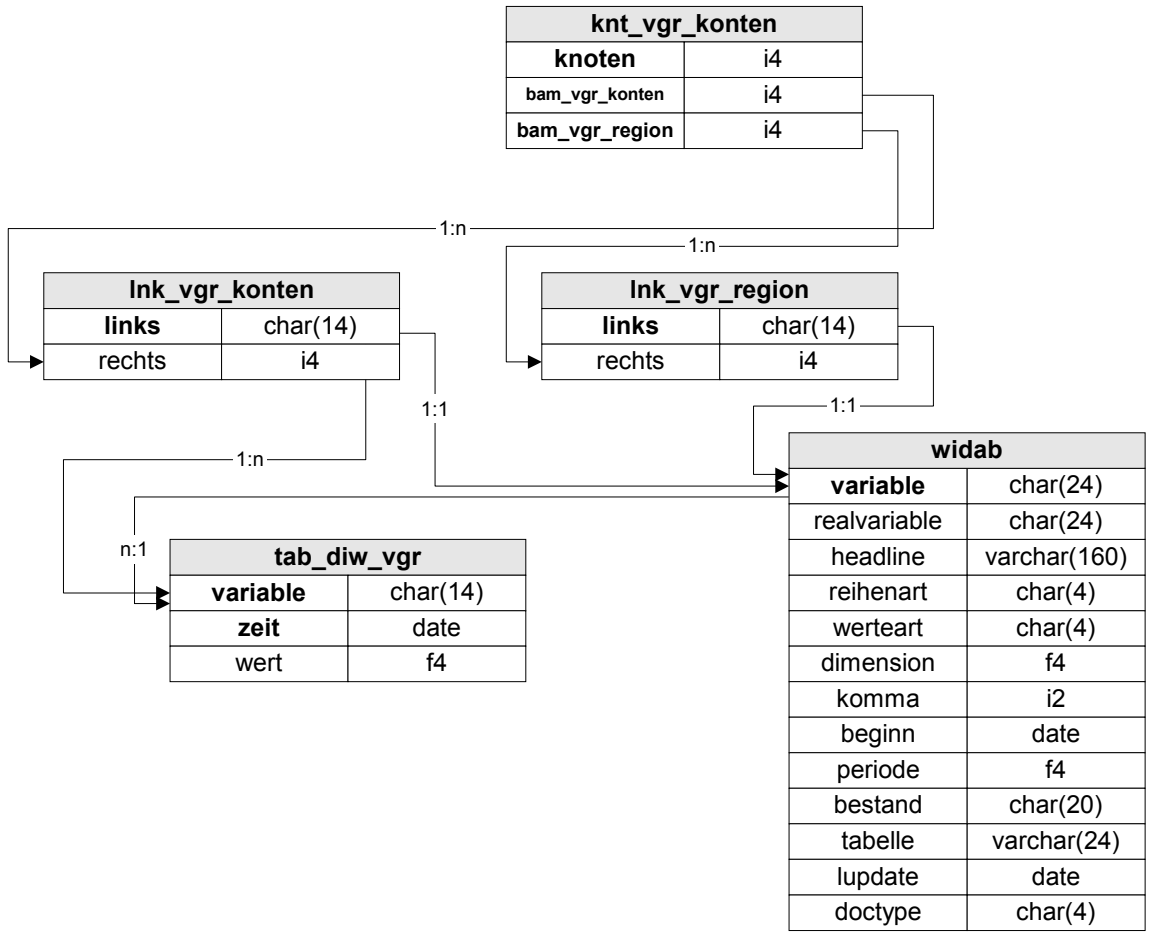



**Abbildung 13: Dokumenten-Tabellen, Referenzierung über Knotentabellen**

Die häufig redundant ausgeführte Referenzierung via Knotentabellen hat den Vorteil einer deutlich schnelleren Antwortzeit des eingesetzten DBMS.
Knotentabellen sind so definiert, daß sie die Gesamtergebnismenge aller über die Kategorien ausführbaren Anfragen bereits enthalten. Die Knotentabellen werden genau dann neu erzeugt, sobald sich die Kategorien oder die Dokumente ändern;, das heißt, wenn neue Referenzen hinzukommen oder alte gelöscht werden müssen. Knotentabellen enthalten also genau ALLE Kategoriekombinationen, für die es mindestens einen Treffer in der Dokumentenmenge dieses Bestandes gibt.

## *2.3. Alphabetische Übersicht über die verwendeten Datenbanktabellen*

bam_bau_bereiche
bam_bau_laender
bam_bau_produzenten
bam_bau_produzenten60
bam_bau_themen
bam_brand_arbeitsmarkt
bam_brand_aregion
bam_brand_bbranchen
bam_brand_bip
bam_brand_bregion
bam_brand_bthemen
bam_brand_eregion
bam_brand_erwt
bam_brand_iregion

bam_brand_nace
bam_brand_nregion
bam_brand_region
bam_brand_themen
bam_buba_tabelle
bam_buba_themen
bam_chile_animal
bam_chile_theme
bam_eurostat_branch
bam_eurostat_branch_de
bam_eurostat_branch_en
bam_eurostat_branch_fr
bam_eurostat_country
bam_eurostat_country_de
bam_eurostat_country_en
bam_eurostat_country_fr
bam_eurostat_themes
bam_eurostat_themes_de
bam_eurostat_themes_en
bam_eurostat_themes_fr
bam_ifo_ibran
bam_ifo_ithem
bam_ifo_kbran
bam_ifo_kthem
bam_ifo_kthem_fapsy
bam_oecd_idis_pland
bam_oecd_idis_rland
bam_oecd_idis_sector
bam_oecd_idis_themes
bam_oecd_land
bam_oecd_periode
bam_oecd_themes
bam_stabu_branche
bam_stabu_crippled_segs
bam_stabu_region
bam_stabu_segmente
bam_stabu_thema
bam_vgr_konten
bam_vgr_kthem
bam_vgr_region
bam_vgr_sektor
bam_vgr_sthem
bam_widab_quellen
bam_zvei_abran
bam_zvei_abran98
bam_zvei_athem
bam_zvei_kbran
bam_zvei_kland
bam_zvei_kthem
bam_zvei_kthem95
bam_zvei_land
bam_zvei_land_ber
bam_zvei_land_par
bam_zvei_pbran98
bam_zvei_pthem
ind_bau_bereiche
ind_bau_bereiche60
ind_bau_laender
ind_bau_produzenten
ind_bau_produzenten60
ind_bau_themen

ind_brand_arbeitsmarkt
ind_brand_bbranchen
ind_brand_bip
ind_brand_bthemen
ind_brand_erwt
ind_buba_tabelle
ind_eurostat_branch
ind_eurostat_country
ind_eurostat_themes
ind_knt_ah98_abran
ind_knt_ah98_athem
ind_knt_ah98_land_ber
ind_knt_ah98_land_par
ind_knt_aussenh_abran
ind_knt_aussenh_athem
ind_knt_aussenh_land_ber
ind_knt_aussenh_land_par
ind_lnk_ifo_ibran
ind_lnk_ifo_ithem
ind_lnk_ifo_kbran
ind_lnk_ifo_kthem
ind_lnk_ifo_kthem_fapsy
ind_lnk_stabu_branche
ind_lnk_stabu_region
ind_lnk_stabu_segmente
ind_lnk_stabu_thema
ind_lnk_vgr_konten
ind_lnk_vgr_region
ind_lnk_zvei_abran
ind_lnk_zvei_abran98
ind_lnk_zvei_athem_re
ind_lnk_zvei_land_ber_re
ind_lnk_zvei_land_par_re
ind_lnk_zvei_land_prod
ind_lnk_zvei_pbran98
ind_lnk_zvei_pthem_re
ind_oecd_idis_pland
ind_oecd_idis_rland
ind_oecd_idis_sector
ind_oecd_idis_themes
ind_oecd_land
ind_oecd_periode
ind_oecd_themes
ind_widab_bestand
ind_widab_quellen
ind_widab_tabellen
ind_zvei_kbran
ind_zvei_kland
ind_zvei_kthem
ind_zvei_kthem95
knt_brand_arbeitsmarkt
knt_brand_bau
knt_brand_bip
knt_brand_erwt
knt_brand_nace
knt_chile_rinder
knt_diw_bauvolumen
knt_diw_bauvolumen60
knt_eurostat_linda
knt_ifo_investitionstest
knt_ifo_konjunkturtest

knt_ifo_ktfapsy
knt_oecd_idis_com
knt_oecd_idis_pla
knt_oecd_idis_sec
knt_oecd_mei
knt_vgr_konten
knt_vgr_sektor
knt_zvei_aussenhandel
knt_zvei_aussenhandel98
knt_zvei_konjunktur
knt_zvei_konjunktur95
knt_zvei_produktion98
knv_brand_arbeitsmarkt
knv_brand_bau
knv_brand_bip
knv_brand_erwt
knv_brand_nace
knv_chile_rinder
knv_diw_bauvolumen
knv_diw_bauvolumen60
knv_eurostat_linda
knv_ifo_investitionstest
knv_ifo_konjunkturtest
knv_ifo_ktfapsy
knv_oecd_idis_com
knv_oecd_idis_pla
knv_oecd_idis_sec
knv_oecd_mei
knv_vgr_konten
knv_vgr_sektor
knv_zvei_aussenhandel
knv_zvei_aussenhandel98
knv_zvei_konjunktur
knv_zvei_konjunktur95
knv_zvei_produktion98
lnk_bau_bereiche
lnk_bau_bereiche60
lnk_bau_laender
lnk_bau_produzenten
lnk_bau_produzenten60
lnk_bau_themen
lnk_brand_arbeitsmarkt
lnk_brand_aregion
lnk_brand_bbranchen
lnk_brand_bip
lnk_brand_bregion
lnk_brand_bthemen
lnk_brand_eregion
lnk_brand_erwt
lnk_brand_iregion
lnk_brand_nace
lnk_brand_nregion
lnk_brand_region
lnk_brand_themen
lnk_buba_tabelle
lnk_chile_animal
lnk_chile_theme
lnk_eurostat_branch
lnk_eurostat_country
lnk_eurostat_themes
lnk_ifo_ibran

lnk_ifo_ithem
lnk_ifo_kbran
lnk_ifo_kthem
lnk_ifo_kthem_fapsy
lnk_met_bestand
lnk_met_log_bestand
lnk_oecd_idis_pland
lnk_oecd_idis_rland
lnk_oecd_idis_sector
lnk_oecd_idis_themes
lnk_oecd_land
lnk_oecd_periode
lnk_oecd_themes
lnk_stabu_branche
lnk_stabu_region
lnk_stabu_segmente
lnk_stabu_thema
lnk_vgr_konten
lnk_vgr_kthem
lnk_vgr_region
lnk_vgr_sektor
lnk_vgr_sthem
lnk_widab_quellen
lnk_zvei_abran
lnk_zvei_abran98
lnk_zvei_athem
lnk_zvei_kbran
lnk_zvei_kland
lnk_zvei_kthem
lnk_zvei_kthem95
lnk_zvei_land_ber
lnk_zvei_land_par
lnk_zvei_land_prod
lnk_zvei_pbran98
lnk_zvei_pthem
met_baum
met_bestand
met_domain
met_lnk_wort_de
met_lnk_wort_en
met_lnk_wort_fr
met_log_bestand
met_reihenart
met_satznummer
met_tab_stopp_de
met_tab_stopp_en
met_tab_stopp_fr
met_tab_wort_de
met_tab_wort_en
met_tab_wort_fr
met_tabellen
met_werteart
num_bau_bereiche
num_bau_laender
num_bau_produzenten
num_bau_produzenten60
num_bau_themen
num_brand_arbeitsmarkt
num_brand_bbranchen
num_brand_bip
num_brand_bthemen

num_brand_erwt
num_brand_nace
num_brand_region
num_brand_themen
num_buba_gesamt
num_buba_tabelle
num_eurostat_branch
num_eurostat_country
num_eurostat_themes
num_ifo_ibran
num_ifo_ithem
num_ifo_kbran
num_ifo_knt_investition
num_ifo_knt_konjunktur
num_ifo_knt_ktfapsy
num_ifo_kthem
num_ifo_kthem_fapsy
num_knt_euro_linda
num_knt_oecd_mei
num_oecd_idis_pland
num_oecd_idis_rland
num_oecd_idis_sector
num_oecd_idis_themes
num_oecd_land
num_oecd_periode
num_oecd_themes
num_stabu_branche
num_stabu_segmente
num_stabu_thema
num_vgr_knt_konten
num_vgr_knt_sektor
num_vgr_konten
num_vgr_region
num_vgr_sektor
num_zvei_abran
num_zvei_abran98
num_zvei_athem
num_zvei_kbran
num_zvei_kland
num_zvei_knt_ahandel
num_zvei_knt_ahandel98
num_zvei_knt_konj95
num_zvei_knt_konjunktur
num_zvei_knt_prod98
num_zvei_kthem
num_zvei_kthem95
num_zvei_land
num_zvei_pbran98
num_zvei_pthem
tab_brand_gesamt
tab_buba_gesamt
tab_chile_rinder
tab_diw_bauvolumen
tab_diw_bauvolumen60
tab_diw_vgr
tab_eurostat_di
tab_eurostat_linda
tab_ifo_investitionstest
tab_ifo_konjunkturtest
tab_oecd_gesamt
tab_oecd_idis

tab_stabu_gesamt
tab_test_text
tab_zvei_knr_dat
tab_zvei_knr_dat98
tab_zvei_konjunktur
tab_zvei_konjunktur95
tab_zvei_prodbuaw
tab_zvei_prodbuaw95
tab_zvei_prodbuaw98
tab_zvei_wnr_dat
tab_zvei_zv_basis
tab_zvei_zv_basis98
widab

## *2.4. Dämon*

In diesem Kapital werden alle Funktionen rund um die Middleware-Komponente, den sogenannten Dämon, beschrieben.

### 2.4.1. Systemarchitektur

Das STATLINK-Projekt wurde als sogenanntes dreistufiges Client/Server System konzipiert.
Die Middleware-Komponente fungiert als Vermittler zwischen dem Benutzeragenten (Client) auf der einen und dem Datenbankmanagementsystem (DBMS) auf andere Seite.

**Abbildung 14: Dreistufiges Client/ Server Modell**

Das Herauslösen einer Vermittlungskomponente aus der Datenbank und aus den verschiedenen Endbenutzerprogrammen (Clients) hat mehrere Vorteile:

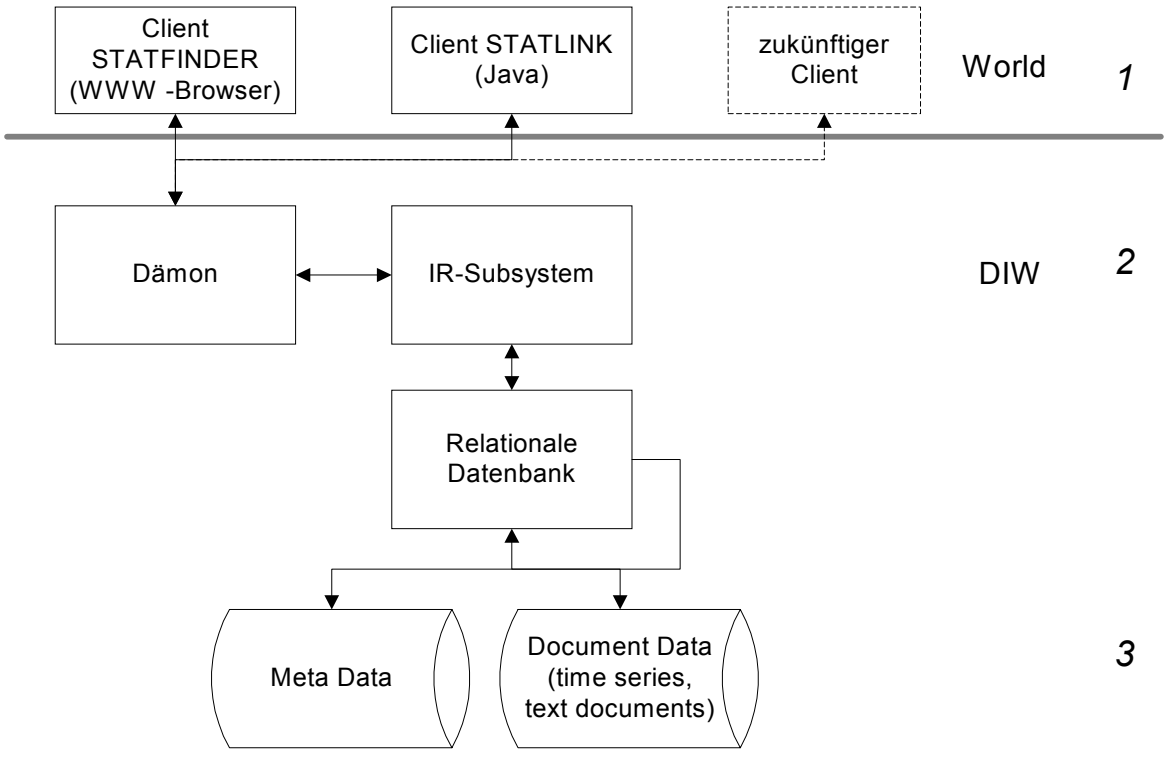
- es erleichtert die Wartung und Aktualisierung der gesamten Abfragekette. Die Entwicklung einzelner Komponenten kann unabhängiger voneinander stattfinden. Die Vermittlungskomponente kann erweitert werden, ohne „alte“ Clients vom Zugang auszuschließen.
- Die relativ einfache Schnittstelle erlaubt es Clientprogrammen, ohne her stellerspezische Datenbanktreiber oder -lizenzen, die Datenbank zu benutzen
- Clientprogramme müssen keine Datenbankabfragesprachen beherrschen und müssen nichts über die Interna der Datenbank wissen
- Anpassungen einzelner Komponenten der Serverseite (z.B. neues Serverbe triebsystem oder neue Datenbankreleases) führen nicht zu inkompatiblen Endbenutzerprogrammen.
- Diese Struktur erlaubt eine bessere Lastverteilung auf der Serverseite, da Datenbank und Vermittlungskomponente nicht auf der gleichen Maschine laufen müssen.
- Es werden einfachere Authorisierungsmethoden im DBMS benötigt.
- Die Trennung der Komponenten erlaubt einen einfachen Betrieb mit einem Internet-Firewall.

### 2.4.2. Datenbankserver (Dämon)

Der Dämon selbst deckt mehrere funktionale Bereiche ab. Der eine ist das Datenbankmanagementsystem (DBMS), der Kommunikations- und schließlich der Clientbereich. Die Bereiche sind auch in der folgenden Tabelle 29 zu finden.

| **DBMS** | **Kommunikation** | **Client** |
|---|---|---|
| SQL Dialekt | TCP/IP Sockets | Protokoll parser |
| Fehlerbehandlung (DB) | Fehlerbehandlung (Netzwerk) | Datenformatierung |
| | Logging von Benutzerdaten | |

**Tabelle 29: Funktionale Bereiche des Dämons**




Der Dämon muß einen sicheren Transport über das Netz gewährleisten. Es wurde kein eigenes Transportprotokoll verwendet, sondern auf das bewährte TCP/IP zurückgegriffen, das seit Jahrzehnten ein weltweit verfügbares Übertragungsprotokoll ist, speziell in Weitverkehrsnetzen. Die Netzwerkschnittstellen und -treiber sind leicht zu beschaffen und teilweise kostenlos. Da das Internet auf diesem Protokoll fußt, ist unser Konzept sozusagen „aus dem Stand“ internetfähig.
Der Protokollparser liest die Eingaben der Clients, prüft sie auf syntaktische

Richtigkeit und leitet sie an die entsprechenden Abfragemodule des Dämons weiter. Der Dämon ist vollständig datengetrieben, d.h. es können ständig neue Bestände in die Datenbank aufgenommen werden, ohne daß der Dämon in irgendeiner Weise davon betroffen wäre. Datenbestandsspezifische Informationen innerhalb des Dämons würden erneute Komplilation desselben nötig machen.
Der Parser ist relativ einfach erweiterbar und kann dann neue Abfragetypen weiterleiten. Die der Datenbank zugewandte Seite des Dämons bekommt vom Parser die Abfragen in der Reihenfolge des Eingangs durchgereicht und führt mit Hilfe einiger Metainformation die Abfrage durch. Die Umsetzung in den SQL-Dialekt der Datenbank erfolgt dynamisch, so daß auch hier kein Anpassungsbedarf besteht, sollten sich z.B die Knotentabellen ändern.

### 2.4.3. Sitzung (Session)

Der Server, konzipiert für den Einsatz in Weitverkehrsnetzen, bietet eine Art Request/ Response-Protokoll an. Clientprogramme sollen zunächst Informationen über die vorhandenen Daten sammeln, um in einem späteren Abfrageschritt die doch teilweise sehr umfangreichen Zeitreihen zu laden. Dieses Verfahren wurde gewählt, um bereits existierende, ältere Clientprogramme nicht von der Benutzung des Systems auszuschließen. Außerdem ist anzunehmen, daß der Zugriff auf die Datenbank nicht hauptsächlich über Hochgeschwindigkeitsnetze, sondern auch über Einwahlverbindungen via Modem oder ISDN, die mit einer Bandbreite von maximal 64k Bit/s arbeiten, erfolgt. Der STATLINK-Client verfolgt deshalb eine ähnliche Strategie wie der FAPSY-Client, mit dem Unterschied, daß er zum einen mehr Informationen auf „seiner“ Seite vorhält und damit den Benutzer auch ohne vorangehende Abfrage informieren kann, und zum anderen, daß er die Schritte Abfrage - Anwort - Abfrage etc. aus Benutzersicht zu einem Schritt verdichtet, in einer Sequenz also Informationen über Daten und die Daten selbst anfordert. Da der Dämon Abfragen in beliebiger Reihenfolge beantworten kann, ist dieses Verfahren problemlos möglich.

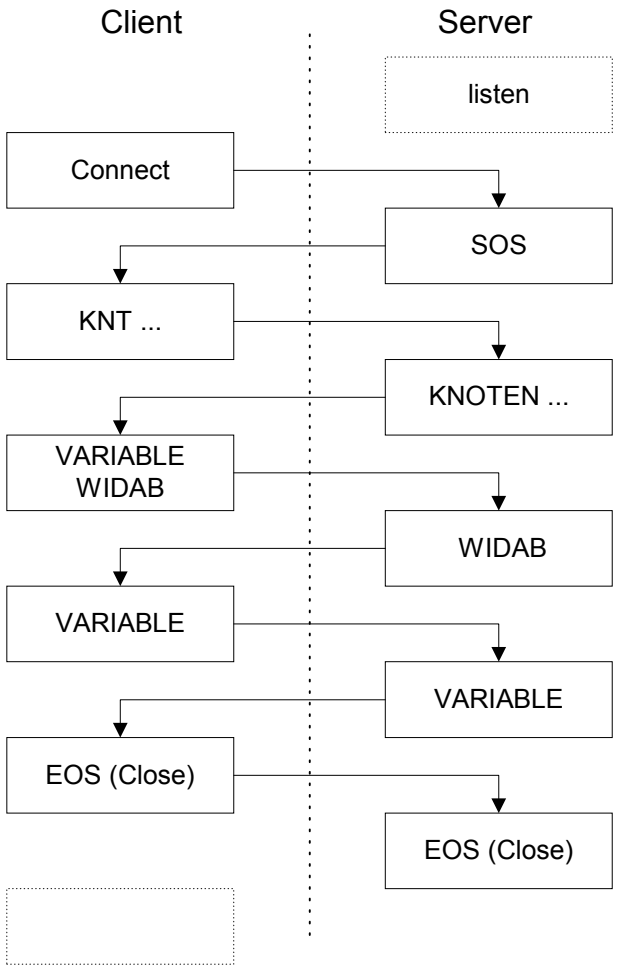



**Abbildung 15: Client/ Server Sitzung**

### 2.4.4. Protokoll: Anfragen an den Server

Abfragen an den Server müssen einem der folgende Schemata folgen:

- Knotenabfrage
- Baumabfrage
- Variablenabfrage (Dokumente)
- Variablendefinitionenabfrage
- Bauminhaltsabfrage

Der Austausch der Daten erfolgt über das Netz als Austausch von ASCII - Textsequenzen. Alle Protokollelemente beginnen mit einem Schlüsselwort, um den Typ der Abfrage zu signalisieren, mit eventuellen Parametern, danach folgen die Begriffe, die erfragt werden sollen. Zum Schluß einer solchen Abfolge wird ein Schluß- oder Protokoll-Ende-Wort gesendet, um das Ende der Abfrage zu signalisieren. Protokollelemente werden durch die Zeilensschaltung, einzelne Objekte in einer Zeile durch Leerraum (Leerzeichen oder Tabulator) getrennt.
Der Typ des Clientprogramms, d.h. seine Programmiersprache, seine Konzeption und seine Algorithmen sind für den Server irrelevant, jedes Programm, das die im folgenden aufgezeigten Abfragen zu generieren in der Lage ist, kann den Server befragen. Es existieren neben dem Statllink- und dem FAPSY-Client noch weitere Programme mit grafischen Benutzeroberflächen, die ausgesprochen primitiv (telnet) oder ausgeklügelt sein können (STATLINK). Alle diese Clients können mit

demselben Server auch zeitgleich kommunizieren, da die Kommunikationskomponente als Dämon ausgeführt ist und für jede Abfrageaktion eine neue Programminstanz startet. Abfragen können in beliebiger Reihenfolge gestellt werden, es ist dank des TCP/IP auch nicht unbedingt notwendig, erst eine Antwort abzuwarten, um die nächste Abfrage absenden zu können.
Der Dämon erweist sich durch diese Eigenschaften als flexibel, belastbar und leistungsfähig.

#### 2.4.4.1. Knotenabfrage

Dies ist zumeist die erste Abfrage einer Sitzung (auch: Session), die sich nicht der IR-bezogenen Verfahren und Daten bedient; sie gibt Auskunft über die zu bestimmten Kombinationen erhältlichen Dokumente.

```
KNT <NODETABLE> <TREE 1> [<TREE 2>] •••• [<TREE n>]
<KEY 1 TREE 1> [<KEY 1 TREE 2> ] •••• [<KEY 1 TREE n> ]
••
<KEY z TREE 1> [<KEY y TREE 2> ] •••• [<KEY z TREE n> ]
EOT
```

Es ist nicht nötig, eine Knotentabelle über alle ihre Baumattribute zu befragen. Werden eine oder mehrere Spalten ausgelassen, erhält man trotzdem die komplette Menge der Kombinationen.
Der Server bricht die Verbindung ab, wenn die angegebenen Attribute nicht in der angegebenen Knotentabelle vorhanden sind oder falsch angegeben wurden.
Falls die Anzahl der Parameter nicht mit der Anzahl der übergebenen Werte übereinstimmt, kommt es entweder zum Abbruch oder zu unsinnigen Antworten.

```
knt knt_zvei_produktion95 bam_zvei_pthem bam_zvei_pbran95
2 30
3 40
EOT
```

Es ist nicht unbedingt nötig, aber sinnvoll, die einzelnen Abfragekombinationen in einer eigenen Zeile anzugeben. Der Server liest Zeilen bis zu einer Länge von 512 Zeichen und wertet danach die Zeile aus. Theoretisch ist auch folgende Abfrage denkbar:

```
knt knt_zvei_produktion95 bam_zvei_pthem bam_zvei_pbran95
2 30 3 40
EOT
```

#### 2.4.4.2. Kategorie (Baum-)abfragen

Dies ist die ältere, etwas langsamere Art der Abfrage der einzelnen Kategorien und deren Kombinationen.

```
BAUM QRY <TREE 1> [<TREE 2>] •••• [<TREE n>]
<KEY 1 TREE 1> e [<KEY 2 TREE 2> e  ] •••• [<KEY a TREE n> e]
••
<KEY x TREE 1> e [<KEY y TREE 2> e ] •••• [<KEY z TREE n> e]
EOT
```

Hier ist zu beachten, daß „BAUM QRY“ das Schlüsselwort ist. Hier darf nur ein Leerzeichen zwischen „Baum“ und „Qry“ auftauchen, da es sonst nicht erkannt wird.
Ein Relikt aus frühen Entwicklungszeiten ist das „e“ nach jedem Schlüsselbegriff. Ursprünglich war daran gedacht, hier einstellen zu können, ob der Baum abgestiegen

werden soll oder nicht. Die Entwicklung der Knotenabfrage führte dazu, daß dieses Problem verschoben wurde. „Alte“ Clientprogramme wollte man nicht mehr ändern, also blieb diese Angabe erhalten.

```
baum qry bam_ifo_kthem bam_ifo_kbran
2 e 1 e
3 e 2 e
EOT
```

Der Server bricht die Verbindung ab, wenn Baumtabellen nicht vorhanden sind oder falsch angegeben wurden. Ein vergessenes „e“ hat den gleich Effekt.

#### 2.4.4.3. Variablenabfrage

Zur Zeit können drei verschiedene Variablentypen abgefragt werden. Der zulässige Zeichenvorrat für einen Variablennamen ist definiert als [0-9A-Za-z-_:@].

```
VARIABLE
<VARIABLE 1>
[<VARIABLE 2>]
••
[<VARIABLE n>]
EOT
```

So lautet allgemein eine Variablenanfrage, unabhängig von seinem Variablentyp.

##### 2.4.4.3.1. Zeitreihen

Die unten notierte Abfrage liefert die Werte der Zeitreihen zurück. Sie wird von den Clientprogramm meist als letzte Frage an den Server geschickt. Es kann allerdings Sinn machen, nur diese Abfrage zu Aktualisierungszwecken loszuschicken.
Nicht vorhandene Variablen werden nicht ausgegeben. Es wird keine Meldung erzeugt, die besagt „Variable XYZ nicht gefunden“.
Die Zeitreihen werden zwar chronologisch sortiert ausgegeben, Lücken in den Reihen werden aber nicht gekennzeichnet.

```
variable
GVR01600
GVR04120
H103_4
H3114DB
2000000:BUS
2690000:XES
2000000:BUS@OD
010AEGESNI
233UMINLRI
AB3611180
DM3633930
EOT
```

Wie man hier sehen kann, ist es möglich, die Bestände einer Domäne gemischt abzufragen. Dies gilt auch für die anderen Variablentypen, die in „2.4.4.3.2 Texte“ und „2.4.4.3.3 BLOb-Text-Dokumente“ aufgezählt sind.

##### 2.4.4.3.2. Texte

Der Variablentyp Text enthält eine begrenzte Anzahl von ASCII-Zeichen. Die Menge bewegt bleibt dabei unter ca. 3000 Zeichen.

2.4.4.3.3. BLOb-Text-Dokumente

Der Variablentyp Blobtext enthält eine Datenmenge, die durch die interne Definition von „Long Var Char“ der Datenbank definiert ist und bei maximal 2 GB liegt.

### 2.4.4.4. Variablendefinitionen

Die Abfrage der Variablendefinitionen ist häufig der zweite Schritt eines kompletten Retrievals. Nachdem man via Knoten- oder Baumabfrage die Zeitreihennamen erhalten hat, holt man sich die detaillierte Beschreibung der Dokumente.

```
VARIABLE WIDAB
<VARIABLE 1>
[<VARIABLE 2>]
••
[<VARIABLE n>]
EOT
```

Auch hier ist zu beachten, daß das Doppelwort „VARIABLE WIDAB“ das Schlüsselwort ist. Hier darf nur ein Leerzeichen zwischen „Variable“ und „Widab“ auftauchen, da es sonst nicht erkannt wird.

```
Variable Widab
GVR01600
GVR04120
H103_4
H3114DB
2000000:BUS
2690000:XES
2000000:BUS@OD
010AEGESNI
233UMINLRI
AB3611180
EUROTEXT1
EOT
```

### 2.4.4.5. Baumartig strukturierte Kategorien

Diese Abfrage dient der, meist intervallartigen Aktualisierung der lokalen Datenbanken der verschiedenen Client-Programme.
Ausgeben wird die komplette Kategorie nach Ebenen (d.h. nach Vorgängerzeigern) sortiert.

```
BAUM INHALT
<TREE 1>
[<TREE 2>]
••
[<TREE n>]
EOT
```

Auch hier ist der Schlüsselbegriff durch ein Leerzeichen getrennt.
Ein falsch benannter Baum führt nicht zum Abbruch des Programms, sondern nur zu einer „leeren“ Antwort.

```
baum inhalt
bam_income_sex
bam_vgr_konten
EOT
```

### 2.4.5. Protokoll: Antworten des Servers

Die Antworten des Servers können folgende Formen haben:

- Knotentabelle
- Baumantwort
- Zeitreihen
- Variablendefinitionen
- Bäume

**Hier gilt im Prinzip das in den Abfragen schon erwähnte Verfahren (s.a. Kapitel „0 Abbildung 15: Client/ Server Sitzung**

Protokoll: Anfragen an den Server“ Seite 44). Ein Schlüsselwort leitet die Antwort ein, eventuell nötige Parameter werden geliefert, danach wird die Antwortmenge in einem definierten Format geschickt und am Ende mit einem definierten Schlußwort abgeschlossen. Der Server trennt die Attribute einer Zeile mit einem Tabulatorzeichen. Der Server schickt die Antworten in der Reihenfolge des Eingangs. Die Antwortmenge ist nicht beschränkt, der Server gibt keine Auskunft über die zu erwartende Information.
Clientprogramme müssen somit auch auf große Datenströme eingerichtet sein. Es kannja durchaus auch „dumme“ Fragen geben. Wollte jemand z.B. die gesamte Importgliederung des ZVEI (in STATLINK leider nicht enthalten) abfragen, kannn er oder sie sich auf eine Größenordnung von mehreren hunderttausend Zeitreihen einstellen.

#### 2.4.5.1. Knotenantwort

Diese Antwort liefert alle Kombinationen zu einer Knotenabfrage, die zu einem „Treffer“ in der Datenbank führen.
Es ist wichtig zu beachten, daß das Anworttupel immer die kompletten Knotenattribute zurückliefert. Eine Abfrage einer Knotentabelle mit vier Atrributen liefert also immer sechs Spalten zurück, egal ob nur eine oder zwei Attribute konkret abgefragt wurden.

```
KNOTEN
variable knoten <BAUM 1> [<BAUM 2>] •••• [<BAUM n>]
[<VARIABLE 1> <KNOTEN 1> <KEY 1 BAUM 1> <KEY 1 BAUM 2> •••• <KEY 1 BAUM n>]
••
[<VARIABLE z> <KEY z BAUM 1> <KEY y BAUM 2> •••• <KEY z BAUM n>]
EOT
```

Das Ergebnis besteht aus dem Kopf und keiner oder mehrerer Ergebniszeilen. Ein EOT, welches unmittelbar auf die Kopfzeilen folgt, bedeutet also eine leere Menge.
Das nachfolgende Beispiel zeigt einen Ausschnitt aus der Knotentabelle des ZVEI Außenhandels.

```
KNOTEN
variable knoten bam_zvei_athem bam_zvei_abran bam_zvei_land_ber bam_zvei_land_par
X11000A__2__4 102093 2 3 5 8
X11000A__2__5 102736 2 3 5 10
X11000A__1__4 205464 2 3 9 8
X11000A__1__5 208045 2 3 9 10
EOT
```

### 2.4.5.2. Baumantwort

Die Baumabfrage und -antwort wird beispielsweise vom WWW-Client dieses Projektes benutzt, für den die HTML-Versionen der Kategorien nur nach Änderung neu erzeugt werden müssen. Das Verfahren ist für kleinere Bestände und auch zum Testen der Zuordnungen außerdem sehr gut. Es kann auch Bestände in der Datenbank geben, die nur eine Kategorie besitzen (z.B. die ältere VGR nach Konten, deren Regionenbaum für den öffentlichen Zugriff entfernt wurde) und für die die Generierung von Knotentabellen nicht nötig wäre.

```
BAUM ANTWORT
<BAUM 1> <KEY 1 BAUM 1> [<KEY 2 BAUM 2>] •••• [<KEY a BAUM n>]
[<VARIABLE 1> <LAST UPDATED 1> <HEADLINE 1>
[<VARIABLE 2><LAST UPDATED 2> <HEADLINE 2>]
••
[<VARIABLE n><LAST UPDATED n> <HEADLINE n>]]
EOT
[BAUM ANTWORT]
```

Das Schlüsselwort dieser Antwort ist „BAUM ANTWORT“, das Leerzeichen ist zu beachten. Im folgenden Beispiel sieht man ein Dialogfragment zwischen Client und Server.
Man kann sehen, daß die Gesamtantwort des Servers in einzelne Antwortsegmente unterteilt wird. Eine leere Antwortmenge besteht aus dem Schlüsselwort, der Wiederholung der Kategorie- (Baum-) namen und der erfragten Schlüssel, gefolgt von dem Schlußwort „EOT“.

```
baum qry bam_ifo_kthem bam_ifo_kbran
2 e 1 e
3 e 2 e
EOT
BAUM ANTWORT
bam_ifo_kthem 2 bam_ifo_kbran 1
2000000:GUS 03.06.1997 11:40:53 Geschäftsbeurt.(S)Verarb.Gew. 2000000:GUS@OD 03.06.1997
11:40:42 Geschäftsbeurt. (S)Verarb.Gew. EOT
BAUM ANTWORT
bam_ifo_kthem 3 bam_ifo_kbran 2
2000001:GES 03.06.1997 11:40:53 Geschäftserwart.(S)Verarb.Gew.(o.Nah
EOT
```

### 2.4.5.3. Dokumente

Im folgenden werden die Serverantworten bezüglich der ´Dokumente beschrieben. Die häufigsten Dokumente sind noch immer

#### 2.4.5.3.1. Zeitreihen

Die Zeitreihenwerte werden als „Variable-Zeit-Wert“ Tupel geliefert. Zur Zeit werden die Variablen in der Reihenfolge des Eingangs bearbeitet und zurückgeliefert. Es ist allerdings vorgesehen, daß der Server einige Optimierungen vornimmt, z.B. nach den Tabellen sortiert, die gemeinsame Zeitreihen haben. Dann würden zunächst alle Werte aus Bestand x, dann alle Werte aus Bestand y usw. geliefert.

```
VARIABLE
[<VARIABLE 1> <TIME 1> <VALUE>
[<VARIABLE 1> <TIME 2> <VALUE>]
••
[<VARIABLE 1> <TIME t> <VALUE>]
```

```
[<VARIABLE 2> <TIME 1> <VALUE>]
[<VARIABLE 2> <TIME 1> <VALUE>]
••
[<VARIABLE 2> <TIME t> <VALUE>]
••
[<VARIABLE n> <TIME t> <VALUE>]]
EOT
```

Eine leere Antwortmenge besteht aus dem Schlüsselwort „VARIABLE“ gefolgt vom Schlußwort „EOT“.
Die Zeitreihen werden „lückenlos“ ausgegeben, d.h. nicht vorhandene Werte werden nicht geliefert. Lücken in den Reihen müssen vom Client entsprechend erkannt und dargestellt werden.
Es gibt auch keine Unterscheidung zwischen Reihen unterschiedlicher Periodizität. Die Periodizität wird als „Anzahl der Beobachtungen pro Jahr“ mit den Reihenbeschreibungen geliefert.
Die Zahl der Nachkommastellen richtet sich nach dem Attribut „Komma“ in der WIDAB-Metatabelle. Üblich sind hier Angaben im Bereich von null bis zwei.

```
VARIABLE
085011099A3M 01.01.1992 73012
085011099A3M 01.02.1992 31161
085011099A3M 01.03.1992 31769
 .. ..
085011099A5W 01.07.1996 3625
085011099A5W 01.08.1996 1319
X11000A__1__4 01.01.1992 372264.00
X11000A__1__4 01.01.1993 301637.00
 .. ..
X11000A__2__5 01.01.1994 201731.00
X11000A__2__5 01.01.1995 257132.00
EOT
```

Das gekürzte Beispiel zeigt den Übergang zwischen verschiedenen Variablen mit unterschiedlicher Periodizität.

#### 2.4.5.3.2. Textdokumente

Im Gegensatz zu Zeitreihen, sind alle Arten von Textvariablen verschieden „groß“, sie sind nicht wie Zeitreihen durch einen bestimmte Länge (Float mit z.B. 2 Nachkommastellen) definiert.
Aus diesem Grund werden die „Werte“ eines Textdokuments mit einem einleitenden Wort (BOLABER) und abschließenden Wort gekennzeichnet (EOLABER).
Textdokumente, sind - wie schon der Titel sagt - echte Textstrings, d.h. sie enthalten keine Steuer- oder sonstigen Sonderzeichen.

```
VARIABLE
[<VARIABLE 1> <TIME 1> BOLABER <STRING 1> EOLABER]
[<VARIABLE 2> <TIME 2> BOLABER  <STRING 2> EOLABER]
••
[<VARIABLE 5> <TIME 5> BOLABER <STRING 5> EOLABER]
••
[<VARIABLE n> <TIME n> BOLABER <STRING n> EOLABER]
EOT
```

Eine leere Antwortmenge besteht aus dem Schlüsselwort „VARIABLE“ gefolgt vom

Schlußwort „EOT“.

```
VARIABLE
EUROTEXT1      08.02.1999
BOLABER
Regulation and Total <CY t 21>Productivity</CY> in <CY knt_eurostat_linda b 35>Electricity</CY>: A Comparison Between <CY knt_eurostat_linda bam_eurostat_country_en 11>Italy</CY>, <CY knt_eurostat_linda bam_eurostat_country_en 6>Germany</CY> and <CY knt_eurostat_linda bam_eurostat_country_en 9>France</CY>
Summary
Notwithstanding the
EOLABER
EOT
```

Im oben gezeigten Beispiel enthält das Textdokument bereits eine prototypenhafte Tagging-Informationen.

2.4.5.3.3. BLOb-Text-Dokumente

TextBLObs werden genauso wie Textdokumente geliefert.
Der einzige Unterschied ist in durch die Länge des Textdokuments gegeben.

### 2.4.5.4. Variablenbeschreibungen

Die Variablendefinitionen werden zeilenweise ausgegeben. Die maximale Zeilenlänge beträgt im Server zwar 999 Zeichen, allerdings sind die Datensätze in der Datenbank noch nicht so groß. Es können Zeilen mit einer Länge von etwa 270 Zeichen entstehen.
Die Größe einzelner Felder der Datenbank wird sich in Zukunft noch erhöhen, da besonders die Titel (das „headline“-Attribut) oft zu kryptisch sind und für die Darstellung in Grafiken oder Tabellen manchmal zu ungenau sind.
Beschreibungen zu angeforderten Variablen, die nicht in der Datenbank vorliegen, werden nicht ausgegeben, d.h. es erfolgt keine meldung darüber, daß eine Variable eventuell nicht gefunden wurde. Eine leere Antwortmenge besteht aus dem Schlüsselwort gefolgt von dem Schlußwort „EOT“.

```
WIDAB
[<VAR 1> <HEAD 1> <R_typ 1> <V_TYP 1> <DIM 1> <BEZ 1> <COMMA 1> <START 1> <PERIOD 1> <BEST 1> <TAB 1> <LAST UPDATED 1> <DOC TYPE 1>
[<VAR 2> <HEAD 2> <R_typ 2> <V_TYP 2> <DIM 2> <BEZ 2> <COMMA 2> <START 2> <PERIOD 2> <BEST 2> <TAB 2> <LAST UPDATED 2> <DOC TYPE 2>]
••
[<VAR n> <HEAD n> <R_typ n> <V_TYP n><DIM n> <BEZ n> <COMMA n> <START n> <PERIOD n> <BEST n> <TAB n> <LAST UPDATED n> <DOCTYPE n>]]
EOT
```

Leere Felder einer Relation werden leer ausgegeben, d.h. die entsprechenden Feldtrenner (Tabulator) werden ausgegeben, damit die Clientprogramme immer die gleiche Anzahl Spalten geliefert bekommen und daher leichter ihre internen Zuordnungen vornehmen können.
Das folgende Beispiel zeigt einen Ausschnitt aus einer Außenhandelsdatenbank am DIW

```
WIDAB
085011099A3M Ausfuhr Waren-Nr. 85011099, Land 3, Menge S O 1 Stück 0 01.01.1992 12.00 ZVEI-BRD-Ahandel tab_zvei_wnr_dat 18.01.1997 13:39:51 LTS
085011099A3W Ausfuhr Waren-Nr. 85011099, Land 3, Wert S O 1000 DM 0 01.01.1992 12.00 ZVEI-BRD-Ahandel tab_zvei_wnr_dat 18.01.1997 13:39:51 LTS
085011099A5M Ausfuhr Waren-Nr. 85011099, Land 5, Menge S O 1 Stück 0 01.01.1992 12.00 ZVEI-BRD-Ahandel tab_zvei_wnr_dat 18.01.1997 13:39:53 LTS
```

```
085011099A5W Ausfuhr Waren-Nr. 85011099, Land 5, Wert S O 1000 DM 0 01.01.1992 12.00 ZVEI-BRD-Ahandel tab_zvei_wnr_dat 18.01.1997 13:39:53 LTS
X11000A__1__4 Export Waren-Nr. 11000 Land __1 nach __4, in 1000 DM S A 1000 DM 1 01.01.1992 1.00 ZVEI-Aussenhandel tab_zvei_zv_basis 18.01.1997 02:56:21 LTS
X11000A__1__5 Export Waren-Nr. 11000 Land __1 nach __5, in 1000 DM S A 1000 DM 1 01.01.1992 1.00 ZVEI-Aussenhandel tab_zvei_zv_basis 18.01.1997 02:56:21 LTS
X11000A__2__4 Export Waren-Nr. 11000 Land __2 nach __4, in 1000 DM S A 1000 DM 1 01.01.1992 1.00 ZVEI-Aussenhandel tab_zvei_zv_basis 18.01.1997 02:56:26 LTS
X11000A__2__5 Export Waren-Nr. 11000 Land __2 nach __5, in 1000 DM S A 1000 DM 1 01.01.1992 1.00 ZVEI-Aussenhandel tab_zvei_zv_basis 18.01.1997 02:56:26 LTS
EOT
```

### 2.4.5.5. Baumartig strukturierte Kategorien

Diese Antwort liefert den Inhalt einer oder mehrere Kategorie- oder Baumtabellen mit der zusätzlichen Angabe, ob ein Baumeintrag einen Treffer liefert oder nicht. Sie dient üblicherweise der Aktualisierung von Clientprogrammen. Sie verschafft einen relativ schnellen Überblick über die „Belegung“ eines Baumes.
Die Reihenfolge der Ausgabe erfolgt nach Schlüsselnummern, es wird z.Z. keine Gliederungsreihenfolge geliefert. Clients müssen sich den Baum, den sie den Benutzern anbieten, mit eigenen Prozeduren gemäß den Vorgänger/ Nachfolgerbeziehungen traversieren und entsprechend grafisch generieren.

```
BAUMINHALT <TREENAME 1>
[<B1 KEY1> <B1 ANCS1> <B1 ENTRY1> <B1 CODE1> <B1 SEQ1> <B1 HIT1>
[<B1 KEY2> <B1 ANCS2> <B1 ENTRY2> <B1 CODE2> <B1 SEQ2> <B1 HIT2>]
••
[<B1 KEYn> <B1 ANCSn> <B1 ENTRYn> <B1 CODEn> <B1 SEQn> <B1 HITn>]]
EOT
[BAUMINHALT <TREENAME 2>]
••
EOT]
[BAUMINHALT <TREENAME n>]
••
EOT]
```

Abfragen zu Bäumen, die in der Datenbank nicht existieren, werden mit einer leeren Menge beantwortet, es wird das Schlüsselwort, der Baumname und das Schlußwort „EOT“ geliefert. Auf diesem Umweg läßt sich das Vorhandensein eines Baumes in der Datenbank testen.
Die Antworten werden in umgekehreter Reihenfolge des Eingangs ausgegeben. Das hat nur programmtechnische Gründe und beschleunigt den Programmlauf des Servers etwas.
Das folgende Beispiel zeigt eine Kategorie aus dem Bestand „WIDAB Brandenburg“.

```
BAUMINHALT bam_brand_region
1 0 Berlin B_ 0 1
2 1 Berlin-Ost BO 0 1
3 1 Berlin-West BW 0 1
4 0 Brandenburg C_ 0 1
5 0 Mecklenburg-Vorpommern D_ 0 1
6 0 Thüringen E_ 0 1
7 0 Sachsen F_ 0 1
8 0 Sachsen-Anhalt G_ 0 1
9 0 Ostdeutschland A_ 0 0
10 0 Niedersachsen H_ 0 1
11 0 Schleswig-Holstein I_ 0 1
12 0 Bremen J_ 0 1
```

```
13 0 Hamburg K_ 0 1
EOT
```

### 2.4.6. Technische Details des Dämons

Der Dämon wurde in C geschrieben und unter SCO-Unix kompiliert. Alle SQL-Anweisungen (ESQL) wurde mit Hilfe des ESQL-Precompilers compiliert.

### 2.4.7. Erweiterung des Standard-Dämons

Während des Projekts hatten wir die Entscheidung zu treffen, ob wir die gegebene Middleware, den Dämon, unverändert lassen und ihm eine parallel arbeitende IR-Komponente zur Seite stellen oder ihn komplett neu schreiben.
Wir entschieden uns für die erstere Möglichkeit, weil

- der Standard-Dämon als Produktionskomponente ausgereift war,
- die häufig prototypischen Entwicklungsschritte in reinem C unnötig aufwendig waren und
- langfristig eine völlig andere Lösung mit Hilfe von Java (RMI) angestrebt wird.

Dieser Lösungsweg ermöglichte uns eine flexible und verteilte Entwicklung der neuen Funktionalität. Der alte, der Standard-Dämon, deckt die Basisfunktionen zuverlässig ab, während der neue parallel bzw. als „Hülle“ um den alten Dämon arbeitet. Wie in der folgenden „Abbildung 16: Standard- und IR-Dämon im Zusammenspiel“ zu erkennen, kann der neue Dämon so benutzt werden, daß er vom Client als einzige Schnittstelle zur Datenbank benutzt wird. Nach außen hin präsentiert sich das System in der Abbildung 16 sichtbar, für den Client 2 wie für Client 1 mit nur einer Schnittstelle.
Faktisch werden für diese Lösung einfach zwei statt einer Portnummer verwendet. Abbhängig von derselben werden also vom System IR-Funktionalität oder nicht angeboten.

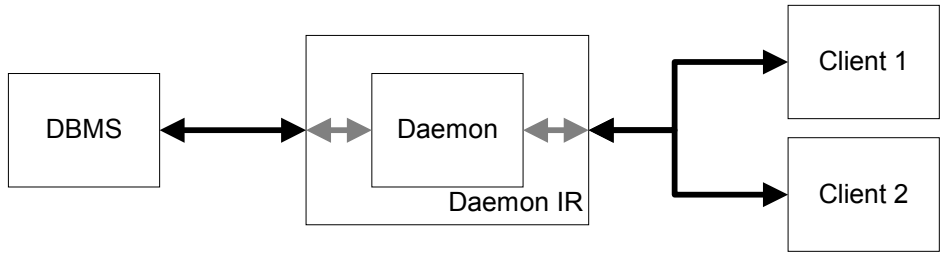



**Abbildung 16: Standard- und IR-Dämon im Zusammenspiel**

Tests dieser Lösung zeigen keinerlei Geschwindigkeitsnachteile, selbst wenn der neue Dämon nur als weiterer Vermittler zwischen Client und altem Dämon arbeitet und seine zusätzlichen IR-Funktionen nicht gebraucht werden.
Die eingebettete SQL-Umgebung funktioniert als standardisierende Schnittstelle zur Datenbank.

Zukünftig wird das ganze System Java-basiert arbeiten. Der Dämon selbst wird in Java geschrieben sein und den zukünftigen Java-Clients mit Hilfe von RMI (Remote Method Invocation = Entfernter Methodenaufruf) die alten und deutlich erweiterte Funktionalität zur Verfügung stellen.

Weitere konrekte Informationen zur Implementierung des Dämons sind im Kapitel 3.8 „Implementierung der Methoden im Dämon“ zu finden.

### 2.4.8. Erweiterung des Standard-Dämons, Protokollerweiterung

Die Erweiterungen des Dämonprotokolls durch den neuen IR-Dämon werden in den folgenden Kapiteln0 und 0 detailliert erläutert.
Grundsätzlich bestehen die Erweiterungen des Protokolls aus der Einführung zweier neuer Protokollwortbereiche.

#### 2.4.8.1. Protokollwortbereich „Schalter“

Der erste Protokollworttyp ist ein Schalter, der eine bestimmte IR-Funktionalität ein bzw. ausschaltet.

| **Protokollwortschalter** | **Funktion** |
|---|---|
| Irswitch | Ein- bzw. ausschalten der IR-Funktionalität |
| Irdebug | Ein- bzw. ausschalten des Debug- oder Überwachungsmodus |

**Tabelle 30: Übersicht über die neuen IR-Schalter**

Der *irswitch* schaltet den gesamten Prozeß des Information Retrieval überhaupt erst ein. Ist dieser Schalter nicht gesetzt, werden auch keine IR-Ergebnis während dieser Sitzung geliefert.
Der *irdebug* Schalter schaltet einen Debug-Modus ein, der den detaillierten Prozeß des Retrievals anzeigt. Dieser Schalter ist besonders dann nützlich, wenn man eine IR-Abfrage über einen textorientierten, einfachen Client wie z.B. telnet ausführt. Die Debug-Ausgabe (bei der Antwort des Servers) ermöglicht die Verfolgung des stattfindenden IR-Prozesses und gleichzeitig eine Übersicht über den Ergebnisraum, der im endgültigen Ergebnis in geplanter Weise nicht mehr enthalten ist.
Die Schalter finden ausschließlich bei der Anfrage an den Server Verwendung.

#### 2.4.8.2. Protokollwortbereich „XML“

Neben den Schaltern wird der IR-Prozeß mit Hilfe von XML-strukturierten Datenströmen initiiert als auch das Ergebnis ausgegeben.
Wir entschieden uns für eine XML-konforme Kommunikation, da diese eine saubere und standardisierte Möglichkeit der Befehlsstrukturierung ermöglicht. XML-Dateien werden u.a. auch deshalb zunehmend für verschiedenste Aufgaben verwendet, weil es bereits einige Werkzeuge und Programme gibt, die XML-Ströme interpretieren können.
In unserem Projekt kann die Ausgabe des IR-Retrievals aus der telnet-Sitzung in ein Textdatei - nach sehr geringen Modifikationen (Entfernen der protokolleigenen Syntax) - direkt in einem XML-fähigen Webbrowser ausgegeben werden, der die Ergebnisse dann baumartig und interaktiv benutzbar darstellen kann (Abbildung 24).
Mit relativ wenig Aufwand ließen sich mit Hilfe von DTDs (Document Type Descriptions) und/ oder Verwendung von XSLT-Prozessoren im Zusammenspiel mit der XML-Ausgabe wesentlich aussagekräftigere Darstellungen der Ergebnisse erzielen. Selbst unter Verwendung allgemein verbreiteter Werkzeuge, wie z.B. dem Internet-Explorer in Abbildung 24, läßt sich eine akzeptable Darstellung erreichen.
Damit erweist sich die XML-konforme Lösung als ausgesprochen einfach und flexibel.
Sowohl die Anfragen als auch die Ausgaben des IR-Dämons erfolgen, eingebettet in eine einfaches Protokoll, in der XML-Syntax.

### 2.4.8.3. Protokollerweiterung: Anfragen an den Server

Alle für den Standard-Dämon definierten Protokollwörter werden semantisch und syntaktisch genau so angenommen wie bisher.
Die neue IR-Funktionalität wird - wie oben bereits beschrieben – hauptsächlich durch XML-konforme Steuerströme vorgenommen.
Grundsätzlich erfolgt also eine IR-Anfrage an den Dämon nach dem Schema

- Schalterbereich
- Query-Bereich mit XML-Vorspann

Um also eine IR-Anfrage zu stellen, bekommt der Server den Datenstrom der Steuerinformationen folgendermaßen gesandt:

switch ir=*on*
switch irdebug=*on*
query free text
*xml-text query*
eot

**Abbildung 17: Schema einer IR-Anfrage**

Nach den Schaltern wird eine freie Textanfrage „query free text“ durch dasselbe Protokollwort initiiert. Die nun folgenden Anfragespezifierungen werden durch XML-Notation dargestellt. Der Abschluß der Anfrage wird mit Hilfe des Protokollwortes *eot* und das Ende der Serververbindung durch *eos* (wie schon beim Standard-Dämon-Verfahren) realisiert.

Der XML-Bereich der Anfrage, im obigen Beispiel durch den Alias *xml-text query* bezeichnet, enthält den interessanteren Hauptteil der IR-Anfrage.
Der XML-Teil der Anfrage gliedert sich in einen XML-Vorspann (Versionsinformation) und das IR-Dokument, das aus einem Kopf `<irheader>` und einem Dokumentteil `<irdoc>`.
Der Dokumententeil ist in Abbildung 18 und Tabelle 31 dargestellt.
Grundsätzlich funktionieren XML-Beschreibung mit Hilfe sogenannter Tags, die, anders als in HTML, einen öffnenden und einen schliessenden Teil zwingend verlagen.
Der gesamte IR-XML-Teil wird mit `<irdoc>` eröffnet und mit `</irdoc>` geschlossen.
Anschließend folgt ein Bereich zwischen `<irheader>` and `</irheader>`, der einige Meta-Informationen zum IR-Prozeß enthält.

```
<irdoc>
  <irheader>
    <usergroup>string</usergroup>
    <password>string</password>
    <language>[de|en|fr]</language>
  </irheader>
<querytext>string</querytext>
</irdoc>
```

**Abbildung 18: XML-Anfrage, Dokumententeil**

Danach folgt der eingentliche Query-Text, eingeschlossen von `<irheader>` und `</irheader>`. Nun kann der Dokumententeil mit `</irdoc>` abgeschlossen werden.
Die genannten Elemente finden sich in der folgenden Tabelle Tabelle 31 wieder.

| *Elemente (Tags)* | *Werteart* | *Bedeutung* |
|---|---|---|

| | | |
|---|---|---|
| **irheader** | - | Beginn/ Ende des XML-Kopfes |
| **usergroup** | String | Name der Benutzergruppe für diese IR-Anfragesitzung. Durch dieses Element wird die Benutzergruppe der Anfrage festgelegt und damit implizit der zur Verfügung stehende Datenpool. |
| | default=[default] | Falls diese Angabe ausgelassen wird, wird die Benutzergruppe “default” angenommen. |
| **password** | String | Passwort der Benutzergruppe für diese IR-Anfragesitzung. Durch dieses Element wird Rechtmäßigkeit für die gewählte Benutzergruppe der Anfrage festgelegt. |
| | default=[] | Wird kein Passwort angegeben, wird “” (kein Passwort) als Default angenommen |
| **language** | de= [german] en= [english] fr= [french] | Der IR-Vorgang läuft auf englischen und auf deutschen Texten. Es ist eine Option für französisch vorgesehen, für die bereits Metadaten vorhanden sind, allerdings ist der fehlenden Methoden für das Stemming nicht aktiviert. |
| | default=[de] | Falls diese Angabe ausgelassen wird, wird die Standardspracheinstellung “de” angenommen. |
| **Querytext** | String | Anfang und Ende des IR-Anfragetextes |

**Tabelle 31: XML-Anfrage, Dokumententeil, Elemente**

Das Element für die Benutzergruppe `<usergroup>` identifiziert die aktuelle Session dem Server gegenüber. Mit der Nennung einer Benutzergruppe ist sowohl die Identifizierung dem Server gegenüber möglich und damit verbunden der zu dieser Benutzergruppe gehörende Dokumentenpool festgelegt, also die Anzahl der Bestände bzw. Kategorien, die in den IR-Prozeß miteinbezogen werden. Das optionale Protokollwort `<password>` enthält das Passwort, das für die spezifizierte Benutzergruppe vergeben wurde. Stimmt das Passwort nicht mit dem auf dem Server hinterlegten überein, erhält der anfragende Client nur eine leere Antwort (eot). Der XML-Kopf enthält die Tag-Elemente `<usergroup>` und `<language>`, wie in Abbildung 19 zu sehen.

```
switch ir=on
switch irdebug=on
query free text
<?xml version="1.0" encoding="ISO-8859-1"?>
<irdoc>
<irheader>
<usergroup>statlink</usergroup>
<password>password</password>
<language>de</language>
</irheader>
<querytext>
Hier ist der anfragende Text
Enthalten.
</querytext>
</irdoc>
Eot
Eos
```

**Abbildung 19: XML-Anfrage an den Server, Beispiel**

Es wird der XML-Teil (XML-Instanz) mit dem XML-Vorspann eingeleitet. Neben der Version des verwendeten XML-Standards wird eine Kodierungsangabe benutzt, die die die innerhalb des XML-Konstrukts verwendete Zeichendefinitiontabelle nach ISO-

Standard (ISO-8859-1) festlegt. Diese beiden Teile des XML-Vorspanns sind sogenannte Attribute.

#### 2.4.8.4. Protokollerweiterung: Antworten des Servers

Alle für den Standard-Dämon definierten Protokollwörter werden semantisch und syntaktisch genau so behandelt wie bisher.
Auch die Antwort durch den Server findet ausschließlich durch XML-konforme Steuerströme statt.
Grundsätzlich erfolgt also eine IR-Antwort des Dämons nach dem Schema

- Response-Bereich mit XML-Verspann

Eine IR-Antwort wird vom Server in der folgenden Art gesendet:
*xml-text response*
eot

**Abbildung 20: Schema einer IR-Antwort**

Eine IR-Antwort wird also alleine durch die *xml-text response* und das abschließende Protokollwortes *eot* realisiert.
Es gibt immer nur eine gleichzeitige IR-Antwort/ Frage pro Session.

Der XML-Bereich der Antwort, im obigen Beispiel (Abbildung 20) durch den Alias *xml-text response* bezeichnet, enthält den wichtigen Hauptbestandteil der IR-Antwort.
Der XML-Teil der Antwort gliedert sich in einen XML-Vorspann (Versionsinformation) und das IR-Dokument, das aus nur aus einem Dokumentteil `<irdoc>` besteht.
Der Dokumententeil ist in Abbildung 18 und Tabelle 31 dargestellt.
Grundsätzlich funktionieren XML-Beschreibungen mit Hilfe sogenannter Tags, die, anders als in HTML, einen öffnenden und einen schliessenden Teil erzwingen.
Der gesamte IR-XML-Teil wird mit `<irdoc>` eröffnet und mit `</irdoc>` geschlossen.
Anschließend folgt ein Bereich zwischen `<irheader>` und `</irheader>`, der die Meta-Informationen zum IR-Prozess enthält. Hier finden sich die Listen zu den Domains, den Kategorien und den Termen, die im anfragenden Text für relevant befunden wurden.
Die Listen sind folgendermaßen aufgebaut
Zwischen `<taggedtext>` und `</taggedtext>` wird nun der ursprünglich anfragende Text zurückgegeben, angereichert mit den `<term>` und `</term>` Tags; diese Tags kennzeichnen die Strings innerhalb des Textes, die im Kopf `<irheader>` als ausgewählte Treffer aufgelistet sind. Dabei werden im Text nicht nur die identischen Terme aufgeführt, sondern auch alle durch Stemming und andere Annäherungen (z.B. Synonyme) gefundenen grammatischen Ausprägungen der Schlüsselwörter. Beispielsweise würden also die im anfragenden Text auftauchenden Zeichenketten Bergbau (Nominativ) und Bergbaus (Genitiv) durch dieselbe Termidentifizierung `<term id=1>Bergbau</term>` und `<term id=1>Bergbaus</term>` gekennzeichnet werden.

```
<?xml                    version="1.0"                    encoding="ISO-8859-1"?>

<irdoc>
 <irheader>
  <subdomainlist id=integer>
    <subdom id=integer>
      <name>string</name>
    </subdom>
  </subdomainlist>
```

```
  <catlist id=integer>
    <cat id=integer>
      <name>string</name>
    </cat>
  </catlist>

  <termlist id=integer>
    <term id=integer subdom=integer cat=integer key=integer order=integer>
      <name>string</name>
    </term>
  </termlist>
 </irheader>
<taggedtext>
   <term id=integer>nice term</term>
    .....
  </taggedtext>
</irdoc>
```

**Abbildung 21: XML-Antwort, Dokumententeil**

Nach dem Header folgt der markierte oder getaggte Text aus der Anfrage. Dieses XML-Elemente umschließt den ursprünglichen, anfragenden Text. Der ursprüngliche Text bleibt dabei soweit erhalten, daß er "wiedererkennbar" ist; bestimmte Sonderzeichen, die ausschließlich der Formatierung dienten, wurden allerdings entfernt. Für Details siehe Kapitel 3.4.2 Terme auf Seite 70.

| *Elemente (Tags)* | *Wert(e) bzw. Attribut = Wert* | *Bedeutung* |
|---|---|---|
| **irheader** | - | Beginn und Ende der Kopf-Information |
| **subdomainlist** | id= integer | Beginn und Ende der Subdomainliste mit einer Identifikationsnummer id *i* (aufsteigend ab 1)<br>Es ist bisher nicht vorgesehen, mehr als eine Subdomainliste als Antwort zurückzugeben. |
| **subdom** | id= integer | Beginn und Ende der Subdomain mit einer Identifikationsnummer id *i* (aufsteigend ab 1) |
| **name** | string | Name der Subdomain |
| **catlist** | id= integer | Beginn und Ende der Kategorienliste mit einer Identifikationsnummer id *i* (aufsteigend ab 1)<br>Es ist bisher nicht vorgesehen, mehr als eine Kategorienliste als Antwort zurückzugeben. |
| **cat** | id= integer | Beginn und Ende der Kategorie mit einer Identifikationsnummer id *i* (aufsteigend ab 1) |
| **name** | string | Name der Kategorie |
| **termlist** | id= integer | Beginn und Ende der Termliste mit einer Identifikationsnummer id *i* (aufsteigend ab 1)<br>Es ist auch hier bisher nicht vorgesehen, mehr als eine Termliste als Antwort zurückzugeben. |
| **term** | id= integer | Beginn und Ende des Terms mit einer Identifikationsnummer id *i* (aufsteigend ab 1) |
| | subdom= integer | Identifikationsnummer id *i* der korrespondierenden Subdomäne |
| | cat= integer | Identifikationsnummer id *i* der korrespondierenden Kategorie |
| | key= integer | Identifikationsnummer id *i* der korrespondierenden Satznummer (innerhalb der Kategorie) |
| | order= integer | Ordnungsnummer oder Plazierung des Terms, Relevanz des Terms im Verhältnis zu den restlichen Termen.<br>Die Werte werden aufsteigend vergeben. 1 ist die beste, n die schlechteste Plazierung. |
| **taggedtext** | | Beginn und Ende des getaggten (markierten) Textes |

| | | |
|---|---|---|
| **term** | id= integer | Beginn und Ende des Terms<br>Identifikationsnummer id *i* (aufsteigend ab 1) |

**Tabelle 32: XML-Antwort, Dokumententeil, Elemente**

Die Elemente in Tabelle 32 sind hierarchisch angeordnet. Dies ist an der Einrückung innerhalb der Tabelle kenntlich gemacht worden. So kann das Element `<subdomain>` nur innerhalb der `<subdomainlist>` vorkommen, diese wiederum nur im `<irheader>`.
Fast alle Elemente enthalten sogenannte Identifikationsnummer-Attribute. Diese dienen dazu, innerhalb eines bestimmten Elementtyps verschiedene Ausprägungen von Elementen darzustellen. So sind alle Listen aufsteigend numeriert, obwohl es in unserem Verfahren bisher nur jeweils eine Liste eines Typs gibt. Diese Definition ermöglicht allerdings eine unkomplizierte Erweiterung des XML-Datenstroms, beispielsweise um parallele Listen, die später als weitere Voting-Alternativen verwendet werden können; der Client könnte dann in einem lokalen Entscheidungsverfahren, u.U. abhängig von Benutzeranforderungen, eine andere Ergebnisauswahl präsentieren.
Innerhalb der `<termlist>` werden die Elemente `<term>` mit den Attributen `id`, `subdom`, `cat`, `key` und `order` geführt. Wie man besonders gut auch in der folgenden Abbildung 22 erkennt, werden damit für jeden Term seine relevanten Zuordnungen aufgeführt (nämlich die Subdomäne, die Kategorie und die entsprechende Satznummer). Dies führt dazu, daß ein Term in der `<termlist>` mehrmals vorkommen kann (Beispiel “years” aus Abbildung 22), wenn er z.B. für viele Satznummern einer Kategorie relevant ist. Dies hat den Vorteil, daß ein weiterverarbeitendes Clientprogramm relativ einfach diesen Datenstrom in interne Repräsentationen (Arrays, Listen, Pointer) transformieren kann. Nachteilig an diesem Verfahren ist, daß die entstehenden Textdatenmengen relativ groß werden können. Bei einer denkbaren technischen Perfektionierung des Protokollübermittlungsverfahrens, die sowohl eine Verschlüsselung als auch eine damit einhergehende Komprimierung zuließe, wäre dieser Nachteil jedoch sehr gering. Darüber hinaus ließen sich Verschlüsselung/ Komprimierung völlig unabhängig von dem IR-Verfahren und wohl auch mit gängigen Algorithmen realisieren.

```
<?xml                version="1.0"                encoding="ISO-8859-1"?>

<irdoc>
 <irheader>
  <subdomainlist id="1">
    <subdom id="1">
      <name>DIW-vgr</name>
    </subdom>
    <subdom id="2">
      <name>DIW-gvr</name>
    </subdom>
  </subdomainlist>

  <catlist id="1">
    <cat id="1">
      <name>bam_vgr_konten</name>
    </cat>
    <cat id="2">
      <name>bam_gvr_bla</name>
    </cat>
  </catlist>
```

```
  <termlist id="1">
    <term id="1" subdom="1" cat="1" key="1" order="2">
     <name> labour costs</name>
    </term>
    <term id="2" subdom="1" cat="1" key="2" order="3">
     <name>D-Mark</name>
    </term>
    <term id="2" subdom="2" cat="2" key="7" order="4">
     <name>D-Mark</name>
    </term>
    . . .
    <term id="5" subdom="2" cat="9" key="2" order="1">
     <name> macroeconomic developments</name>
    </term>
  </termlist>
 </irheader>
```

**Abbildung 22: XML-Antwort des Servers, `<irheader>`, Beispiel**

Das `<term>`-Element enthält zu den bereits erläuterten Attributen noch das Attribut `order`. Dieses bezeichnet die Relevanz des Terms im Verhältnis zu den anderen Termen, je niedriger der ganzzahlige Wert ist, desto relevanter wurde der Term bewertet. `Order` entspricht also einer Rangfolgezahl.
Die XML-Beispielantwort des Servers in Abbildung 22 und Abbildung 23 enthält einen englischen Text, um die Mehrsprachigkeit des Verfahrens zu demonstrieren. Die Anzahl der markierten, relevanten Terme wurde hier absichtlich niedrig gehalten, um die Anschaulichkeit zu erhalten.

```
 <taggedtext>
Over a period of more than two years, <term id="3">macroeconomic
developments</term> in Germany were characterised by rapid export growth.
A central factor behind this trend was the steady fall in the real
external value of the <term id="2">D-Mark</term> to the spring of last
year; the decline amounted to around 11% on the mid-1995 level. Low unit
<term id="1">labour costs</term>, reflecting a policy of wage restraint,
have contributed to the improvement in international competitiveness; they
have fallen by 3½% since their most recent peak at the end of 1995.
However, in the wake of the crises in Asia, Russia and Latin America,
which have led to a renewed slight increase in the real external value of
the D-Mark, goods exports began to show significant signs of a weakening
in the first half of last year; in the second half of the year - earlier
than had been widely expected - they fell sharply. <term id="2">D-
Mark</term> ...
 </taggedtext>
</irdoc>
```

**Abbildung 23: XML-Antwort des Servers, `<taggedtext>`, Beispiel**

Die XML-Antworten des Servers kann man, wie bereits in Kaptitel 2.4.8.2 Protokollwortbereich „XML“ auf Seite 54 beschrieben, mit einem XML-fähigen Programm anzeigen. In der Abbildung 24 erschließt sich besonders gut die Struktur der Antwort des Servers. Als sehr hilfreich erweist sich die interaktiv bedienbare Hierarchiestruktur der Antwort, die sich nach Bedarf entfalten und auch wieder zusammenfalten läßt.

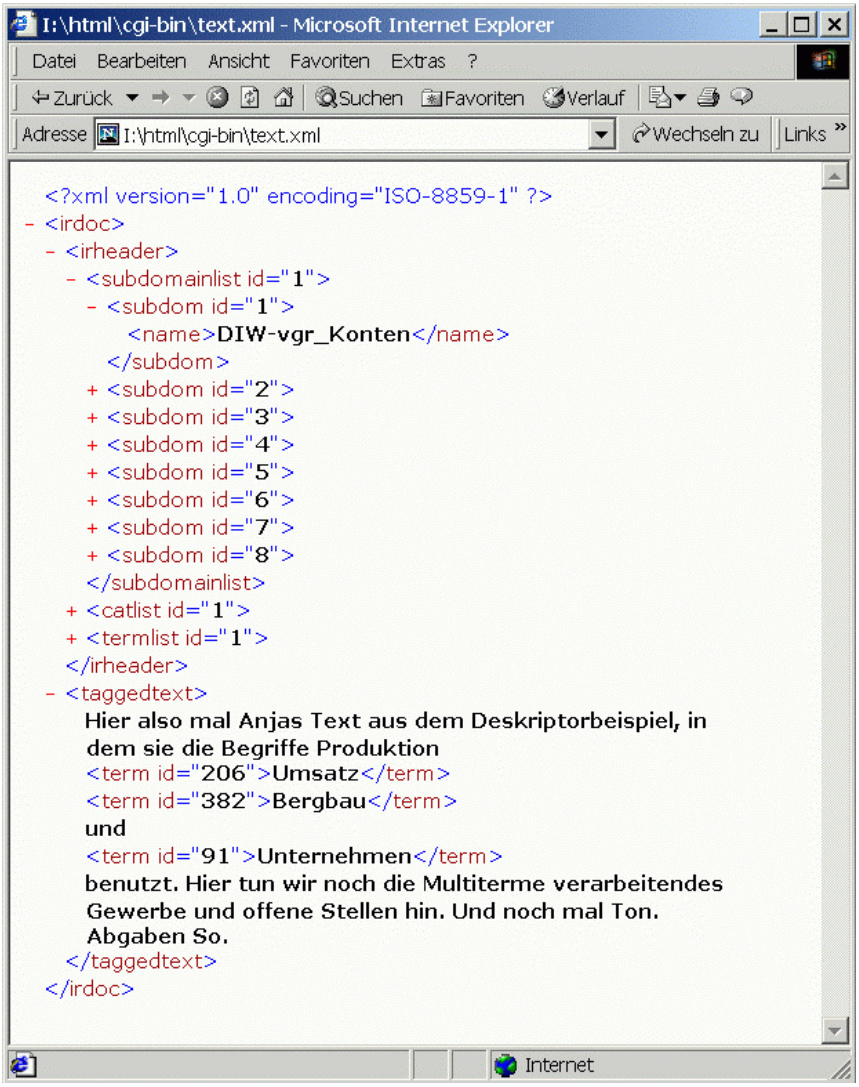



**Abbildung 24: IR-Dämon-Ausgabe in XML-fähigen Browser**

Natürlich läßt sich diese XML-Ausgabe auch in anderen XML-interpretierenden Programmen anzeigen.
Eine komplette Serverantwort findet sich im Anhang dieses Dokuments.

# 3. IR-Subsystem

## *3.1. Allgemeines*

Das IR-Subsystem beschreibt den gesamten algorithmischen Bereich der angewandten IR-Methoden. Es besteht aus den Funktionen des IR-Dämons und fußt inhaltlich auf die Metadatentabellen der Datenbank.
Das IR-Subsystem wurde eingeführt, um den standardisierten und bestandsbezogenen Zugriff auf die hauptsächlich statistischen Daten unserer Datenbank zu erweitern.
Es verhilft dem Benutzer mit Hilfe eines entsprechenden Clients dazu, Suchanfragen zu formulieren, die begriffliche Gleichheiten mit den statistischen Daten unserer Datenbank aufzeigen. Dies kann durch die Eingabe eines einzigen Stichwortes erfolgen oder aber durch die Formulierung eines ganzen Satzes bis hin zu einem vollständigen Textdokument. Bei sehr langen Textdokumenten kann die Antwortzeit des Dämons entsprechend ansteigen.

Die gefundenen Ergebnisse werden dem Benutzer im Kontext der vorhandenen Schemata (Bestände) gezeigt. Der Benutzer kann nun in einem der gefundenen Bestände seine Suche fortsetzen oder aber den Focus seiner Suchanfrage verändern, wenn der Richtung seines Ergebnisses nicht zufrieden war. Dazu formuliert wer eine neue Suchanfrage mit Hilfe eines reduzierten Suchtextes oder er navigiert innerhalb der Bestände direkt in den ihn interessierenden Kategorienbereich.
Zunächst führen wir hier Grundlagen und Überblick über das Gebiet des Information Retrieval ein, um dann die konkreten Funktionen des IR-Systems von STATLINK zu erläutern.

### *3.2. Einführung in das Information Retrieval*

Die rechnergestützte Auswertung von über einfache Datenbankanfragen hinausgehenden Anfragen an Textbibliotheken jeder Art wird Information Retrieval genannt [Salton 1983].
Der Teil der Informationsdienste, die mit Hilfe eines Information Retrieval Systems automatisiert wurden, ist von sehr komplexer Struktur, was wiederum an der komplexen Natur der zugrundeliegenden Daten liegt, die ein solches System verarbeiten muß. Eine Datenbank, die von einem IR-System gemanagt wird, besteht üblicherweise aus:

- einer Menge von Dokumenten
- einer strukturierten Menge von Hilfsdaten

Die Menge der Dokumente besteht üblicherweise aus einer großen Anzahl von textuellen Dokumenten.
Allgemein wird ein Information-Retrieval-Prozeß durch eine Person in Gang gesetzt, die ein bestimmtes Ziel verfolgt und hierfür ihr vorhandenes Wissen erweitern möchte[Steffens 1997].

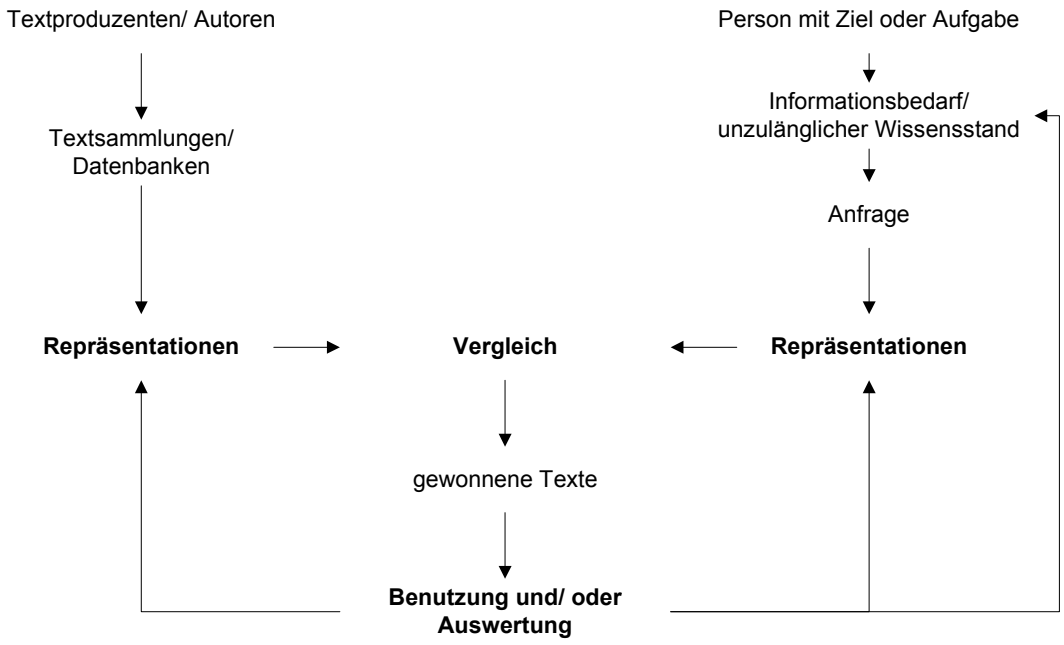



**Abbildung 25: Ablaufdiagramm des Informatin Retrieval nach [Belkin 1992]**

Zu diesem Zweck nutzt sie ein System, das auf Dokumentensammlungen zugreifen kann, die zuvor in rechner-adäquate Repräsentationen umgewandelt worden sind.

Die Anfrage der Person an das System wird nun ebenfalls in eine Rechnerrepräsentation umgeformt. Das System vergleicht Anfrage- und Dokumentenrepräsentationen und liefert eine Menge von Dokumenten zurück, die den Informationsbedarf des Anfragenden erfüllen sollen. Der Anfragende kann diese Dokumente verwenden, um die ihm gestellte Aufgabe zu lösen. In bestimmten Systemen kann er zusätzlich auch die Qualität des Retrieval-Ergebnisses bewerten. Die Konsequenz der Bewertung kann zum einen sein, daß der Anfragende seine Anfrage präzisiert, um so ein treffenderes Ergebnis zu erzielen. Zum anderen kann die Bewertung jedoch auch genutzt werden, um die Dokumentenrepräsentationen so zu verändern, daß das Retrieval grundsätzlich zu effektiveren Ergebnissen führt.
In einer rechneradäquaten Repräsentation der Textdaten liegt die Hauptanforderung beim Entwurf von Information-Retrieval-Systemen. Die natürlichsprachlichen Inhalte sowohl der in den Bibliotheken enthaltenen Dokumente als auch der Anfragen darauf sind formal nicht vollständig auf dem Rechner darstellbar.

So geht bei der Repräsentation von Dokumenten häufig der Kontext verloren, in dem bestimmte Begriffe verwendet werden. Der Begriff „Bank“ beispielsweise kann in einem finanztechischen Text eine andere Semantik aufweisen als in einem allgemeinen Artikel. Es ist daher wünschenswert, auch inhaltliche Zusammenhänge innerhalb von Dokumenten auf dem Rechner zu repräsentieren. Auf der Seite der Anfragen wachsen die Anforderungen an ein Information-Retrieval-System insofern, als daß der Anfragende nicht immer in der Lage ist, das ihn interessierende Themengebiet eindeutig abzugrenzen. Neben einer möglichst genauen Repräsentation der Anfrage ist es erforderlich, daß das System eine Benutzungsschnittstelle zur Verfügung stellt, die die Herangehensweise von Benutzern an ein Rechercheproblem berücksichtigt und somit eine intuitive Benutzung des Systems ermöglicht.

Um die Anforderungen an Information-Retrieval-Systeme möglichst weitgehend erfüllen zu können, wurden verschiedene Konzepte entwickelt, die mit den Eigenheiten der natürlichen Sprache befaßt sind [Broglio 1994]. So werden bei der Stammformreduktion Wortflexionen durch den entsprechenden Wortstamm repräsentiert. Bei der Stoppwortelimierung werden hochfrequente Wörter,
wie z.B. „der“, „doch“ oder „von“, für das Retrieval nicht berücksichtigt. Die Begriffserkennung ermöglicht die Einbeziehung von Konzepten, wie das Erkennen von Firmen- oder Personennamen. Bei der Feldindizierung können die durchsuchten Dokumente in kleinere Abschnitte, wie z.B. Titel und Haupttext, unterteilt werden. Die phonetische Suche ermöglicht das Auffinden von Begriffen, die nicht in der korrekten Schreibweise angefragt wurden. Weitere, hier nicht genannte Konzepte sind vorstellbar.
Viele dieser Konzepte finden sich in Gesamtmodellen wieder, die festlegen, auf welche Weise die Repräsentation von Texten und die Auswertung von Anfragen insgesamt zu erfolgen hat [Turtle 1992]. Ziel solcher Modelle ist es, Information Retrieval so zu beschreiben, daß es in einer Form ausgeübt werden kann, die der menschlichen Auffassung dieses Prozesses zumindest annähernd entspricht.

Die Modelle des Information Retrieval lassen sich in zwei Kategorien unterteilen, siehe Abbildung 26: Modelle des Information Retrieval.

<table>
<tr><td colspan="3">Information- Retrieval- Modelle</td></tr>
<tr><td>Exact-Match-Modelle</td><td colspan="2">Best-Match-Modelle</td></tr>
<tr><td rowspan="3">- Dokumente sind relevant oder nicht<br>- Abstufungen sind nicht möglicht</td><td colspan="2">Grad der Relevanz für jedes Dokument</td></tr>
<tr><td>Vektorraummodell</td><td>probabilistische Modelle</td></tr>
<tr><td>- Repräsentation durch Vektoren<br>- Abstufung über Distanz der Vektoren</td><td>- stochastische Verfahren<br>- Abstufung über Wahrscheinlichkeiten</td></tr>
</table>

**Abbildung 26: Modelle des Information Retrieval**

Exact-Match-Modelle legen für jedes Dokument einer Dokumentendatenbank fest, ob es für eine Anfrage relevant ist oder nicht. Diese Vorgehensweise spiegelt die Realität jedoch nur lückenhaft wider: Ein Begriff kann in einem Dokument das zentrale Thema sein, während er in einem anderen nur am Rande erwähnt wird. Abstufungen dieser Art sind in Exact-Match-Modellen nicht vorgesehen. Die Anfrageergebnisse von Best-Match-Modellen dagegen beinhalten für jedes Dokument einen Grad, zu dem dieses Dokument der Anfrage genügt. Die Differenzierung, die diese Modelle vornehmen, entspricht dem intuitiven Blick auf das Information Retrieval eher als die zweiwertige Logik der Exact-Match-Modelle.
Best-Match-Modelle lassen sich nach der Art der Repräsentation und Auswertung von Texten und Anfragen weiter aufgliedern in Vektorraum- und probabilistische Modelle.

#### 4.3.4.1 Vektorraum-Modelle

In Vektorraum-Modellen werden Dokumente und Anfragen als Vektoren im n-dimensionalen Raum repräsentiert. Jede Dimension stellt dabei eine Eigenschaft der Texte dar. Eine einfache Eigenschaft ist das Vorkommen eines bestimmten Begriffes. Es lassen sich jedoch auch beliebige andere Eigenschaften modellieren. Der Wert eines Vektors für die Dimension k beschreibt den Grad, zu dem der entsprechende Text die Eigenschaft k erfüllt. Bei der Auswertung von Anfragen an ein solches System wird die Distanz zwischen dem Anfragevektor und den einzelnen Dokumentenvektoren errechnet. Von dem Dokument, dessen Vektor den geringsten Abstand zum Anfragevektor besitzt, wird angenommen, daß es für die Anfrage am relevantesten ist.

#### 3.2.1.1. Probabilistische Modelle

Probabilistische Modelle bilden die semantische Unsicherheit des Retrieval-Prozesses ab, indem sie mit Hilfe von stochastischen Verfahren die Wahrscheinlichkeit ermitteln, daß ein bestimmtes Dokument für eine bestimmte Anfrage Relevanz besitzt [Fuhr 1994].

### 4.3.3 Bewertungskriterien im Information Retrieval

Um die Qualität der verschiedenen Modelle beurteilen zu können, werden für das Information Retrieval die Bewertungskriterien des Rücklaufs und der Genauigkeit definiert [Frei 1991]:
Der Rücklauf gibt die Fähigkeit eines Systems an, nützliche Dokumente zu finden. Hierzu wird der Quotient aus der Anzahl der gefundenen relevanten Dokumente und

der Anzahl der in der Textsammlung befindlichen relevanten Dokumente gebildet.
Die Genauigkeit stellt dar, wie konsequent „nutzloses“ Material ausgeschlossen wird. Dieses wird durch den Quotienten aus der Anzahl der gefundenen relevanten Dokumente und der Gesamtanzahl der gefundenen Dokumente zum Ausdruck gebracht.

### 3.2.2. Konkrete Information-Retrieval-Systeme

Verschiedene Information-Retrieval-Systeme kommen in der Praxis zum Einsatz. Nachdem hier zunächst Systemen, die nach Exact-Match-Modellen arbeiten, der Vorzug gegeben wurde, werden sie gegenwärtig von Systemen, die dem Best-Match-Prinzip folgen, abgelöst.
In dieser Kategorie sind unter anderem die Systeme WAIS 8 (Wide Information Server) und INQUERY zu nennen:

Das System WAIS wurde von der Firma Thinking Machines entwickelt. Es hat das Vektorraum-Modell zur Grundlage. Die Client-Server-Architektur des Systems ermöglicht eine verteilte Nutzung und damit auch die Suche auf Texten, die nicht lokal vorhanden sind [Pfeiffer 1995].

INQUERY wurde im Information Retrieval Laboratory der Universität von Massachusetts entwickelt. Es folgt dem probabilistischen Modell der Inferenznetze [Callan 1991]. Inferenznetze besitzen den Vorteil, daß sie erweiterbar sind, so daß INQUERY mit verschiedenen Textrepräsentations und Anfragekonzepten arbeiten kann [Belkin 1992].

### 3.2.3. Forschungsaspekte im Umfeld von Information Retrieval

Information-Retrieval-Systeme für sich genommen stellen solide Lösungen für Rechercheprobleme dar. Bisher gab es jedoch weitere, pragmatischere Anforderungen im Zusammenhang mit dem Information Retrieval. Diese rücken durch das wachsende Potential von Softwaresystemen in den Bereich des Möglichen [Croft 1995]. Solcherlei Anforderungen und mögliche Lösungsansätze werden am Arbeitsbereich Datenbanken- und Informationssysteme des Fachbereichs Informatik der Universität Hamburg
untersucht [Steffens 1996].

Information-Retrieval-Systeme bestanden bis jetzt unabhängig von anderen Systemen. Oft kann jedoch die Retrieval-Funktionalität sinnvoll mit den Diensten anderer Systeme ergänzt werden. Die angestrebte Lösung ist die Kombination von Information-Retrieval-Systemen mit verschiedenen Softwarekomponenten unter einer Benutzungsschnittstelle.

Bei vielen Information-Retrieval-Systemen stand bislang die Modellierung im Vordergrund. Ihre Oberflächen sind daher selten benutzerfreundlich gestaltet. Die Integration von Information Retrieval mit Fenstersystemen wäre hier von Vorteil, zumal sich dadurch Chancen ergeben, dem Anfragenden semantische Zusammenhänge graphisch verständlicher darzustellen.
Die Tatsache, daß durch Information Retrieval Texte direkt zur Verfügung stehen, erzeugt das Bedürfnis, diese Texte mit Hilfe desselben Softwaresystems auch verändern zu können. Hierdurch ist implizit eine Kombination von Information Retrieval und Texteditorsystemen gefordert.

Information-Retrieval- und Datenbanksysteme erfüllen, grob betrachtet, eine ähnliche Aufgabe, nämlich die Suche nach Daten. Daher läßt sich fordern, daß mit einem System sowohl nach Daten mit festgelegtem Feldformat, wie zum Beispiel den Daten eines Angestellten in einer Personaldatenbank, und nach Texten, die keiner zuvor festgelegten Form entsprechen, gesucht werden kann [Croft 1995a].

#### 3.2.4. Persistenz im Information Retrieval

Arbeitsschritte im Information Retrieval sowie ihre Ergebnisse sind häufig über eine Sitzung hinaus für den Benutzer interessant. Daher kann man es ermöglichen, Anfragen und ihre Ergebnisse dauerhaft (persistent) zu speichern. Hierdurch könnte beispielsweise ein Benutzer regelmäßig dieselben Anfragen an eine sich schnell verändernde Textsammlung stellen. Kataloge mit Standardanfragen an Textbestände wären ebenso denkbar wie eigene „Bibliotheken“ für einzelne Benutzer, in denen sie ihre Anfrageergebnisse ablegen und in denen sie auch wiederum suchen könnten. Beispiele finden sich in [Matthes 1993].

#### 3.2.5. Einbindung mehrerer verteilter Information-Retrieval-Systeme

Durch globale Netzwerke, wie z.B. das Internet, steht jedem Teilnehmer eine Vielzahl von Information-Retrieval-Systemen, die auf verschiedene Textsammlungen zugreifen, zur Verfügung. Es ist jedoch mühsam, für jede Sammlung eine unabhängige Anfrage abzusetzen. Die Lösung ist daher ein System, das nur einmal eine Anfrage entgegennimmt, und von sich aus alle ihm zugänglichen oder vom Benutzer ausgewählten Textdatenbanken befragt.

#### 3.2.6. Erweiterte Information-Retrieval-Anforderungen

Klassische Retrieval-Funktionalität kann um oben genannte Aspekte erweitert werden, indem sie in allgemeine Softwareumgebungen eingepaßt wird. Auf diese Art können Systeme so zusammengefaßt werden, daß sie für einen Anwendungsprogrammierer sämtlich zugreifbar sind und dem Benutzer als ein Gesamsystem erscheinen. Hierzu muß man geeignete Schnittstellen und Sprachkomponenten finden, die IR-Systeme, Datenbanken und weitere Systemelemente verbinden. Ein Beispeil für eine solches System ist Tycoon (Typed communicating objects in open environments) [Matthes 1994].

#### 3.2.7. Zusammenfassung

Die inhaltliche Suche auf Textdaten mit Rechnerunterstützung ist aufgrund der Vielfalt natürlicher Sprache nur mit Einschränkungen möglich. Es wurden jedoch verschiedene Konzepte und Modelle entwickelt, die eine zufriedenstellende Annäherung realisieren können. Aus dem konsequenten Einsatz von Informationstechnologie können sich aber noch weiterreichendere Vorteile für den Sektor des Information Retrieval ergeben. Die Integration mit anderen Softwarediensten, die Nutzung von Persistenz und die Durchführung von verteilten Retrieval-Prozessen sind Überlegungen in diese Richtung, die sich in Softwareentwicklungsumgebungen, wie sie momentan zur Verfügung stehen, realisieren lassen.

### *3.3. IR Funktionen von STATLINK*

In STATLINK haben wir einen speziellen Fall von Dokumenten zu betrachten. Die am häufigsten enthaltenen Dokumente sind statitische Zeitreihen. Diese sind an sich in keinen IR-System benutz- oder zugreifbar, weil sie keinen „Text“ darstellen. Jede Zeitreihe besitzt allerdings eine Zeitreihenbeschreibung, die als „Text“ betrachtet

werden kann. Wir bereichern also die Definition der klassischen Texte des Information Retrieval.

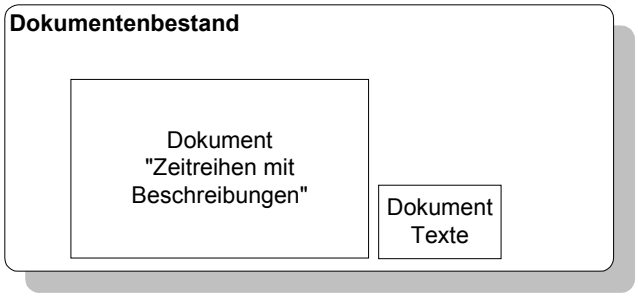



**Abbildung 27: Dokumententypen in STATLINK**

Die Größenverhältnisse der „Abbildung 27: Dokumententypen in STATLINK“ in zeigen deutlich an, daß die überwiegende Mehrheit der Dokumente unserer Datenbasis vom Typ „Statistische Zeitreihe“ ist. Textdokumente sind vorhanden, stehen aber nicht im aktuellen Fokus dieses Projektes.

Die IR-Funktionen des IR-Subsystems von STATLINK dienen dazu

- eine eintreffende Textanfrage eines Clients zu auszuführen und
- und einen entsprechend den IR-Ergebnissen angereicherten Text wieder zurückzusenden

Die IR-Funktion nimmt also Anfragen in Form eines Textes entgegen. Diese Texte können einfache Fließtexte (Abstracts, Zeitungsartikel), Tabellen oder sonstige in Textform existierende Dokumente sein. Zu beachten ist allerdings, daß die Formatierung von Anfragetexte bei der Rückgabe geringfügig verändert sein kann. Dies ist bereits dadurch der Fall, daß Fundstellen im Text ausgezeichnet werden müssen („Tagging“) und daß diese Auszeichnungen das Layout, bespielsweise die Spalten von Tabellen, beeinträchtigen können. Siehe hierzu Kapitel 3.4.2 Terme auf Seite 70.

Das Hauptziel aller IR-Prozesse des Systems ist das Abwägen oder Gewichten der Terme eines Textes der Benutzeranfrage – in Hinblick auf die in unserer Datenbank liegenden Dokumente. Diese Dokumente stammen aus dem wirtschafts- und wirtschaftswissenschaftlichen Umwelt.

Dies bedeutet praktisch eine Verlagerung der Funktionalität klassischer Information-Retrieval-Systeme. Im klassischen Sinne werden hauptsächlich Textbibliotheken in Datenbanken gehalten, die dann vom Benutzer möglichst zielgerichtet abgefragt werden. In Falle von STATLINK entspricht auch der anfragende Text des Benutzers einem Textdokument des IR. Dieses selbst wird mit den unten beschriebenen Methoden des Information-Retrieval verarbeitet.

Die Terme des anfragenden Textes werden nach einem Stimmverfahren (Voting) hinsichtlich ihrer Relevanz unserer Zeitreihendokumente bewertet und mit einem Gewichtungswert versehen und dann ihrer Gewichtungsreihenfolge nach ausgezeichnet oder markiert.
Bevor dieses eingentliche Verfahren beginnen kann, sind allerdings einige vorbereitende Prozesse nötig, um den anfragenden Text „voten“.

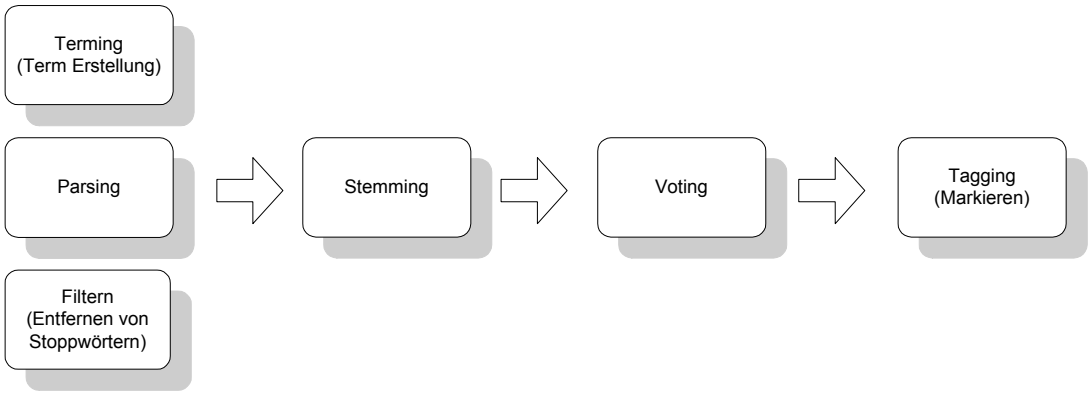



**Abbildung 28: Schematischer Aufbau des IR-Prozesses**

Die folgende Abbildung 28 gibt einen Überblick über den Gesamtprozeß des IR während der Anfrage an das STATLINK-System.
Terming, Parsing, Filtering und Stemming sind vorbereitende, das Voting ist die zentrale und das Markieren die abschliessende Funktion des IR-Prozesses.

- Das **Terming** erkennt Wortgrenzen, indem es Satzzeichen wie Komma, Punkt, Semikolon oder Wortseparatoren wie Leerzeichen und Bindestrich zur Hilfe nimmt. Innerhalb dieses Prozesses ist es werden auch Multiwort-Terme identifiziert. Die Syntax dieser Terme wird möglichst beibehalten.
- **Parsing** bedeutet den Prozeß des Extrahierens und Speicherns unterschiedlicher Worte
- Das **Filtern** ist eine Prüfung auf grundsätzliche Relevanz eines Wortes. Es wird entschieden, ob dieses überhaupt die weitere IR-Prozesse durchlaufen kann oder nicht. Beispielsweise werden ein- und zweibuchstabige Wörter oder als irrelevante Abkürzungen erkannte Worte (wie in, an, um) nicht mehr berücksichtigt. Wurde dieser Prozeß erfolgreich durchlaufen, folgt eine linguistische Analyse mit Hilfe des
- **Stemming**s. Es bezeichnet die Ableitung der Stamm- oder Grundform eines Wortes. Sie wird für die deutsche und die englische Sprache mit Hilfe des Porter-Stemm-Algoritmus durchgeführt, ein weltweit etabliertes Verfahren, das für verschiedene Programmiersprachen frei verfügbar ist. Eine französische Variante dieses Stemmalgorithmus ist bisher nicht entwickelt worden.
- Das **Stimm-** oder **Voting-Verfahren** meint den Prozeß der Abschätzung und der Definition der Relevanz eines Wortes oder Terms im Kontext der Anfrage. Das Voting in unserem System ist ein klassisches numerisches Verfahren, in dem Treffervergleiche ausgeführt und akkumuliert werden. Diese Methode ist durch unsere präzise beschriebenen und erstellten Linkinformationen möglich, die per Hand und intellektuell erstellt wurden und einen definierten und ganzheitlichen Charakter bezüglich unserer verwendeten ökonomischen Modelle (der Bestände) haben.
- Das **Markier**- oder **Tagging**- Verfahren bezeichnet den abschließenden Prozeß der Verankerung der IR-Ergebnisse im zurückzuliefernden Text. Dies geschieht, wie in 2.4.8.2 Protokollwortbereich „XML“ und 2.4.8.4 Protokollerweiterung: Antworten des Servers ausführlich geschildert, nach dem XML-Standard.

Die Abbildung 29 zeigt einen detaillierten IR-Gesamtprozeß, seine Funktionen und die entstehenden Zwischenergebnisse. Der IR-Prozeß ist sequentiell organisiert.

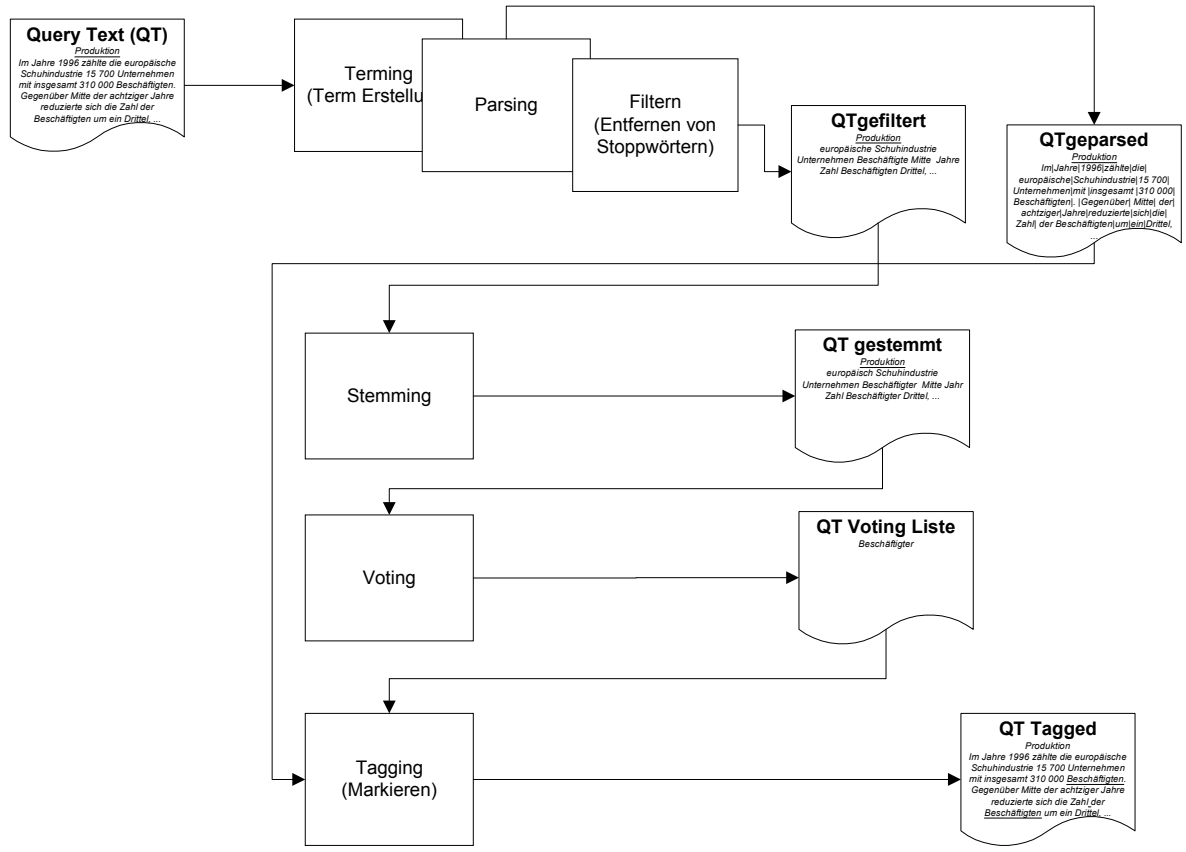



**Abbildung 29: Übersicht über den IR-Prozeß**

Entstehende Zwischenergebnisse, gekennzeichnet durch den bogenförmig abschließenden Kasten in der Abbildung 28, werden u.U. länger aufbewahrt und erst in einem späteren Prozeßschritt verwendet. Beispielsweise wird „QT parsed“, erzeugt durch die Parsing-Funktion, erst später in der Tagging-Funktion wiederverwendet. Dort nämlich werden zu den dann gevoteten Termen die ursprünglichen syntaktischen und grammatischen Instanzen wieder rekonstruiert und anschließend durch die Markierung kenntlich gemacht.

## *3.4. Terming, Parsing und Filtern*

### 3.4.1. Synonyme

Synonyme werden außerhalb des aktuellen IR-Prozesses vorbereitend in die Schlagwortliste mit denselben Referenzen wie ein Originalwort aufgenommen. Sie stehen quasi parallel zu ihren Originalwörtern gleichberechtigt in dieser Liste.
Die Synonyme wurden aus dem tatsächlich zur Verfügung stehenden Dokumentenbestand erstellt. Sie sind also konkret definiert und gebunden an den wirtschaftlichen Kontext unserer Dokumente.
Die untenstehende Tabelle 33 zeigt einen kleinen Ausschnitt aus dem Rohdatenfile der Synonyme.

| | |
|---|---|
| alte Bundesländer | früheres Bundesgebiet |
| ABL | früheres Bundesgebiet |
| ABL | alte Bundesländer |
| neue Bundesländer | neue Länder |
| NBL | neue Länder |
| NBL | neue Bundesländer |
| Deutschland | Bundesgebiet |

**Tabelle 33: Teil der Synonymliste (Rohdaten)**

Hier ist deutlich zu sehen, daß bestimmte Schlagwörter mehr als ein Synonym haben (ABL = alte Bundesländer und früheres Bundesgebiet). Gleichzeitig ist es sinnvoll, bestimmte Kurzformen, wie hier ABL, in die Synonymaliste aufzunehmen, um deren Sonderbedeutung frühzeitig im IR-Prozeß anzuzeigen.

### 3.4.2. Terme

Wörter sind durch Zwischenräume und Trennzeichen abgegrenzt [Drosdowski 1984]. Terme definieren wir hier als ein oder mehrere zusammenstehende Wörter.
Der Information- Retrieval- Prozeß versucht, anhand von definierten Trennzeichen eine sinnvolle Zerlegung des Gesamttextes. Tabelle 34 zeigt die Trennzeichen, die bei der Separierung verwendet werden.

| **Trennzeichen** | **Erläuterung** |
|---|---|
| Leerzeichen, TAB(*) | Syntaktische Worttrennzeichen, „White Space" |
| - | Bindestrich |
| = + * / | Mathematische Operatoren |
| ; . , : ! ? | Satzzeichen |
| ( ) [ ] | Klammern |
| ´ ` ' " | Anführungszeichen |
| \n | Zeilenumbruch |

**Tabelle 34: Term-Separatoren**

(*) Tabulatoren werden in diesem Prozeß unwiderruflich gelöscht!

Ursprünglich sollte der Text, der durch das Terming verarbeitet wird, vollkommen unverändert bleiben. Das erwies sich allerdings im Laufe der Entwicklungen als schwierig, da bestimmte Zeichen, besonders der Tabulator, der z.B. in großen statistischen Tabellen vorkommt, die Ausgabe des später markierten Textes fast unmöglich machen können. Aus diesem Grund wird der Tabulator bis zur Einführung eines besseren Lösung gelöscht.

Terme werden vom IR-Prozeß zu Listen angeordnet und intern so abgelegt, daß einerseits ihre sequentielle Abfolge erhalten bleibt und gleichzeitig aber ein schlüsselgestützter Schnellzugriff über eine Hash-Array-Funktion möglich ist.

### 3.4.3. Stoppworte

Die gebräuchlichste Form einer Negativliste ist eine sogenannte Stopwortliste (beim Information-Retrieval werden Negativ- und Positivlisten verwendet; eine Positivliste ist eine Schlagwortliste). Jedes Textwort wird mit der Stoppwortliste abgeglichen und, falls es dort gefunden wird, als solches gekennzeichnet und vom weiteren IR-Prozeß ausgeschlossen. Der Zweck der Verwendung einer Stopwortliste ist die Reduktion der Anzahl laufender Wortformen eines Textes auf etwa die Hälfte (bei einer Größenordnung der Stopwortliste von 100 bis 1000 Einträgen). Es wird also Speicherplatz und v.a. Dingen Verarbeitungszeit gespart. Die Stoppwortliste in unserer Dämonimplementation im Arbeitsspeicher gehalten, auf den um Zehnerpotenzen schneller als auf externe Speicher (für Wörterbücher z.B.) zugegriffen werden kann.
Dennoch ist in unserem System die Ablage der Stoppwortliste als Datenbanktabelle vorgesehen, die einmalig, nämlich beim Start des Dämons, in den Speicher gelesen werden kann.

```
...
Davon
Davor
Dazu
Dazubehalten
```

```
Dazulassen
Daß
...
wolle
wollen
wollend
wollest
wollet
wollt
wollte
wollten
wolltest
wolltet
...
```

**Tabelle 35: Stoppworte**

In Tabelle 35 ist ein Auszug aus dem Rohdatenfile der Stoppwortliste abgebildet. Sie enthält aktuell 53714 Stoppwörter. Die große Anzahl kommt dadurch zustande, daß wir unser IR-System bezüglich der Verarbeitung einer Optimierung unterzogen haben. Denn eine ausführliche Stoppworte erweist sich als sehr effizienzsteigernd. Oder anders gesagt: es ist in dem Kontext des hier eindeutig wirtschaftlich geprägten Umfeldes sinnvoll, z.B. theaterwissenschaftliche oder andere völlig themengebietsfremde Wörter durch Eintrag in die Stoppwortliste auszuschließen. Wir haben sie, ebenso wie die Schlagwortliste, durch einen wirtschaftswissenschaftlichen Projektmitarbeiter manuell prüfen und erstellen lassen.
Die Stoppwortliste enthält deshalb

- Artikel
- Pronomen
- Partikel (Adverbien, Präpositionen, Konjunktionen) und
- Interjektionen,
- Hilfsverben, Modalverben (“wollen”, “müssen”, “können”) ohne lexikalische Bedeutung und
- eine ausgewählte Anzahl von Vollverben

Die die in dieser Liste aufgeführten Stoppwörter wurden nicht nur in ihrer Grundform aufgehommen, sondern auch mit ihren verschiedenen Flektionen (Deklinationen bzw. Konjugationen)

### *3.5. Stemming*

Den wissenschaftlichen Hintergrund liefert die Morphologie, ein Teilgebiet der Linguistik, die sich mit der internen Struktur von Wörtern beschäftigt. So wie Sätze aus Wörtern zusammengesetzt sind, bestehen Wörter wiederum aus kleineren Bausteinen. Betrachtet man etwa die Wörter *windig* und *Windeseile*, so wird unmittelbar klar, daß beide Wörter den gemeinsamen Baustein *wind* haben: *wind* ist ein Morphem und, weil es eine lexikalische Bedeutung trägt, ist es ein lexikalisches Morphem. Dabei geht es nicht rein formal um Buchstabenfolgen: Die Übereinstimmung von *winzig* und *windig* ist zufällig, also die Vorsilbe *win* ist nicht etwa ein Morphem. Es gibt Morpheme, die der Wortbildung dienen (Derivate), wie etwa ‚*ig*‘ in *wind**ig*** und solche, die der Anpassung an den grammatischen Kontext des Wortes dienen (Flexive), wie etwa ‚*es*‘ in *die Kraft des Wind**es***.
Verfahren, die Flexive entfernen, erzeugen eine **Grundform** des Wortes (Wortart bleibt erhalten); die **Stammform** entsteht durch zusätzliches Entfernen der Derivate.
Im Projektzusammenhang werden über den Porter-Stemmer die Grundformen bestimmt und verwendet.

### 3.5.1. Schlagworte

Schlagworte sind die aus dem definierten Wirtschaftsdatenraum unserer Datenbank extrahierten Wörter mit Bedeutung.
Die Bedeutung definiert sich durch grundsätzliche Relevanz von Wörtern für den Suchprozeß einer Anfrage. Hierzu werden die Kategorien und darin wieder die einzelnen Kategorieeintragungen der einzelnen Bestände untersucht. Die Bedeutung von Wörtern wird per Hand und intellektuell von einem wirtschaftswissenschaftlichen Experten festgestellt. Auf diese Art wird ein Wort zum Schlagwort definiert. Gleichzeitig wird die Referenz dieses Schlagwortes bezogen auf den Kategorieeintrag festgehalten. Dieser Vorgang wiederholt sich für alle Kategorien aller Bestände.
In einem daran anschließenden Prozeß werden diese Informationen in die Datenbank übertragen. Schlagwörter werden fortlaufend und eindeutig numeriert. Die Schlagwörter werden maschinell mit Hilfe des Porterstemmers gestemmt.
Alle Schlagworte der intellektuellen Indexierung liegen als mit fortlaufend vergebenen Satznummern versehene Listen mit den Elementen Orginalwort und Stemmwort in der Datenbank.

| | | |
|---|---|---|
| ... | | |
| 752 | Ladenmöbel | Ladenmöbel |
| 709 | Lager | Lag |
| 543 | Lagerbehälter | Lagerbehält |
| 877 | Lagerhaltung | Lagerhaltung |
| ... | | |

**Tabelle 36: Schlagwortliste der Datenbank**

Die in diesem Prozeß festgelegten Referenzen werden werden in weiteren Listen in der Datenbank abgelegt: In einer n:m- Beziehung werden die Satznummern jedes Schlagwortes den entsprechenden Kategorieeinträgen (Kategoriename und Kategorieeintrag) zugeordnet.

| | | |
|---|---|---|
| ... | ... | ... |
| 752 | bam_eurostat_branch_de | 12 |
| 752 | bam_eurostat_branch_de | 13 |
| 709 | bam_brand_bip | 7 |
| ... | ... | ... |

**Tabelle 37: Schlagwortreferenz der Datenbank**

Diesen Vorgang nennen wir Indexierung.
Er findet nur dann einmal statt, wenn sich grundsätzliche Änderungen in der Datenbasis ergeben haben.

## *3.6. Voting*

Unter dieser Überschrift haben wir das verwendete Stimm- oder Voting-Verfahren gestellt, das die zentrale Entscheidungsmethode bzw. die Relevanz in der Datenbank vorhandener Dokument bezüglich der **Anfrage des Benutzers** definiert.

### 3.6.1. Übersicht

Das von uns verwendete Voting-Verfahren basiert auf den im Information Retrieval verwendeten Verfahren.
Nachdem alle Terme eines anfragenden Textes (z.B. ein kompletter Fließtext) durch Terming, Parsing und Filtern bearbeitet wurden und nach der Identifizierung der Schlüsselwörter wird die folgend beschriebene Voting- oder Stimmmethode benutzt.
Eine kurze Beschreibung des Verfahrens in Stichworten:

1. Vergleich aller Terme des Textes mit der Schlagwortreferenz der Datenbank
2. Akkumuliere alle Treffer für jedes Wertepaar *Kategorie & Eintrag-der-Kategorie*
3. Errechne die Treffer pro logischem Bestand für jedes Wertepaar *Kategorie & Eintrag-der-Kategorie*
4. Gewichte alle Treffer in einem zweistufigen Bewertungsverfahren
5. Lege eine Rangfolge der Treffer fest
6. Markiere die „gewinnenden“ Treffer und alle ihre dazugehörigen logischen Bestände

### 3.6.2. Votingverfahren

Ein Votingverfahren beschreibt den vollständigen Vorgang der Auswahl relevanter Kategorieeinträge in den (für eine Benutzergruppe zugewiesenen) Beständen unserer Datenbank.
Ausgehend von den in der Datenbank existierenden (logischen) Beständen wird die Häufigkeit eines Schlagwortes (Deskriptors) über alle (logischen) Bestände ermittelt. Denn in je mehr (logischen) Beständen ein Schlagwort enthalten ist, desto mehr Informationen unterschiedlicher Herkunft existieren zu diesem Begriff im Suchraum der Datenbank. Für den Deskriptor wird mit Hilfe des nachfolgend erklärten Gewichtungsverfahren eine Relevanz (Bedeutung) errechnet. Einem Nutzer soll also ein sinnvoll großer Pool von Beständen und damit mehrere Rechercherichtungen angeboten werden, in die die Informationen zu einem Schlagwort weisen.

#### 3.6.2.1. Gewichtungsverfahren

Gewichtungsverfahren werden potentiell an zwei verschiedenen Stellen verwendet.
Die erste betrifft die Gewichtung der Gesamtheit der vorkommenden Schlagwörter im anfragenden Text im Verhältnis zu denen im Suchraum der Datenbank vorkommenden. An dieser Stelle findet das hauptsächliche Gewichtungsverfahren statt.
Das zweite Gewichtungsverfahren wird an der Stelle angewendet, an der der Benutzer die Recherche fortführen will und Kontextinformationen zur der anzustrebenden Suchrichtung vorgibt. In diesem zweiten Schritt, der vorerst geplant, aber noch nicht realisiert ist, werden zwei Schlagwörter des anfragenden Textes zueinander in Beziehung gesetzt und bewertet, welche der gefundenen logischen Bestände respektive ihre zugehörigen Kategorien zur Fortführung der Recherche die höherwertigen Ergebnisse (die meisten Treffer) auf sich vereinigen. Bisher werden dem Benutzer alle weiteren, für die anderen Kategorien des Bestandes gefundenen Einträge angeboten.

Das Modell von Luhn [Luhn 1959] basiert auf der Ermittlung von Worthäufigkeiten aufbauend auf folgenden drei Schritten:

- Ermittlung der absoluten Häufigkeit jedes einzelnen Begriffes für jedes Dokument
- Ermittlung der Häufigkeit eines Begriffes in der gesamten Dokumentensammlung
- Eliminierung aller Begriffe, die öfters als der obere Schwellenwert und seltener als der untere Schwellenwert vorkommen

Bei diesem Modell wird davon ausgegangen, daß alle Begriffe, die in einem Datenbestand sehr oft vorkommen, nicht als Deskriptoren geeignet sind, da durch diese Begriffe die Dokumente nicht voneinander abgegrenzt werden können. Begriffe, die sehr selten vorkommen, werden als zu unbedeutend erklärt, um den Inhalt eines Dokuments repräsentieren zu können.

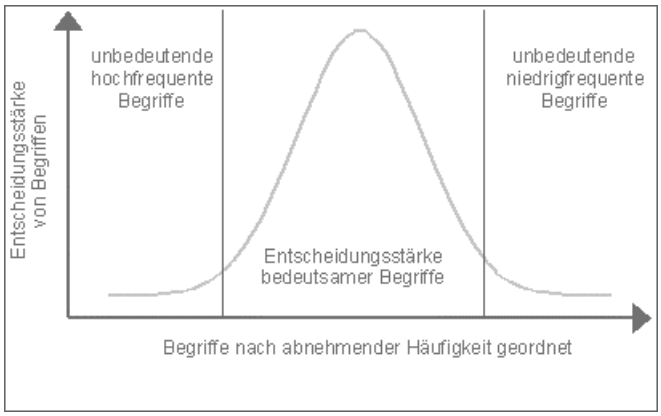



**Abbildung 30: Entscheidungsstärke bedeutsamer Begriffe**

Innerhalb der statistischen Modelle geht Sparck Jones' Modell [Sparck Jones 1998] davon aus, daß die Bedeutsamkeit eines Begriffes proportional zur Häufigkeit des Begriffes im Dokument, jedoch umgekehrt proportional zur Gesamtanzahl der Dokumente der gesamten Datenbasis, in denen der Begriff vorkommt, ist. Der Schwerpunkt der Gewichtung wird auf die Begriffe gelegt, die im Dokument selbst relativ oft vorkommen, die aber gleichzeitig in relativ wenigen Dokumenten der gesamten Datenbasis zu finden sind. Damit ist die Abgrenzung und Unterscheidung der einzelnen Dokumente voneinander gewährleistet.
An der Kombination dieser Modelle haben wir uns im Projekt orientiert.

### 3.6.2.2. Berechnungsformel

Die Bewertungskriterien werden in der Berechnung der inversen Dokumentenhäufigkeit haben wir gemäß den Modellen von Luhn/ Sparck realisiert.

Ausgehend von den auf der Datenbank existierenden (logischen) Beständen ist die Häufigkeit eines Schlagwortes (Deskriptors) über alle (logischen) Bestände zu ermitteln.
Z. B.:

Produktion bam_ifo_kthem 9,13,23,24,25,26,27
Produktion bam_eurostat_themes_de 1,2
Hier ist das Schlagwort „Produktion" in zwei von insgesamt 11 verfügbaren logischen Beständen enthalten; diese Bestände sind die Konjunkturthemen des ifo-Instituts München und die Themen der europäischen Statistik (Statistik für die Europäische Union).

Umsatz bam_brand_themen4,5,6
Umsatz bam_eurostat_themes_de 3,4,5,6
Umsatz bam_brand_bthemen 12,13,14
Das Schlagwort „Umsatz" ist in drei verfügbaren Beständen enthalten.

Bergbau bam_vgr_konten 17
Bergbau bam_eurostat_branch_de 3,11,12,13,37,38,39,44,45,46,47,48,57,59
Bergbau bam_brand_nace 1,32
Das Schlagwort „Bergbau" ist in drei verfügbaren Beständen enthalten.

Unternehmen bam_vgr_konten 14,15,16,17,18,19,20,21,22,23,24,25,26,47,

49,50,61,76,101,122,123,124,127,141
Unternehmen bam_vgr_sektor 17,23,38,42,57,64,75,79,90,91,92,93,94,95,
96,97,101,103,104,105,106,107,108,109,
110,111,112,113,114,115,116,117,118,119,
120,121,122,123,124,125,126,127,128,129,
144,150,154
Unternehmen bam_brand_bip 13,15,17,18
Unternehmen bam_brand_erwt 13
Das Schlagwort „Unternehmen“ ist in vier verfügbaren logischen Beständen enthalten.

Es ist also die Häufigkeit eines Schlagwortes s in den Beständen B zu bestimmen und daraus ergibt sich eine entsprechende Gewichtung für den Begriff:

Gewichtung $_{Bs}$ = Häufigkeit $_{Bs}$

Mit max Gewichtung $_{Bs}$ = 11 $_{Bs}$ und min Gewichtung $_{Bs}$ = 1 $_{Bs}$

Angewendet auf oben stehende Beispiele ergibt sich:
Zu (1) Gewichtung $_{Produktion}$ = 2 $_{Produktion}$ über alle Bestände.
Zu (2) Gewichtung $_{Umsatz}$ = 3 $_{Umsatz}$ über alle Bestände.
Zu (3) Gewichtung $_{Bergbau}$ = 3 $_{Bergbau}$ über alle Bestände.
Zu (4) Gewichtung $_{Unternehmen}$ = 4 $_{Unternehmen}$ über alle Bestände.

Am meisten Information könnte ein Nutzer der Datenbank nach dieser Bewertung also über den Begriff „Unternehmen“ erhalten, da dieser in diesem Zusammenhang die größte Häufigkeit aufweist (ist in vier Beständen enthalten).
Darüber hinaus muß die Häufigkeit eines Schlagwortes s in den Baumeinträgen b der betroffenen Bestände B ermittelt werden (anders gesagt: wie viele Satznummern sind einem Schlagwort über die Bestände zugeordnet, in denen es vorhanden ist).

Die Anzahl der Satznummern ergibt die Dokumentenhäufigkeit eines Schlagwortes, pro Bestand bzw. über alle Bestände, in denen es enthalten ist. Je mehr Satznummern der betreffenden Bestände im Zusammenhang mit einem bestimmten Deskriptor stehen, desto unspezifischer ist der Begriff bzw. desto mehr ist der Begriff als (relativ undifferenzierter) Oberbegriff anzusehen. Diese Deskriptoren sind nicht dafür geeignet, Dokumente voneinander abzugrenzen.

Daher wird diese Undifferenziertheit von Schlagwörtern als (negativ wirkendes) Bewertungskriterium ergänzt. Auf die selbe Weise werden Deskriptoren berücksichtigt, die so selten in untersuchten Dokumenten vorkommen, das sie zu spezifisch sind und damit auch nicht repräsentativ für das Dokument.
Die Bewertungskriterien werden in der Berechnung der inversen Dokumentenhäufigkeit berücksichtigt:

$$\frac{\log_2}{Dokumentenhäufigkeit_{Bbs}+1} = \log_2(n) - \log_2(Dokumentenhäufigkeit_{Bbs})$$

**Abbildung 31: Berechnungsformel der inversen Dokumentenhäufigkeit**

Mit n als Anzahl der „Dokumente“ (alle Baumeinträge der betroffenen Bestände), in denen das Schlagwort enthalten ist.

Wenn man die oben aufgeführten Beispiele benutzt, kommt man zu folgenden Ergebnissen:

Zu (1) Dem Schlagwort „Produktion“ sind im ifo-Bestand 7 Satznummern und im Eurostat-Bestand 2 Satznummern zugeordnet (insgesamt ergibt sich also eine Dokumentenhäufigkeit von 9).
Zu (2) Dem Schlagwort „Umsatz“ sind insgesamt 10 Satznummern über drei Bestände zugeordnet (Dokumentenhäufigkeit = 10).
Zu (3) Dem Schlagwort „Bergbau“ sind insgesamt 17 Satznummern über drei Bestände zugeordnet (Dokumentenhäufigkeit = 17).
Zu (4) Dem Schlagwort „Unternehmen“ sind insgesamt 76 Satznummern über vier Bestände zugeordnet (Dokumentenhäufigkeit = 76).

Eine Gewichtung über die inverse Dokumentenhäufigkeit ergibt sich wie folgt:

Zu (1) Gewichtung $_{\text{Produktion}} = 2_{\text{Produktion}} * [\log_2(96) – \log_2(9_{\text{Produktion}}) +1] = 4{,}06$
66 Baumeinträge bei ifo-kt (Themen) und 30 Baumeinträge bei Eurostat (Themen), also n = 96
Zu (2) Gewichtung $_{\text{Umsatz}} = 3_{\text{Umsatz}} * [\log_2(60) – \log_2(10_{\text{Umsatz}}) + 1] = 5{,}33$
Zu (3) Gewichtung $_{\text{Bergbau}} = 3_{\text{Bergbau}} * [\log_2(620) – \log_2(17_{\text{Bergbau}}) + 1] = 7{,}8$
Zu (4) Gewichtung $_{\text{Unternehmen}} = 4_{\text{Unternehmen}} * [\log_2(356) – \log_2(76_{\text{Unternehmen}}) + 1] = 6{,}68$

Ausgehend von oben angeführten Beispielen würde das Schlagwort „Bergbau“ mit der höchsten Gewichtung abschließen.

### 3.6.3. Algoritmische Methoden

Die Kernmethoden des IR-Subsystems basieren auf einer Sammlung von SQL-formulierten Anfragen. Diese SQL-Anfragen können aussschließlich innerhalb und mit prozeduraler Unterstützung der Datenbank ausgeführt werden und nutzen damit speziell die Vorteile schneller Vergleichsoperationen und generell das Konzept eines DBMS aus.
Eine der zentralen SQL-Anfragen haben wir beispielhaft in Abbildung 32 aufgezeigt.

```
declare global temporary table session.result
                          ( wkey           integer,
                            stemeintrag    varchar(100),
                            eintrag        varchar(100),
                            treffer_im_text integer,
                            kategorie      varchar(40),
                            satznummer     integer4,
                            Treffer_in_DB  integer       )
                       on commit preserve rows with norecovery, noduplicates";

Insert into session.result
                       (       wkey,
                               stemeintrag,
                               eintrag,
                               treffer_im_text,
                               kategorie,
                               satznummer,
                               Treffer_in_DB  )

                     Select distinct  mtw.wkey, mtw.stemeintrag, mtw.eintrag,
                               ses.treffer_im_text, trim(mlw.kategorie),
                               mlw.satznummer, count(*) as Treffer_in_DB
                     from met_tab_wort_".$sprache." mtw,
                               session.votetmp ses,
                               met_lnk_wort_de mlw
```

```
                    where mlw.wkey = mtw.wkey and
                          ses.eintrag = mtw.stemeintrag
group by       mtw.stemeintrag, mtw.eintrag,
               mlw.kategorie, mlw.satznummer,
               mtw.wkey, treffer_im_text
```

**Abbildung 32: Beispiel zentrale SQL-Anfrage**

Eine nährere Erläuterung zu den oben abgebildeten SQL-Anweisungen findet sich im Kapitel 3.10.5 Modul vote auf Seite 96.

## *3.7. Tagging (Markieren)*

## *3.8. Implementierung der Methoden im Dämon*

Die Intelligenz des Voting-Verfahrens ist einerseits an die SQL-Anfragen und andererseits an die Verwendung von Dämon-internen Datenstrukturen und entsprechenden Prozeduren geknüpft.
Unsere Implementierung der Methoden fand mit Hilfe von Perl-Modulen statt. Perl bot sich innerhalb unserer UNIX-Umgebung als verlässliches, flexibles und performantes Werkzeug an, zumal es Bibliotheken zur Verfügung stellt, die direkt in vielen DBMS die erforderlichen und sich in der Entwicklungszeit häufig verbesserten SQL-Abfragen zu formulieren erlaubt. Aufgrund der Vielzahl von Perlmodulen können problemlos auch andere Datenbanken als die bei uns benutzte INGRES-DB verwendet werden.
Für die zukünftige Weiterentwicklung des Dämons wird allerdings eine Java-Lösung angestrebt. Diese soll den bisherigen Dämoncharakter durch eine verteilte Methodenimplementierung ablösen; der Java-Client benutzt sogenannte RMI-Verfahren (Remote Method Invocation), die direkt auf dem Datenbankrechner bereitliegen und dort die Verbindung via JDBC o.ä. realilsieren.

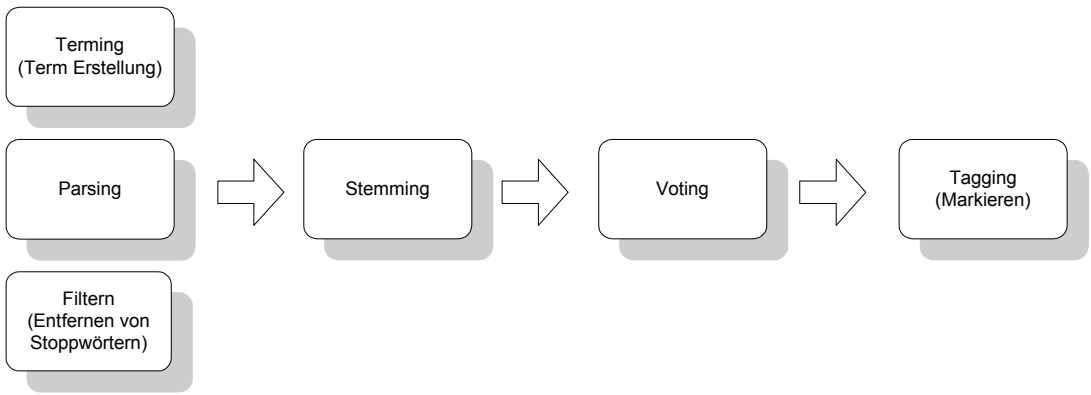



**Abbildung 33: Module des Dämons**

Die Quelltexte der Module wurden soweit wie möglich englischsprachig gehalten. So wurden Variablen und Kommentartexte weitgehend englisch benannt, um auch für die internationale Zusammenarbeit und Dokumentation (Kongresse) gerüstet zu sein.

### 3.8.1. Dämonprogramm irdaemon

Das Startprogramm des Dämon. Hier werden die betriebssystem- und kommunikationsspezifischen Einstellungen aufbewahrt, z.B. die Adress- und Portdaten des Standard-Dämons, die eigene Adresse mit Rechnernamen und Portnummer sowie die Zugangsparameter zur Datenbank.
Hier ist auch die spezielle Art des Dämons und dessen Verankerung als Hintergrundprozeß definiert. Der gestartete Dämon arbeitet als unabhängiger Prozeß

im Hintergrund und ist dann nur noch durch seine Prozess-ID erreichbar und nur durch ein spezielles Signal beendbar.
Das Dämonprogramm führt beim Start einige Initialisierungsroutinen durch. Insbesondere werden hier wichtige Daten einmalig aus der Datenbank ausgelesen und im Hauptspeicher abgelegt, da sie für jeden IR-Prozeß gebraucht werden. Das System verbessert hierdurch die Performance.
Fehlermeldungen und Verbindungsprotokolldaten werden über die betriebssystemspezifischen Logdämonen abgewickelt und schließlich in eine spezielle Logdatei geschrieben. Durch Auswerten dieser Datei lassen sich Zugriffsfrequenzen herausfinden und sowie Benutzergruppenanalysen und ähnliches generieren.

#### 3.8.2. Modul prefs

In diesem Modul werden systemweite Voreinstellungen gehalten.
Hier werden auch die internen Datenstrukturen definiert und erläutert. Das sind z.B. die Definitionen für die XML-Tags, Arrays mit Schlüsselzugriff für alle IR-relevanten Dateneinheiten (Domäne, Bestand, logischer Bestand etc.) Datenstrukturen sind hier - wenn sie darstellbar sind - mit Beispielen im Quellcode erläutert.

#### 3.8.3. Modul util

Dieses Modul enthält eine Sammlung von allgemeinen Routinen, die sich keiner speziellen Funktionalität zuordnen ließen.
Hauptsächlich sind dies Routinen zur Konvertierung von Ausgabedaten in verschiedene Formate.

#### 3.8.4. Modul stem

Enthält alle für das Stemming wichtigen Funktionen.
Die Funktionen für die Stoppwortinitialisierung und –ermittlung und für das sprachspezifische Stemming.

#### 3.8.5. Modul vote

Enthält alle für das Voting (Stimmverfahren) wichtigen Funktionen.
Die Funktion `prepare_number_of_snrs_each_category` wird nur beim Start der Dämonkomponente des Systems verwendet. Sie ermittelt die für unsere Gewichtungsformel notwendigen Summen von Einträgen pro benutzter Kategorie. Siehe auch Gewichtungsformel in Kapitel 3.6.2.2 auf Seite 74.

Die Funktion `prepare_MultiTermKeywords` ist ebenfalls eine einmalig beim Start des Dämons verwendete Funktion, die sämtliche Multiterm-Schlagwörter in interne Speicherstrukturen lädt (für deutsch, englisch und französisch). Als Multiterm-Schlagwörter werden im IR-Prozeß diejenigen Schlagwörter bezeichnet, die mehr als ein Wort lang sind. Beispiele sind „Bundesanstalt für Arbeit“ oder „Schleswig Holstein“. Diese langen Schlagwörter werden sowohl in ihrer ursprünglichen als auch in ihrer gestemmten Form (die während des Ladens dieser Daten ermittelt wurde), in eine interne Datenstruktur gelesen. Dieses Verfahren ist aufgrund der relativ geringen Anzahl von langen Multitermen möglich, die bei ca. 89 von 1554 (entspricht 5,72%) im deutschen oder bei 558 von 903 (entspricht 61,79%) liegt. Hier ist deutlich ein sprachspezifischer Unterschied festzustellen; der im Deutschen relativ geringe Anteil an Mehrworttermen ist auf die Möglichkeit der Neubildung von Wörtern durch Zusammeziehen zurückzuführen. Im Deutschen ist wird also aus „Bearbeitung von

Metall“ das Wort „Metallbearbeitung“, während im englischen eine Mehrwortkonstruktion „Treatment of Metal“ notwendig ist.

Die übrigen Funktionen dieses Moduls werden für jede Session (jeden Retrievallauf) erneut benutzt. Der Ablauf der der zentralen Voting-Funktion `query_db_ir_tables` erfolgt in folgenden Schritten:

1. Zuerst wird die vom Client gelieferte Benutzergruppe mit den Daten aus der Datenbank verglichen und - wenn sie existiert – die dieser Benutzergruppe zugeordneten Bestände gesetzt.
2. Nun werden alle im anfragenden Text identifizierten Terme, die die Stoppworte-Ausfilterung überstanden haben, in eine temporäre, sitzungstemporäre Tabelle mit Original und Stemmworteintrag geladen.
3. Eine Tabelle zur Aufbewahrung der Votingergebnisse wird vorbereitet.
4. Die für die spezifische Benutzergruppe gültigen logsichen Bestände und Kategorien werden selektiert und für den weiteren IR-Prozeß in eine temporäre Tabelle gespeichert.
5. Aus der benutzergruppen-reduzierten Anzahl der Kategorien wird nun ein Vergleich mit den Schlagwörtern des Anfragetextes (die in eine temporäre Tabelle der Datenbank geschrieben wurden) durchgeführt. Hierbei wird die Anzahl der Treffer pro Schlagwort in der temporären Ergebnistabelle vermerkt.
6. Die Ergebnisse in der temporären Tabelle werden nun schrittweise in die internen Datenstrukturen des Dämons übertragen.
   Gibt es an diesem Punkt des Voting-Verfahrens keine Ergebnismenge, wird der gesamte Prozeß abgebrochen. Die Query hat damit keine Treffer erzeugt und der Client muß darauf entsprechend reagieren.
7. Kann der IR-Prozeß an dieser Stelle weitergeführt werden, hat es also mindestens einen Treffer als Ergebnis gegeben, kann nun das Gewicht für die gevoteten Schlagwörter berechnet werden. Dies geschieht mit Hilfe der in 3.6.2.2 Berechnungsformel auf Seite 74 erläuterten Formel. Diese Gewichtungsergebnisse werden in eine Rangfolge umgesetzt und in den internen Datenstrukturen abgelegt.

Im Anschluß an diesen Voting-Prozeß stehen die Ergebnisse für diese IR-Sitzung in den internen Datenstrukturen für die Weiterverabeitung (das Tagging) bereit.

### 3.8.6. Modul tag

Enthält alle für das Tagging (Markieren) wichtigen Funktionen.
Hier sind Funktionen abgelegt, die einen anfragenden Text mit reinen HTML-Tags oder mit XML-Tags versehen.
Der erstere Funktionsbereich dient für Test- und Übersichtszwecke über die Recherche-Ergebnisse. Die vorgenommenen HTML-Tags haben einen rein deskriptiven Charakter und werden NICHT in der Produktionumgebung verwendet. Neben dem reinen Tagging werden hier auch Beginn und Ende-Definitionen von HTML-Seiten sowie statistischen Angaben erzeugt.
Der zweite Funktionsbereich dient dem XML-Tagging. Hier gibt es zwei Versionen, von denen eine mit einem internen Ausgabepuffer arbeitet und die andere eine sofortige Übertragung in den Ausgabekanal des Dämons gewährleistet. Diese zweite Variation war nötig geworden, da die Länge der Anfragetexte erheblich angestiegen war und den internen Puffer zum Überlauf brachte.

#### 3.8.7. Modul main

Dies ist das Hauptmodul des IR-Subsystems, wenn es NICHT als Dämonprozeß verwendet werden soll. Dies ist insbesondere für den Testbetrieb relevant. Die normale Methode des IR-Subsystemstarts ist die als Dämonprozeß. Siehe hierzu Kapitel 3.10.1 Dämonprogramm irdaemon.
In diesem Modul wird die freie Texteingabe mit Hilfe einer Datei an das System übergeben. Nach der Abarbeitung dieser Eingabe terminiert diese Modul.

## *3.9. Entwicklungsumgebung*

Während der Entwicklung des gesamten STATLINK-Systems wurden verschiedene Entwicklungstools benutzt.

### 3.9.1. Web-basierte Entwicklungstools

Um die IR-bezogenen Metadaten zu verifizieren, zu testen und zu verwalten, wurden HTML-Formulare und entsprechende CGI-Programme entwickelt.
So ist es möglich, die Inhalte von Metadatentabellen vollständig oder selektiv zu durchsuchen. Mit Hilfe dieser Instrumente können gleichzeitig Retrievalschritte im einzelnen ausgeführt und geprüft werden.
Die untenstehende Abbildung 34 ermöglicht in Form 1 und 2 die Abfrage und Anzeige der Inhalte der zentralen Metatabellen für die Stoppworte, die Schlagworte (pro Sprache), die Verweiseinträge der Schlagworte.

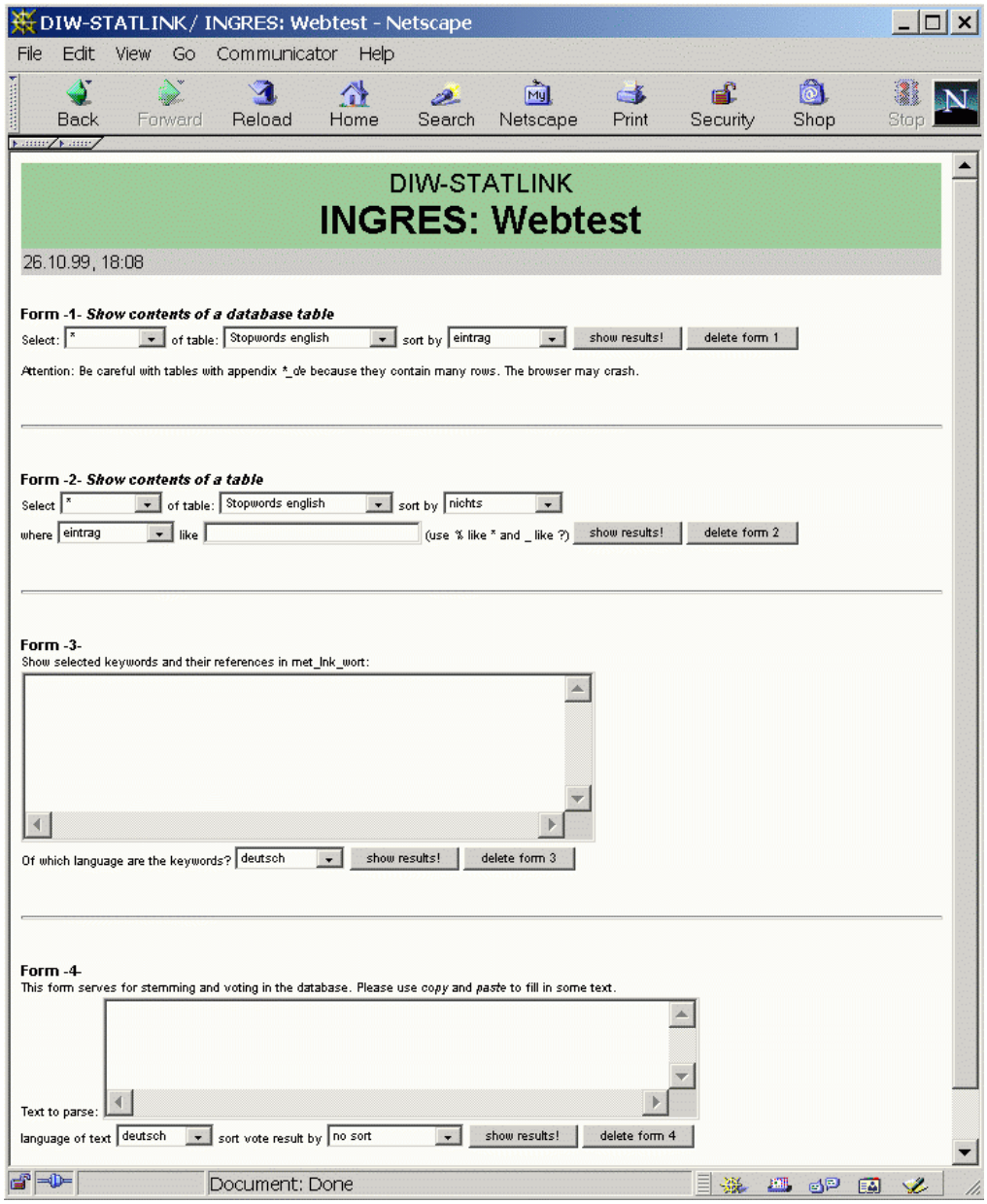



**Abbildung 34: HTML-Formular zur Prüfung der Metadatentabellen**

In Form 3 kann eine freie Abfrage auf Teilstrings durchgeführt werden. Die Ergebnisse sind in der beschreiben folgenden Abbildung 35 gezeigt.

Antwort

# DIW-STATLINK: Webtest Formnr 3

$ing_database: "schlupp::widab" $dbd: "Ingres"

SQL-Statement wird ausgeführt:
"select mtwo.eintrag, mtwo.stemeintrag, mlw.kategorie, mlw.satznummer from met_lnk_wort_de mlw, met_tab_wort_de mtwo, session.irtmp ses where mlw.wkey = mtwo.wkey and mtwo.stemeintrag = ses.stemeintrag group by mtwo.eintrag, mtwo.stemeintrag, mlw.kategorie, mlw.satznummer having count (*) > 0"

| EINTRAG | STEMEINTRAG | KATEGORIE | SATZNUMMER |
|---|---|---|---|
| Eisen | Eise | bam_eurostat_branch_de | 260 |
| Eisen | Eise | bam_eurostat_branch_de | 261 |
| Eisen | Eise | bam_eurostat_branch_de | 262 |
| Eisen | Eise | bam_eurostat_branch_de | 267 |
| Eisen | Eise | bam_eurostat_branch_de | 297 |
| Eisen | Eise | bam_eurostat_branch_de | 298 |
| Eisen | Eise | bam_eurostat_branch_de | 299 |
| Eisen | Eise | bam_eurostat_branch_de | 302 |
| Eisen | Eise | bam_stabu_branchen_de | 231 |
| Eisen | Eise | bam_stabu_branchen_de | 233 |
| Eisen | Eise | bam_stabu_branchen_de | 234 |
| Eisen | Eise | bam_stabu_branchen_de | 235 |
| Eisen | Eise | bam_stabu_branchen_de | 269 |

13 rows read from schlupp::widab.

Hier beendete sich noice.cgi.

**Abbildung 35: Ergebnis aus Form 3**

Hier wurde das Schlagwort “Eisen” eingegeben. Hier kann geprüft werden, in welchen Kategorien/ Satznummern dieses Schlagwort enthalten ist und wie die gestemmte Form dieses Wortes aussieht.
Eine Eingabe in Form 4 erzeugt die in Abbildung 36 sichtbare Ausgabe. Form 4 nimmt komplette Suchanfragetexte an und stellt ein das Ergebnis des Stemmings und eine erste Beurteilung der Treffer dar.

handelbare (handelbar) (5557) Zertifikate (Zertifikat) (5568) als überlegenes (überlegen) (5584) Instrument (Instrument) (5596) der Umweltpolitik (Umweltpolitik) (5611) dargestellt Diese unterscheiden (scheiden) (5644) sich jedoch weder (wed) (5670) in ihrer Wirkung (Wirkung) (5685) noch in den Problemen (Problem) (5705) ihrer Ausgestaltung (Ausgestaltung) (5721) wesentlich von Umweltabgaben (Umweltabgabe) (5750) Die höhere ökologische (ökologisch) (5776) Zielgenauigkeit (Zielgenauigkeit) (5788) durch die exakte (exakte) (5814) Festlegung (Festlegung) (5821) der zulässigen (zulässig) (5836) Emissionsmengen (Emissionsmenge) (5847) fällt bei langfristigen (langfristig) (5873) Reduktionszielen (Reduktionsziele) (5887) weniger ins Gewicht (Gewicht) (5916) Ein ökonomischer (ökonomisch) (5929) Effizienzvorteil (Effizienzvorteil) (5942) gegenüber Abgaben (Abgabe) (5969) besteht nicht Schließlich (Schließlich) (5995) werden in den letzten Jahren (Jahr) (6029) Selbstverpflichtungen (Selbstverpflichtung) (6036) der Wirtschaft (Wirtschaft) (6062) diskutiert Dieses Instrument (Instrument) (6092) ist allerdings besonders anfällig (anfällig) (6128) für Lobbying (Lobbying) (6141) und setzt kaum dynamische Anreizwirkungen (Anreizwirkung) (6176) im Hinblick (Hinblick) (6195) auf technologischen (technologisch) (6208) und strukturellen (strukturell) (6228) Wandel (Wandel) (6242) Es ist sehr fraglich (fraglich) (6262) ob durch Absprachen (Absprache) (6281) und Vereinbarungen (Vereinbarung) (6296) weitreichende (weitreichend) (6311) umweltpolitische (umweltpolitisch) (6325) Ziele (Ziel) (6342) realisierbar (realisierbar) (6348) sind Selbstverpflichtungen (Selbstverpflichtung) (6367) stellen daher keine Alternative (Alternativ) (6409) zur ökologischen (ökologisch) (6425) Steuerreform (Steuerreform) (6438) dar (dar) (6451) sondern höchstens (höchstens) (6464) eine Ergänzung (Ergänzung) (6479)

**Abbildung 36: (a) getaggter Text aus Form 4**

Wie hier gut zu erkennen, machen die in türkis dargestellen Stoppworte einen erheblichen Anteil an der gesamten Textmenge aus.
Die violett dargestellten Terme wurden als Schlagwort in der Datenbank gefunden; in Klammern steht dann die gestemmte Version des Schlagwortes sowie die Position des Schlagwortes im anfragenden Text. All dies ist in der folgenden Abbildung 37: (b) Legende aus Form 4 auch dargestellt.

| legend | |
|---|---|
| is in stopwortlist | This term was found in stopwordfile |
| was stemmed (original word) | This word was stemmed (stemmed word) |
| got voted | This word got voted |

**Abbildung 37: (b) Legende aus Form 4**

In Abbildung 38: (c) Statistische Auswertung aus Form 4 sind noch einige statistische Angabe über den gerade abgelaufenen Stemmingprozeß sichtbar, nämlich

- Größe des anfragenden Textes,
- seine Wortanzahl,
- die gesamte Anzahl der enthaltenen Stemmwörter
- mit ihrem mehrfachen Vorkommen,
- den Stemmwörtern, die kleiner als drei Zeichen sind,
- der tatsächlich im IR-Prozeß verwendeten Stemmwörter sowie
- die Gesamtanzahl von Stoppwörtern.

Diese Angaben sind jeweils in englischer Sprache ausgegeben, um den internationalen Kontakt zu erleichtern.

| statistics | |
|---|---|
| Total number of bytes | 6498 |
| Total number of words | 696 |
| | |
| Total number of stemwords | 513 |
| Of which ... are simular | 245 |
| Number of stemwords length < 3 | 3 |
| Effective number of stemwords | 268 |
| Total number of stopwords | 180 |

**Abbildung 38: (c) Statistische Auswertung aus Form 4**

Schließlich bietet Abbildung 39: (d) Ergebnisauswertung aus Form 4 noch einen ersten Einblick in die temporäre Ergebnistabelle. Diese ist als “Rohergebnis” aufzufassen, da hier noch nicht die Gewichtungsoperationen des Dämons sichtbar sind, die erst nachträglich innerhalb der internen Datenstrukturen ausgeführt werden können.

| Votelist | | | | | |
|---|---|---|---|---|---|
| WKEY | STEMEINTRAG | EINTRAG | TREFFER_IM_TEXT | KATEGORIE | TREFFER_IN_DB |
| 1342 | Abgabe | Abgaben | 2 | bam_vgr_konten | 1 |
| 59 | Unternehme | Unternehmen | 1 | bam_brand_ewt | 1 |
| 1173 | Wirtschaftsbereich | Wirtschaftsbereiche | 1 | bam_brand_ewt | 1 |
| 368 | Energie | Energie | 1 | bam_eurostat_branch_de | 1 |
| 878 | Steuerung | Steuerungen | 1 | bam_ifo_kbran | 1 |
| 59 | Unternehme | Unternehmen | 1 | bam_stabu_thema_de | 1 |
| 1365 | Haushalte | Haushalte | 1 | bam_brand_bip | 2 |
| 1328 | Verkehr | Verkehr | 1 | bam_brand_bip | 2 |
| 1365 | Haushalte | Haushalte | 1 | bam_brand_ewt | 2 |
| 1328 | Verkehr | Verkehr | 1 | bam_brand_ewt | 2 |
| 123 | Einnahme | Einnahme | 1 | bam_buba_tabelle | 2 |
| 1143 | Steu | Steuern | 1 | bam_stabu_thema_de | 2 |
| 1328 | Verkehr | Verkehr | 1 | bam_vgr_konten | 2 |

**Abbildung 39: (d) Ergebnisauswertung aus Form 4**

Immerhin ist aber auch hier schon ein Trend des Ergebnisses des Voting-Verfahrens abzulesen: neben den Schlagwörtern sind hier auch schon die Kategorien und auch die Trefferanzahl in der Datenbank und das Vorkommen des Schlagwortes im anfragenden Text dokumentiert.

### 3.9.2. Kommandozeilen-Tools

Das klassische Telnet-Programm ist eine weitere Möglichkeit, Anfragen an das IR-Subsystem zu senden.
Zu diesem Zweck nimmt einfach eine Verbindung über Adresse und Port zu dem Server auf, auf dem der erweiterte IR-Dämon arbeitet.
Während der STATLINK-Entwicklung wurde häufig von diesem Werkzeug Gebrauch gemacht. Zum Zwecke des detaillierten Monitorings des IR-Prozesses kann der erweitererte IR-Dämon in einen sogenannten Debuggingmodus geschaltet werden. In diesem Modus liefert er ausführliche Information über die einzelnen Zwischenergebnisse des IR-Prozesses. Es ist möglich, bis auf die Ebene einzelner SQL-Abfragen zurückzugehen und deren Zwischenergebnisse zu prüfen. Ebenfalls ist es möglich, die intern gehaltenen Informationen des Dämons strukturiert auszugeben.

## *3.10. Perl-Sourcen des erweiterten Dämons (IR-Dämon)*

Hier finden Sie die Sourcetexte des erweiterten Dämons, des IR-Dämons. Sie sind als Perl- Quellen hier veröffentlicht.

### 3.10.1. Dämonprogramm irdaemon

```
#!/usr/local/bin/perl
#package irdaemon;
use strict;
BEGIN { $ENV{PATH} = '/usr/ucb:/bin' }
use POSIX;
use Socket;
use FileHandle;
use Carp;
use Sys::Syslog;
require "./ir_connect_daemon.pl";

# Become independent from the starting terminal
POSIX::setsid() or die ("Can't start a new session (setsid): $!");

# to make the irdaemon.pl really start as a daemon, make
# it fork and kill the parent.
$main::startpid= fork;
exit if $main::startpid;
die "Couldn't fork $! at startup to run as daemon. " unless defined
($main::startpid);

# Begin of effective daemon code
# declaration of constants

$irdaemon::own_daemon_hostname    = (uname)[1];     # local host
$irdaemon::own_daemon_port        = 54322;
# We try to configure ourself: if daemon is running at DIW, we use
# lukas as widab-daemon.
# If we are not at DIW, we assume the widab-daemon running on the same
# machine like in all development systems for Statfinder/IRAIA
if ($irdaemon::own_daemon_hostname =~ /diwsysv/) {
  $irdaemon::widab_daemon_hostname  = "lukas.diw-berlin.de"; } # host of
widab-daemon at DIW
else {$irdaemon::widab_daemon_hostname  = (uname)[1];} # local host
$irdaemon::widab_daemon_port      = 54321;

# Start logging
# Logging assumes a running syslogd -r (SuSE-Linux) or equivalent
# having definined the log facility local2
openlog("irdaemon.pl", "pid", "local2");
```

```
# Begin of procedure and functions

sub spawn;  # forward declaration
sub logmsg {
  syslog("info", "--> ". join '', @_ );
}

my $port = shift || $irdaemon::own_daemon_port;
my $proto = getprotobyname('tcp');
$irdaemon::port = $1 if $port =~ /(\d+)/; # untaint port number

socket(Server, PF_INET, SOCK_STREAM, $proto)        || die "socket: $!";
setsockopt(Server, SOL_SOCKET, SO_REUSEADDR,
                               pack("l", 1))        || die "setsockopt:
$!";
bind(Server, sockaddr_in($irdaemon::port, INADDR_ANY))  || die "bind: $!";
listen(Server,SOMAXCONN)                            || die "listen: $!";

logmsg "*** server initially started on port $irdaemon::port
($irdaemon::own_daemon_hostname)\n    using widab-daemon
$irdaemon::widab_daemon_hostname : $irdaemon::widab_daemon_port";

# Prepare multiterm keywords ONCE, only when daemon initially starts
&reset_global_variables;
logmsg "  * Reset global variables";
&prepare_number_of_snrs_each_category;
&prepare_MultiTermKeywords;
logmsg "  * MultiTermKeywords prepared";
logmsg "  * irdaemon ready ...";

my $waitedpid = 0;
my $paddr;

sub REAPER {
$waitedpid = wait;
$SIG{CHLD} = \&REAPER;  # loathe sysV
logmsg "Query finished, terminated: $waitedpid" . ($? ? " with exit $?" :
'');
}

$SIG{CHLD} = \&REAPER;

for ( $waitedpid = 0;
 ($paddr = accept(Client,Server)) || $waitedpid;
 $waitedpid = 0, close Client)
{
next if $waitedpid and not $paddr;
my($port,$iaddr) = sockaddr_in($paddr);
my $name = gethostbyaddr($iaddr,AF_INET);

logmsg "connection from $name [",
       inet_ntoa($iaddr), "]
       at port $port";

spawn sub {
   #print "Hello there, $name, it's now ", scalar localtime, $EOL;
   #print "Perl Wrapper Widab Daemon ready...\n";

   logmsg("Begin of communication by client $name.\n");
   { # begin block
     my ($go, $antwort, $frage, $localhost, $hostname, $sockaddr, $name,
$pport, $aliases,
       $pproto, $type, $len, $thisaddr, $thataddr, $this, $that,
       $tmpantwort, $tmpfrage);
     #$frage   = "variable variable\nGH1100\nEOT\n";
     $sockaddr = 'S n a4 x8';  # Magic Number
```

```
#chop($localhost = `hostname`);
$localhost = $irdaemon::own_daemon_hostname;
$hostname  = $irdaemon::widab_daemon_hostname;
$pport     = $irdaemon::widab_daemon_port;

($name, $aliases, $pproto)                = getprotobyname('tcp');
($name, $aliases, $type, $len, $thisaddr) = gethostbyname($localhost);
($name, $aliases, $type, $len, $thataddr) = gethostbyname($hostname);

# Socket holen, binden und verbinden
$this = pack($sockaddr, AF_INET, 0, $thisaddr);
$that = pack($sockaddr, AF_INET, $pport, $thataddr);
socket(S, AF_INET, SOCK_STREAM, $pproto) || die "socket: $!";
bind(S, $this)                           || die "bind: $!";
connect(S, $that)                        || die "connect: $!";

S->autoflush(1);
$irdaemon::communication = 1;

# create a new child process
if (defined ($irdaemon::kidpid = fork())) {
  logmsg "New listen client proc, spawning: $irdaemon::kidpid";
}
else {die "can't fork $!"}
# PROCESSING client --> me --> widabdaemon

# Enable independent communication from client to
# widab daemon: write to widab daemon with the help of
# a new child with pid $kidpid
if ($irdaemon::kidpid)
{
  my ($printtoggle, $queryfreetexttoggle);
  $main::queryfreetexttoggle= 0;
  while (<STDIN>)
  {
    $main::printtoggle= 1;
  $tmpfrage = $_;
  #print(STDERR "Debug: $tmpfrage");
  if ($main::queryfreetexttoggle)
  { if ($tmpfrage =~ /^EO/i)
    { $main::queryfreetexttoggle= 0;
      logmsg("Input free query text");
      # run textquery!
      if ($main::DO_IR) {
        &main::ir_process_text($main::queryfreetext);
        print "EOLABER\n";
      }
      print "EOT\n";
      $main::queryfreetext= "";
        $main::printtoggle= 1;
    }
    else { $main::queryfreetext.= $tmpfrage; $main::printtoggle= 0;}
  }
  elsif ($tmpfrage =~ /^EO/i)
  { logmsg("read from client: $tmpfrage"); }
  elsif ($tmpfrage =~ /^variable/i)
  { logmsg("read from client: $tmpfrage"); }
  elsif ($tmpfrage =~ /^baum/i)
  { logmsg("read from client: $tmpfrage"); }
  elsif ($tmpfrage =~ /^query free text/i)
  { logmsg("read from client: $tmpfrage");
    $main::queryfreetexttoggle= 1; $main::printtoggle= 0;}

  elsif ($tmpfrage =~ /^switch ir=on/i)
  { logmsg("read from client: $tmpfrage");
```

```
        $main::printtoggle= 0;
        # do the switch
        $main::DO_IR=1;
        }
      elsif ($tmpfrage =~ /^switch ir=off/i)
      { logmsg("read from client: $tmpfrage");
        $main::printtoggle= 0;
        # do the switch
        $main::DO_IR=0;
        }
      elsif ($tmpfrage =~ /^switch irdebug=on/i)
      { logmsg("read from client: $tmpfrage");
        $main::printtoggle= 0;
        # do the switch
        $main::DO_IR_DEBUG=1;
        }
      elsif ($tmpfrage =~ /^switch irdebug=off/i)
      { logmsg("read from client: $tmpfrage");
        $main::printtoggle= 0;
        # do the switch
        $main::DO_IR_DEBUG=0;
        }
      else {}

      # DocType "text" berücksichtigen

      print (S $tmpfrage) if $main::printtoggle;

      #if (!$tmpfrage =~ /^NULL/) { print (S $tmpfrage);}

      last if ($tmpfrage =~ /^EOS/i)
      }
      #kill ("TERM" => $kidpid); # send SIGTERM to child
    }
    else
    {


      while ($irdaemon::communication)
      {
        # PROCESSING client <-- me <-- widabdaemon
      # Processing widab daemon's output
      logmsg("Begin receiving output from widab daemon");
      while (<S>)
      {
        $tmpantwort = $_;
        print $tmpantwort;
        #print(STDERR "$tmpantwort");
        if ($tmpantwort =~ /^EOT/i) { logmsg("read from daemon:
$tmpantwort"); }
        if ($tmpantwort =~ /^variable/i) { logmsg("read from daemon:
$tmpantwort"); }
        if ($tmpantwort =~ /^baum/i) { logmsg("read from daemon:
$tmpantwort"); }
        last if ($tmpantwort =~ /^EOS/i);
      } #endwhile (<S>)
        $irdaemon::communication=0;
      } #endwhile ($communication)
      #print("Goodbye from WIDAB Server\n");
      #print("EOS\n");
      close(S);
      logmsg("End receiving output from widab daemon");
      #kill ("TERM" => $kidpid); # send SIGTERM to child

    } # end else $kidpid
    #kill ("TERM" => $kidpid); # send SIGTERM to child
```

```
      select STDOUT;
      print(STDERR "");


      #print $antwort;
    } # end block
};

}

sub spawn {
my $coderef = shift;

unless (@_ == 0 && $coderef && ref($coderef) eq 'CODE') {
   confess "usage: spawn CODEREF";
}
my $pid;
if (!defined($pid = fork)) {
   logmsg "cannot fork: $!";
   return;
} elsif ($pid) {
   logmsg "New query, spawning: $pid";
   return; # I'm the parent
}
# else I'm the child -- go spawn

open(STDIN,  "<&Client")   || die "can't dup client to stdin";
open(STDOUT, ">&Client")   || die "can't dup client to stdout";
## open(STDERR, ">&STDOUT") || die "can't dup stdout to stderr";
exit &$coderef();
}
```

### 3.10.2. Modul prefs

```
#!/usr/local/bin/perl

# Modul preferences for IR-Abilities of STATFINDER daemon
#

# The following variables exist and are not reset!
#
# %DBMultiTermKeywords_de
# %DBMultiTermKeywords_en
# %DBMultiTermKeywords_fr
# %countsnrcat

sub reset_global_variables {
  # definitions for Sys::syslog,
  # these lines are just descriptive:
  # "local2";       # Logging facility definition
  # "info";         # Log priority definition
  # Just commentary, already defined in Sys::Syslog;
  # loglevel used for irdaemon
  # LOG_INFO

  $DEBUG=0;   # Switch on intense debugging by setting '1'

  ##########################################################
  # database values ??????????????????
  $dbname = 'schlupp::widab';
  $dbd    = 'Ingres';
```

```
  ############################################################
  # markers/ switches
  $STOPWORD = "_stopword_";
  $NOSTOPWORD = "_nostopword_";
  $NOSTEMWORD = "_nostemword_";
  $STEMWORD = "_stemword_";
  $VOTEWORD = "_voteword_";
  # switch
  # Do not reset the two following values
  #$DO_IR = 0;          # Do the IR-processes
  #$DO_IR_DEBUG = 0;    # Do the debugging SQL-Output

  ############################################################
  # tags
  $tag_html_replace = "_replace_";
  $tag_html{$STOPWORD} = "<SPAN
CLASS=A10stop>".$tag_html_replace."</SPAN>";
  $tag_html{$NOSTOPWORD} = "<SPAN CLASS=A10>".$tag_html_replace."</SPAN>";
  $tag_html{$VOTEWORD} = "<SPAN
CLASS=A10vote>".$tag_html_replace."</SPAN>";
  $tag_html{$STEMWORD} = "<SPAN
CLASS=A10stem>".$tag_html_replace."</SPAN>";

  # ir xml tags
  $QUERYTEXTTAG = "<querytext>";
  $USERGROUPTAG = "<usergroup>";
  $LANGUAGETAG = "<language>";

  ############################################################
  # constants

  ############################################################
  # files
  # stopwordlist
  $STOPWORDFILE_DE = "stopwords_de.dat";
  $STOPWORDFILE_EN = "stopwords_en.dat";
  $STOPWORDFILE_FR = "stopwords_fr.dat";

  ############################################################
  # global variables
  $original_text ="";         # keeps the original text
  $tagged_text ="";           # keeps the tagged text


  ############################################################
  # central textitem arrays
  ############################################################
  # all @titem* arrays hold the information about the textitems as
  # they appeared in the original text; thus they have a sequential order
  # textitem is a string recognized more or less as word (delimited by
space
  # or synatactical signs like . , ? ! etc. See grammar of
ir_parse_text*.pl
  # @titem can contain strings/words more often than one time. @titem is
just
  # a mirror of the original text.
  undef @titem;               # textitems of the parsed text
  undef @titempos;            # contains position of the textitem in the
original text
  #
  # Example of a filled titem - array:
  #
  ############################################################

  ############################################################
  # Hasharray of processed titems
```

```
  ##########################################################
  # Hasharray of all already PROCESSED textitems
  # All strings/words of @titem get processed to decide wether they are
  # stopwords or words to be processed further.
  # If they are stopwords they get marked as stopword (4).
  # If they are no stopswords they get stemmed and - if they occured
already -
  #   they get counted (7).
  undef %ptitem;              # processed textitems
  # ptitem{titem}             (1) ptitem contains its titem as the key
  # ptitem{titem}{"wkey"}           (2) contains the wkey or undefined
  # ptitem{titem}{"stemmed"}  (3) contains the stemmed titem or $UNDEFINED
  # ptitem{titem}{"stopped"}  (4) contains $STOPWORD if item is a stopword
  # ptitem{titem}{"voted"}    (5) contains the number of ocurences of titem
in DB
  # ptitem{titem}{"catlist"}  (6) contains a list of category/snr pairs of
all hits of titem in DB
  # ptitem{titem}{"count"}    (7) contains the number of ocurences of titem
in text
  # ptitem{titem}{"code"}           (8) contains bit-coded context
information of titem
  #                     Bit 1:  Item is at the beginning of the sentence
  #                                   and therfore written in capital
letters
  $CAPITAL = 1;               # Bit 1 - Constant
  #                     Bit 2:  ....
  #$XYZ = 1;                  # Bit 2 - ....
  # ptitem{"titem"}{"count voted domains"}   (9) contains the number of
domains
  #                                              touched by this titem
  # ptitem{"titem"}{"count hits total"}     (10) contains the total number
of hits
  #                                              for this titem
  # ptitem{"titem"}{"grand total items all log domains"} (11) grand total
of items of each
  #                                              touched by this item
  # ptitem{"titem"}{"weight"}               (12) contains the calculated
IR-weight
  #                                   for this item
  # ptitem{"titem"}{"order"}                (13) contains the calculated
place of relevance
  #                                   for this item, which can be 1, 2 or 3
  #
  # Example of a filled ptitem - array:
  # given: string of item "query text":
  # " This is a text of the items Bergbau and potatoes. "

  # ptitem{"This"}{"stemmed"} = $NOSTEMWORD
  # ptitem{"This"}{"wkey"}    = UNDEF or 0
  # ptitem{"This"}{"stopped"} = $STOPWORD
  # ptitem{"This"}{"voted"}   = UNDEF or 0
  # ptitem{"This"}{"catlist"} = UNDEF
  # ptitem{"This"}{"count"}   = 1
  # ptitem{"This"}{"code"}    = BIN00000001
  # ptitem{"This"}{"count voted domains"}   = UNDEF or count
  # ptitem{"This"}{"count hits total"} = UNDEF or count
  # ptitem{"This"}{"grand total items all log domains"} = UNDEF or count
  # ptitem{"This"}{"weight"}  = UNDEF or weight (float)
  # ptitem{"This"}{"order"} = undef or order (integer)
  # ...
  # ptitem{"Bergbau"}{"stemmed"} = "fruit"
  # ptitem{"Bergbau"}{"wkey"}    = 310
  # ptitem{"Bergbau"}{"stopped"} = $NOSTOPWORD
  # ptitem{"Bergbau"}{"voted"}   = 1
  # ptitem{"Bergbau"}{"catlist"} = [ ["bam_vgr_konten",1] ]
  # ptitem{"Bergbau"}{"count"}   = 1
```

```
# ptitem{"Bergbau"}{"code"}    = bin00000001
# ptitem{"Bergbau"}{"count voted domains"}   = 3
# ptitem{"Bergbau"}{"count hits total"} = UNDEF or count
# ptitem{"Bergbau"}{"grand total items all log domains"} = UNDEF or count

# ptitem{"Bergbau"}{"weight"} = 7.8
# ptitem{"Bergbau"}{"order"} = 1
############################################################

############################################################
# Hasharray of db data
############################################################
# Hasharray domains touched during actual IR-process
undef %dbdomain;
# $dbdomain{"ifo-kt"}=1;
# $dbdomain{"ifo-it"}=2;
# Array domains touched during actual IR-process, indexed by id
undef @dbdomainid;
# $dbdomainid[1]="ifo-kt";
# $dbdomainid[2]="ifo-it";


# Hasharray logical domains touched during actual IR-process
undef %dblogdomain;
# $dblogdomain{"DIW-vgr-Konten"}=1;
# $dblogdomain{"DIW-vgr-Sektor"}=2;
# Array domains touched during actual IR-process, indexed by id
undef @dblogdomainid;
# $dbdomainid[1]="DIW-vgr-Konten";
# $dbdomainid[2]="DIW-vgr-Sektor";


# Hasharray categories touched during actual IR-process
# This hasharray contains some more specific information
undef %dbcat;
# $dbcat{"bam_vgr_konten"}=1;
# $dbcat{"bam_vgr_konten"}{"logicaldomain"}="DIW-vgr-Konten";
# $dbcat{"bam_vgr_konten"}{"total number of items"}=158;
# $dbcat{"bam_vgr_region"}="2";
# $dbcat{"bam_vgr_region"}{"logicaldomain"}="DIW-vgr-Konten";
# $dbcat{"bam_vgr_region"}{"total number of items"}=3;
# Array categories touched during actual IR-process, indexed by id
undef @dbcatid;
# $dbcatid[1]="bam_vgr_konten";
# $dbcatid[2]="bam_vgr_region";

# This hasharray keeps the number of rows for each category
# Don't make a global reset!
# undef %countsnrcat;
# This array will be filled once when daemon initially starts.
# Example
# $countsnrcat{"bam_vgr_konten"}=158
# $countsnrcat{"bam_vgr_sektor"}=154

# remember weight for item in the %totalweightlist
undef %totalweightlist;
# The key for this list is somewhat rare: it consists of the
# weight concatenated with the stemmed ptitem. This serves for
# sorting this list to find the third heaviest weights.
# A second field {titem} contains the item for which this weight
# is valid
# Example:
# $totalweightlist{"03.5563452346_*_Unternehme"}=03.5563452346;
# $totalweightlist{"03.5563452346_*_Unternehme"}{"titem"}="Unternehmen";

# accumulate total weight for all items (sum of all weights)
```

```
  undef $totalweight;
  # count number of all weighted items
  undef $totalweightitems;

} # end reset_global_variables

1;
```

### 3.10.3. Modul util

```
#!/usr/local/bin/perl

# Modul utils for IR-Abilities of STATFINDER daemon
# Collection of utilities for different kinds of tasks

sub log10 {
  my $n = shift;
  return log($n)/log(10);
}

sub Extract_tag_content {
  # This function extracts any content between $starttag and $endtag
  # in search space $xmltext
  # The function just uses one parameter (the starttag) and generates
  # the endtag or closing tag by adding "/" into it.
  # E.g. <querytext> -> </querytext>

  my ($starttag, $endtag, $xmltext);

  $xmltext = $_[0];   # sourcetext to search
  $starttag = $_[1];  # tag to extract content from
  $endtag= $starttag;
  # Add '/' behind < to contruct the closing tag
  $endtag =~ s/\</\<\//i;

  # regular expression: find starttag, anything, endtag
  $xmltext =~ /($starttag)(.*)($endtag)/si;
  return $2;
}

sub begin_html_output {
  my ($headline);
  $headline = $_[0];
  print <<Ausgabe_Beginn001;

<HTML>
<HEAD>
   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
   <TITLE>DIW-STATLINK: Webtest IR</TITLE>
<LINK REL="StyleSheet" HREF="../ir.css" MEDIA="screen">
</HEAD>
<BODY BGCOLOR="#FAFFF9">
<P>
<SPAN CLASS=A18>DIW-STATLINK: Webtest IR formnr $headline</SPAN>
</P>
<SPAN CLASS=A10>
Ausgabe_Beginn001
} # end sub begin_html_output


sub print_original_html_text {
  my ($otext);
  $otext = $_[0];


  print <<original_html_text;
```

```
  <P>
 <table width="54%" border="0" cellspacing="1" cellpadding="1"
bgcolor="#FFFFCC">
  <tr bgcolor="#C0C0D0">
    <td class="A10" width ="100%" bgcolor="#C0C0C0"><b>Original
text </b></td>
  </tr>
  <tr bgcolor="#C0C0D0">
    <td class="A10" width ="100%" bgcolor="#C0C0C0">$otext</td>
  </tr>
  </table>
original_html_text
}


sub print_html_stemwords
{
  my ($otext, $ttitem1, $ttitem2, $ttitem3, $count, $countg, $j);
  print <<html_stemword;
  <P>
 <table width="54%" border="0" cellspacing="1" cellpadding="1"
bgcolor="#FFFFCC">
  <tr bgcolor="#C0C0D0">
    <td class="A10" width ="70%" bgcolor="#C0C0C0"><b>Stemwordlist</b></td>
    <td class="A10" width ="30%" bgcolor="#C0C0C0"> </td>
  </tr>
  <tr bgcolor="#C0C0D0">
    <td class="A10" width ="70%" bgcolor="#C0C0C0"><b>Stemword</b></td>
    <td class="A10" width ="30%" bgcolor="#C0C0C0">Count</td>
  </tr>
html_stemword

foreach $ttitem1 (sort keys %main::ptitem)
{
  $ttitem2 = $main::ptitem{$ttitem1}{"stemmed"};
  print "(print_html_stemwords) \$ttitem2: $ttitem<BR>" if $DEBUG;
  if ($ttitem2 ne $NOSTEMWORD)
  {
    $count = $main::ptitem{$ttitem1}{"count"};
    $countg = "";
    for ( $j=1; $j <= $count; $j++) {$countg.= "*";}

  print <<html_stemword1;
  <tr bgcolor="#C0C0D0">
    <td class="A10" width ="70%" bgcolor="#C0C0C0"><b>$ttitem1
($ttitem2)</b></td>
    <td class="A10" width ="30%" bgcolor="#C0C0C0">$count $countg</td>
  </tr>
html_stemword1
  } # end if
}

print "</table>";
}

sub show_statistics {
  my ($booboo, $csword, $cswordt, $cswords, $cswordd, $chword,
$totalstemwords, $totalbytes, $totalwords);
  foreach $booboo (%main::ptitem) {
    $cswords++ if ((exists $main::ptitem{$booboo}{"stemmed"}) &&
($main::ptitem{$booboo}{"stemmed"} ne $NOSTEMWORD) && ( length
($main::ptitem{$booboo}{"stemmed"}) < 2));
    $cswordt++ if ((exists $main::ptitem{$booboo}{"stemmed"}) &&
($main::ptitem{$booboo}{"stemmed"} ne $NOSTEMWORD));
    $cswordd+=$main::ptitem{$booboo}{"count"}-1 if ((exists
$main::ptitem{$booboo}{"count"}) && ($main::ptitem{$booboo}{"count"} > 1));
```

```
    $chword++ if ((exists $main::ptitem{$booboo}{"stopped"}) &&
($main::ptitem{$booboo}{"stopped"} eq $STOPWORD));
  }
  $csword = $cswordt - $cswords;
  $totalwords=$#main::titem + 1;
  $totalbytes=length($main::original_text);
  $totalstemwords=$csword+$cswordd;

  print <<printstatistics;
 <p>
 <table width="54%" border="0" cellspacing="0" cellpadding="0"
bgcolor="#FFFFCC">
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><b>statistics</b></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"> </td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Total
number of bytes</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span class=A10><b>
$totalbytes</b></span></td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Total
number of words</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span class=A10><b>
$totalwords</b></span></td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"> </td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"> </td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Total
number of stemwords</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span class=A10><b>
$totalstemwords</b></span></td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Of which
... are simular</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span
class=A10><b>$cswordd</b></span></td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Number of
stemwords length < 3</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span class=A10><b>
$cswords</b></span></td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Effective
number of stemwords</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span class=A10><b>
$csword</b></span></td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><span class=A10>Total
number
      of stopwords</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"><span class=A10><b>
$chword</b></span></td>
  </tr>
</table>
printstatistics
}
```

```
1;
```

### 3.10.4. Modul stem

```
#!/usr/local/bin/perl -s

# Datei ir_stem.pl

# Modul stem for IR-Abilities of STATFINDER daemon
use DBI;
use Text::German;
use Text::English;

sub prepare_stopwortlist
{
  my ($language, $stopwordfile, $zeile, $counter,
      $dbh);
  # parameter for the stopwordfile
  $language = $_[0];
  $stopwordfile = $_[1];
  if (open(DATAIN,  "< $stopwordfile"))
  {
    print "<SPAN CLASS=A8>Getting stopwordlist from file...</SPAN>\n" if
$DEBUG;
    while (<DATAIN>)
    {
      chop;
      $zeile = $_; # in zeile wird die akt. Zeile von STDIN original
aufbewahrt
      $counter++;
      $stopword{$zeile} = 1;
    } # end while (<DATAIN>)
    print "<SPAN CLASS=A8>$counter stopwords read from file.</SPAN>\n" if
$DEBUG;
  } # endif open file
  elsif ($dbh = DBI->connect($dbname, "", "", $dbd))
  {
    print "Getting stopwordlist from $dbname...\n";
    $cursor = $dbh->prepare("Select eintrag from met_tab_stopp_".$sprache);
    $cursor->execute;
    die "Prepare Fehler: $DBI::err .... $DBI::errstr" if $DBI::err;

    while ( (@frow = $cursor->fetchrow)  )
    { $frow[0] =~ s/\s//g;
      $stopword{$frow[0]} = 1;
      #print "\"$frow[0]\"\n";
      $counter++;
    }

    print "$counter stopwords read from $dbname.\n";
    #print "$#stopword stopwords read from $dbname.\n";
    $cursor->finish;
    $dbh->commit;
    $dbh->disconnect;
  }
  else {die "Error (ir_stem): Unable to open source for stopword:
[$stopwordsource].\n"; }
} #end prepare_stopwortlist


sub stop_word
{
  my (@args, $stopw);
  $stopw = $_[0];
  # Is the stopword at the beginning of sentence? If yes lc stopw
  if ($main::ptitem{$stopw}{"code"} & $CAPITAL ) {$stopw = lc($stopw);}
```

```
  # lowercase for german disabled
  # if (exists ($stopword{lc($stopw)})) { return $main::STOPWORD; }
  if (exists ($stopword{$stopw})) { return $main::STOPWORD; }
  else { return $main::NOSTOPWORD}
} #end stop_word


sub stem_word
{
  my ($spr, @args, $stemword, $stemmedword, $pid,
      @stemworde, $ele, @tarray, @stemmed);
  $spr = $_[0];
  $stemword = $_[1];
  # $stemword already exists?
  if (exists $main::ptitem{$stemword}{"stemmed"})
    { print "*" if $DEBUG;
      # $main::ptitem{$stemword}{"count"}++;
     return $main::ptitem{$stemword}{"stemmed"}}
  #if ((exists $main::ptitem{$stemword}{"stopped"}) &&
($main::ptitem{$stemword}{"stopped"} eq $main::STOPWORD))
  #  {print "+" if $DEBUG; return $main::NOSTEMWORD}

  # Perl-Stemming for different languages

  SWITCH1: {
    if ($spr eq "de")
    {
      @stemworde = split (/[\s |\-\=\+\*|;.,:!?()\/\'\]\[\`\´\"\n]/,
$stemword);
      foreach $ele (@stemworde) {
        push (@tarray, Text::German::reduce ($ele));
      }
      $stemmedword = join (" ", @tarray);
      undef @tarray;
      last SWITCH1;
    }
    if ($spr eq "en")
    {
      @stemworde = split (/[\s |\-\=\+\*|;.,:!?()\/\'\]\[\`\´\"\n]/,
$stemword);
      @stemmed = Text::English::stem (@stemworde);
      $stemmedword = join (" ", @stemmed);
      last SWITCH1;
    }
    # keine Stemmingalgorytmem für französisch bekannt
    if ($spr eq "fr") { $stemmed = $stemword; last SWITCH1; }
  } # end switch1

  #if ($stemmedword=~ /Haushalte/) { print "\"$stemword\" Gestemmt zu
\"$stemmedword\"\n";}
  return $stemmedword;

} #end stem_word

# default "code" for every perl module
1;
```

### 3.10.5. Modul vote

```
#!/usr/local/bin/perl

# Modul vote for IR-Abilities of STATFINDER daemon
# Database defaults
# Database procedures

# Grundeinstellungen zur Datenbank
```

```
use DBI;
# $dbname $dbd already set in ir_prefs.pl

#####################################################
# Prepare %countsnrcat with the number of items of each category or:
# Find out how big is the total number of SNRs (item)
# for every category.
# Do this once (when daemon starts) and keep it in an array.
#####################################################
sub prepare_number_of_snrs_each_category {
  my ($sql_statement, $ttitem1, $ttitem2, %attr, $i, $j,$first);

  $dbh = DBI->connect($dbname, "", "", $dbd);

  undef %countsnrcat;
  # This sql retrieves all relevant categories (bam_* tables)
  # It is the preparation for the following count command for all
  # of these tables
  $sql_statement= "Select trim (lmlb.tabellenname),
                       lmben.logbestandsname
                        from
                       lnk_met_log_bestand lmlb,
                       lnk_met_benutzergruppe lmben
                        where lmben.benutzergruppe = \'all\' and
                        lmlb.logbestandsname = lmben.logbestandsname and
                        lmlb.tabellenname like \'bam_\%\'
                        ";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;
  $i=0;
  while ( (@row = $cursor->fetchrow)  )
  {
    $i++;
    $countsnrcat{lc($row[0])} = 0;
  }
  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  logmsg "     No of bam_% tables read: $i\n";
  $cursor->finish;
  $dbh->commit;

  $i=0;
  foreach $tempcat (keys %countsnrcat) {

    $sql_statement= "select count(*) from  $tempcat";
    $cursor = $dbh->prepare($sql_statement);
    #logmsg $sql_statement."\n";
    #logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
    @namea=@{$cursor->{NAME}};
    #logmsg join (" ", @{$cursor->{NAME}},"\n");
    $cursor->execute;
    @row = $cursor->fetchrow;
      $i++;
      $countsnrcat{$tempcat}= $row[0];
      if ($countsnrcat{$tempcat} <= 0) {
        logmsg "entries for $tempcat: $countsnrcat{$tempcat}\n";}
      #logmsg "$tempcat: $countsnrcat{$tempcat} rows\n";
    if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err ....
$DBI::errstr"); die; }
    $cursor->finish;
    $dbh->commit;
```

```
  } # end foreach

  logmsg "  * Category count finished for: $i categories";
  $dbh->disconnect;

 } # end sub prepare_number_of_snrs_each_category

#######################################################
# Prepare the internal array to keep multiple term keywords
# (MultiTermKeywords) ONCE, exactly when daemon initially
# starts
# So we have all MULTITERM keywords with us. The number of
# MultiTermKeywords has a maximal number of 550 terms 8/2000.
# It is not likely that this number will grow much in future.
#######################################################
sub prepare_MultiTermKeywords {
  my ($sql_statement, $ttitem1, $ttitem2, %attr, $i, $j,$first);

  $dbh = DBI->connect($dbname, "", "", $dbd);

  # Fill up $DBMultiTermKeywords_de (german)
  undef %DBMultiTermKeywords_de;
  $sql_statement= "Select
                    wkey, stemeintrag, eintrag
                     from met_tab_wort_de
                where stemeintrag like \'\% \%\'";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;
  $i=0;
  while ( (@row = $cursor->fetchrow)  )
  {
    $i++;
    $DBMultiTermKeywords_de{$row[1]} = $row[2];
  }
  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  $cursor->finish;
  $dbh->commit;
  logmsg "MultiTermKeywords de (german): $i";

  # Fill up $DBMultiTermKeywords_en (english)
  undef %DBMultiTermKeywords_en;
  $sql_statement= "Select
                    wkey, stemeintrag, eintrag
                     from met_tab_wort_en
                where stemeintrag like \'\% \%\'";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;
  $i=0;
  while ( (@row = $cursor->fetchrow)  )
  {
    $i++;
    $DBMultiTermKeywords_en{$row[1]} = $row[2];
  }
  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  $cursor->finish;
  $dbh->commit;
```

```
  logmsg "MultiTermKeywords en (english): $i";

  # Fill up $DBMultiTermKeywords_fr (french)
  undef %DBMultiTermKeywords_fr;
  $sql_statement= "Select
                   wkey, stemeintrag, eintrag
                    from met_tab_wort_fr
                where stemeintrag like \'\% \%\'";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;
  $i=0;
  while ( (@row = $cursor->fetchrow)  )
  {
    $i++;
    $DBMultiTermKeywords_fr{$row[1]} = $row[2];
  }
  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  $cursor->finish;
  $dbh->commit;
  logmsg "MultiTermKeywords fr (french): $i";

  $dbh->disconnect;
}

#########################################################
# Reverse voting function for MultiTermKeywords.
# We allready have all MultiTermKeywords which are contained
# in the DB (see &prepare_MultiTermKeywords).
# Now we take these stemmed MTKw's and search for their
# occurence in the querying text. If we find similar
# MTKw's we know they are in the DB tables and put them
# in the list of SingleTermKeywords ("normal keywords") and
# send them together into the normal voting process.
#########################################################
sub prevote_MultiTermKeywords {
  my ($i, $prevoted_total, $MTKw, $MTKwS, $joined_ptitems, $No_term_items);
  print "List of pre-voted MulitTermKeywords\n" if $DO_IR_DEBUG;
  $prevoted_total= 0;
  # for all MultiTermKeywords
  foreach $MTKw (keys %DBMultiTermKeywords) {
    #$MTKwS= $DBMultiTermKeywords{$MTKw};
    # The following senseless replace operation just counts the number
    # of blanks " " in terms which is the number of terms - 1
    $No_term_items= ($MTKw=~ s/ / /g) + 1;
    print "MTKw item: $MTKw ($No_term_items items)\n" if $DO_IR_DEBUG;
    # for alle items in querying text
    for ( $i=1; $i <= $#main::titem; $i++)
    {
      # pre-test if first term fits, if not process next item in text
      $stulle= $main::ptitem{$main::titem[$i]}{"stemmed"};
      if ($MTKw=~ /^$stulle/)
      {
        undef $joined_ptitems;
        for ( $j=1; $j <= $No_term_items; $j++)
        { $joined_ptitems.= "
".$main::ptitem{$main::titem[$i+$j]}{"stemmed"} } # endfor
      print "Joined item: $joined_ptitems\n" if $DO_IR_DEBUG;
      if ($joined_ptitems eq $MTKw)
        { $prevoted_total++;
          print "$MTKw ab Stelle $titempos[$i], $No_term_items lang\n" if
$DO_IR_DEBUG;
```

```
        } # endif
      } # endif
    } # endfor
  } # end foreach
  print "Number of prevoted MultiTermItems: $prevoted_total\n\n" if
$DO_IR_DEBUG;

} # end sub prevote_MultiTermKeywords

#######################################################
# Do voting functionality of db ir tables for html-requests
#######################################################
sub get_dbdata {
  my ($sql_statement, $ttitem1, $ttitem2, %attr, $i, $j,$first);
  $i   = 0;

  # prepare temporary table
  $dbh = DBI->connect($dbname, "", "", $dbd);
  print "Fehler beim Verbinden mit widab; $DBI::errstr\n" if not $dbh;
  $sql_statement= "declare global temporary table session.votetmp
                       (eintrag varchar(100), treffer_im_text integer)
                   on commit preserve rows with norecovery, noduplicates";
  print $sql_statement . "<BR>" if $DEBUG;
  $dbh->do ($sql_statement);
  print "Fehler beim Erstellen von session.votetmp; $DBI::errstr\n" if
$dbh->errstr;

  # reads items and their sums into temporary table
  while ( ($ttitem1, $v) = each %main::ptitem ) {
    $ttitem2 = $main::ptitem{$ttitem1}{"stemmed"};
    if (($ttitem2 ne $NOSTEMWORD) && ($ttitem2 ne "") &&
($main::ptitem{$ttitem1}{"count"} > 0))
    {
       $sql_statement= "insert into session.votetmp (eintrag,
treffer_im_text) values (\'".$ttitem2."\', "
.$main::ptitem{$ttitem1}{"count"}.")";
       print $sql_statement."<BR>" if $DEBUG;
       $dbh->do ($sql_statement);
       print "Fehler beim Insert in session.votetmp; $DBI::errstr\n" if
$dbh->errstr;
    }
  };

  $dbh->commit;

  # prepare temporary result table
  print "Fehler beim Verbinden mit widab; $DBI::errstr\n" if not $dbh;
  $sql_statement= "declare global temporary table session.result
                                              (wkey            integer,
                                    stemeintrag     varchar(100),
                                    eintrag         varchar(100),
                                    treffer_im_text integer,
                                    kategorie       varchar(40),
                                    satznummer      integer4,
                                    Treffer_in_DB   integer)
                   on commit preserve rows with norecovery, noduplicates";
  print $sql_statement . "<BR>" if $DEBUG;
  $dbh->do ($sql_statement);
  print "Fehler beim Erstellen von session.result; $DBI::errstr\n" if $dbh-
>errstr;
  $dbh->commit;

  # This sql query should'nt be changed no more
  $sql_statement= "Insert into session.result (wkey,stemeintrag,eintrag,
                                    treffer_im_text,
```

```
                                        kategorie,satznummer,Treffer_in_DB)


                      Select distinct mtw.wkey, mtw.stemeintrag,
mtw.eintrag,
                                        ses.treffer_im_text,
trim(mlw.kategorie),
                             mlw.satznummer, count(*) as Treffer_in_DB
                      from met_tab_wort_".$sprache." mtw,
                       session.votetmp ses,
                     met_lnk_wort_de mlw
                      where mlw.wkey = mtw.wkey and
                        ses.eintrag = mtw.stemeintrag
                    group by mtw.stemeintrag, mtw.eintrag, mlw.kategorie,
mlw.satznummer, mtw.wkey, treffer_im_text";
  print "GRFZ: ".$sql_statement."<BR>"; # if $DEBUG;

  $cursor = $dbh->prepare($sql_statement);
  print "Fehler bei prepare GRFZ: $DBI::errstr\n" if $dbh->errstr;
  $dbh->do ($sql_statement);


  $sql_statement= "Select
                       wkey, stemeintrag, eintrag, kategorie, satznummer,
treffer_im_text, Treffer_in_DB
                       from session.result
                  where treffer_im_text > 2
                    group by stemeintrag, eintrag, kategorie, satznummer,
wkey, treffer_im_text, Treffer_in_DB";
  if ($dborder ne "nosort") {
    $sql_statement.= " order by ".$dborder." desc";}
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."<BR>"; # if $DEBUG;
  print "Fehler bei prepare: $DBI::errstr\n" if $dbh->errstr;
  print "(get_dbdata) \$cursor->{NUM_OF_FIELDS}: ".$cursor-
>{NUM_OF_FIELDS}."<BR>" if $DEBUG;
  @namea=@{$cursor->{NAME}};
  print "(get_dbdata) \$cursor->{NAME}: " if $DEBUG;
  print join (" ", @{$cursor->{NAME}},"<BR>\n") if $DEBUG;
  $cursor->execute;

  die "Prepare Fehler: $DBI::err .... $DBI::errstr" if $DBI::err;
  print <<get_dbdata1;
  <P>
 <table width="54%" border="0" cellspacing="1" cellpadding="1"
bgcolor="#FFFFCC">
  <tr bgcolor="#C0C0D0">
    <td class="A10" width ="10%" bgcolor="#C0C0C0"><b>Votelist</b></td>
    <td class="A10" width ="45%" bgcolor="#C0C0C0"> </td>
    <td class="A10" width ="45%" bgcolor="#C0C0C0"> </td>
    <td class="A10" width ="45%" bgcolor="#C0C0C0"> </td>
    <td class="A10" width ="45%" bgcolor="#C0C0C0"> </td>
    <td class="A10" width ="45%" bgcolor="#C0C0C0"> </td>
  </tr>
get_dbdata1

  print"<tr bgcolor=\"#C0C0C0\"><td class=\"A10\" width=\"10%\"
bgcolor=\"#C0C0C0\">";
  print join ("</td> <td class=\"A10\" width=\"45%\" bgcolor=\"#C0C0C0\">
", @{$cursor->{NAME}}) ,"</td></tr>\n";

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {
```

```
          print"<tr bgcolor=\"#FFFF66\"><td class=\"A10\" width=\"10%\"
bgcolor=\"#FFFFCC\">";
        print join ("</td> <td class=\"A10\" width=\"45%\"
bgcolor=\"#FFFFCC\">", @row), "</td></tr>\n";
          # Where does the stemmed item exist?
        $tmp_vote_count = 0;
        if ($first == 1) {$first = 0} # Skip headline of rows
        else {
          for ( $j=0; $j < $count_titem; $j++)
          {
            if ($main::ptitem{$titem[$j]}{"stemmed"} eq $row[2]) {
            if ($tmp_vote_count eq 0) {$tmp_vote_count= $row[6];
                  $main::ptitem{$titem[$j]}{"voted"}+=$tmp_vote_count;
                  push ( @{$main::ptitem{$titem[$j]}{"catlist"}}, [
$row[3], $row[4] ] );
                #print "\"".$row[2] ."\",\"".
$row[5]."\",\"".$main::ptitem{$titem[$j]}{"voted"}."\" <BR>";
            } # end if
            } # end if
          } # end for
        } # end else
        $i++;
  }
  print "</table>\n";

  $cursor->finish;
  $dbh->commit;
  $dbh->disconnect;
  print "$i rows displayed.\n";
} # end sub


##############################################################################
##
# Voting functionality based on ir tables of daemon

##############################################################################
##
sub query_db_ir_tables {
  my ($sql_statement, $ttitem1, $ttitem2, $v, %attr, $i, $j,$first,
%tmpitem,
      $tmp_where);
  $i   = 0;

  # Connect to DB

  $dbh = DBI->connect($dbname, "", "", $dbd);
  if (not $dbh) {
    logmsg ("Fehler beim Verbinden mit widab; $DBI::errstr\n");
    return -1;
  }

  # first test if the desired usergroup exists in DB
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                     name from met_benutzergruppe
                 where name= \'".$usergroup."\'
                     order by name";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;
```

```
  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
return -1; }
  if ($DO_IR_DEBUG) {
    print "Usergroups in DB:\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {       $i++;
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;

  # Usergroup doesn't exist, stop ir Process
  if ($i==0) {
    $dbh->disconnect;
    return 0;
  }


  # Show contents of already fetched $DBMultiTermKeywords!
  if ($DO_IR_DEBUG) {
    print "DBMulitTermKeywords german:\n";
    foreach $i (sort keys %DBMultiTermKeywords) {
      print "$i : $DBMultiTermKeywords{$i}\n";
    }
    print "\n";
  }

  &prevote_MultiTermKeywords;

  # prepare temporary table of stemwords in querying text
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######

  $sql_statement= "declare global temporary table session.votetmp
                       (eintrag varchar(100), treffer_im_text integer)
                    on commit preserve rows with norecovery, noduplicates";
  print ($sql_statement . "\n") if $DO_IR_DEBUG;
  $dbh->do ($sql_statement);
  logmsg ("Fehler beim Erstellen von session.votetmp; $DBI::errstr\n") if
$dbh->errstr;

  # reads items and their sums into array first
  # Because we process all data item-wise, we have redundant
  # stemmed items: e.g. "Haushalte" and "Haushalten" become stemmed the
  # same: "Haushalte".
  # With the array tmpitem we will be sure storing the stemmed item only
once.
  print "Reading items in temporary array:\n" if $DEBUG;
  while ( ($ttitem1, $v) = each %main::ptitem ) {
    # $ttitem2 is the STEMMED word of $ttitem1:
    $ttitem2 = $main::ptitem{$ttitem1}{"stemmed"};
    # is $ttitem2 really a stemmed item?
    if (($main::ptitem{$ttitem1}{"stopped"} ne $STOPWORD) &&
         ########### $ttitem2 ne $NOSTEMWORD) &&
        ($ttitem2 ne "") &&
      ($main::ptitem{$ttitem1}{"count"} > 0))
    {
      # does $ttitem2 already exist in %tmpitem?
```

```
      if (exists $tmpitem{$ttitem2})
      { # Don't store $ttitem2, just add the hits in text
        $tmpitem{$ttitem2}+= $main::ptitem{$ttitem1}{"count"};
      }
      else {
        $tmpitem{$ttitem2}= $main::ptitem{$ttitem1}{"count"};
      }
    }
  };
  print "Reads items of %tmpitem in temporary table:\n" if $DEBUG;
  while ( ($ttitem1, $v) = each %tmpitem ) {
    $sql_statement= "insert into session.votetmp (eintrag, treffer_im_text)
values (\'".$ttitem1."\', " .$v.")";
    print $sql_statement."\n" if $DEBUG;
    print ($sql_statement."\n") if $DEBUG;
    $dbh->do ($sql_statement);
    logmsg ("Fehler beim Insert in session.votetmp; $DBI::errstr\n") if
$dbh->errstr;
  };

  $dbh->commit;

  # prepare temporary result table
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #######
  $sql_statement= "declare global temporary table session.kat_usergroup
                                          (kategorie       varchar(40),
                                  logbestand      char(40),
                                  bestand           char(20) )

                  on commit preserve rows with norecovery, noduplicates";
  print $sql_statement . "\n" if $DEBUG;
  $dbh->do ($sql_statement);
  logmsg ("Fehler beim Erstellen von session.result: $DBI::errstr") if
$dbh->errstr;
  $dbh->commit;

  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #######
  # This sql query fills session.kat_usergroup
  # It reduces the ir process for that categories which are linked to
  # the specified usergroup
  $sql_statement= "Insert into session.kat_usergroup
                        (kategorie, logbestand)

                    Select lmlb.tabellenname,
                       lmben.logbestandsname
                      from
                     lnk_met_log_bestand lmlb,
                     lnk_met_benutzergruppe lmben
                      where lmben.benutzergruppe = \'".$usergroup."\' and
                      lmlb.logbestandsname = lmben.logbestandsname and
                      lmlb.tabellenname like \'bam_\%\'
                      ";
  $cursor = $dbh->prepare($sql_statement);
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  $dbh->do ($sql_statement);
  print ($sql_statement."\n") if $DO_IR_DEBUG;
  $dbh->commit;

  # simple select of all entries in session.kat_usergroup
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #######
  $sql_statement= "Select
                    *
                      from session.kat_usergroup
                    order by kategorie";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
```

```
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Content of session.kat_usergroup: \n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {       $i++;
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;


  # prepare temporary result table
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "declare global temporary table session.result
                                              (wkey            integer,
                                    stemeintrag     varchar(100),
                                    eintrag         varchar(100),
                                    treffer_im_text integer,
                                    kategorie       varchar(40),
                                    satznummer      integer4,
                                    Treffer_in_DB   integer,

                                    bestand           char(20),
                                    logbestand      char(40)    )

                   on commit preserve rows with norecovery, noduplicates";
  print $sql_statement . "\n" if $DEBUG;
  $dbh->do ($sql_statement);
  logmsg ("Fehler beim Erstellen von session.result: $DBI::errstr") if
$dbh->errstr;
  $dbh->commit;

  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  # This sql query fills session.result
  $sql_statement= "Insert into session.result
                          (wkey,stemeintrag,eintrag,
                     treffer_im_text,
                     kategorie,satznummer,
                     Treffer_in_DB, logbestand)

                     Select distinct
                      mtw.wkey, mtw.stemeintrag, mtw.eintrag,
                          ses.treffer_im_text,
                    trim(sesku.kategorie), mlw.satznummer,
                    count(*) as Treffer_in_DB, sesku.bestand
                       from
                      met_tab_wort_".$sprache." mtw,
                      session.votetmp ses,
                    met_lnk_wort_de mlw,
                    session.kat_usergroup sesku
                       where mlw.wkey = mtw.wkey and
```

```
                        ses.eintrag = mtw.stemeintrag and
                      sesku.kategorie = trim(mlw.kategorie)
                    group by
                       mtw.stemeintrag, mtw.eintrag, sesku.kategorie,
                       mlw.satznummer, mtw.wkey, treffer_im_text,
                     sesku.bestand";
  print ($sql_statement."\n") if $DO_IR_DEBUG;
  $dbh->commit;

  $cursor = $dbh->prepare($sql_statement);
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  $dbh->do ($sql_statement);


  # complex select over all rows of session.result
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                     wkey, stemeintrag, eintrag,
                kategorie, satznummer, logbestand,
                treffer_im_text, Treffer_in_DB
                     from session.result
                 where treffer_im_text > 0
                   group by stemeintrag, eintrag, kategorie, satznummer,
logbestand, wkey, treffer_im_text, Treffer_in_DB
                     order by treffer_im_text desc";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Votelist\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
          # Where does the stemmed item exist?
        $tmp_vote_count = 0;
        #if ($first == 1) {$first = 0} # Skip headline of rows
        #else {
          for ( $j=0; $j < $count_titem; $j++)
          {
            if ($main::ptitem{$titem[$j]}{"stemmed"} eq $row[1]) {
            if ($tmp_vote_count eq 0) {$tmp_vote_count= $row[7];
                  $main::ptitem{$titem[$j]}{"voted"}+=$tmp_vote_count;
              # Store WkeyNo also
                  $main::ptitem{$titem[$j]}{"wkey"}=$row[0];
                  push ( @{$main::ptitem{$titem[$j]}{"catlist"}}, [
$row[3], $row[4] ] );
            } # end if
            } # end if
          } # end for
        #} # end else
        $i++;
  }
  print "\n" if $DO_IR_DEBUG;
```

```
  $cursor->finish;
  $dbh->commit;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;

  ##########################################
  # If this central query didn't lead to any result, other queries are
  # useless and we can stop the whole retrieval process:
  if ($i==0) {
    $dbh->disconnect;
    return 0;
  }

  # simple select of all entries in session.votetmp
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                     *
                       from session.votetmp
                     order by eintrag";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Content of session.votetmp: all keywords found in querying
text\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {       $i++;
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;

  # select of all categories in special Order
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                     kategorie, satznummer, eintrag
                       from session.result
                     group by kategorie, satznummer, eintrag
                     order by kategorie, satznummer desc";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print "(get_dbdata) \$cursor->{NAME}: " if $DO_IR_DEBUG;  print join ("
", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Vote categories grouped by categorie/Satznummer\n";
```

```
    print "--------\n";
    print "Sql-Statement:\n$sql_statement\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {     $i++;
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;


  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  # select of all categories grouped by category, hits each category AND
keyword
  $sql_statement= "Select
                     trim(kategorie) as tkategorie, count(*) as treffer,
eintrag
                       from session.result
                   group by kategorie, eintrag
                     order by tkategorie";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Number of hits for each keyword in each category:\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {       $i++;
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;


  # simple select of all categories
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                     trim(kategorie) as tkategorie, count(*) as treffer,
logbestand
                       from session.result
                   group by kategorie, logbestand
                     order by tkategorie, treffer desc";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
```

```
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Simple Vote categories, hits all together\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
        # storing categories redundant:
        # #1 name of cat => key of hasharray (e.g. "bam_vgr_konten")
        $dbcat{$row[0]}=++$i;
        # Additionally store the number of rows for this category which
        # was calculated at the initial start of daemon and stored in
        # $countsnrcat
        $dbcat{$row[0]}{"total number of items"}=$countsnrcat{$row[0]};
          print"countsnrcat for ".$row[0].": ".$countsnrcat{$row[0]}."\n"
if $DO_IR_DEBUG;
        # #2 if of cat => key of hasharray (e.g. 1)
        $dbcatid[$i]=$row[0];
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;


  # Select all keywords and the depending logical domains
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                    eintrag, trim(logbestandsname) as log_bestand,
trim(kategorie) as tkategorie,count(*) as treffer
                     from session.result, lnk_met_log_bestand
                where tabellenname = kategorie
                   group by eintrag, logbestandsname, kategorie";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Voted words and belonging logical domains/ categories\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";
  }

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
```

```
        $i++;
        # counting number of logical domains touched by this keyword
          $ptitem{$row[0]}{"count voted domains"}++;
        #print "No. of domains in total for ".$row[0].":
".$ptitem{$row[0]}{"count voted domains"}. "\n" if $DO_IR_DEBUG;;
          $ptitem{$row[0]}{"count hits total"}+=$row[3];
        print "No. of hits in total for ".$row[0].":
".$ptitem{$row[0]}{"count hits total"}. "\n" if $DO_IR_DEBUG;;
          $dbcat{$row[2]}{"logicaldomain"}=$row[1];
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;


  # simple select of all domains
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                    trim(logbestand) as tbestand, count(*) as treffer
                     from session.result
                where treffer_im_text > 0
                   group by logbestand
                     order by tbestand desc";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Simple Vote domains\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
        # storing categories redundant:
        # #1 name of dom => key of hasharray (e.g. "bam_vgr_konten")
        $dbdomain{$row[0]}=++$i;
        $dbdomainid[$i]=$row[0];
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;


  # simple select of all log domains
  ########  #  #  #  #  S Q L  S T A T E M E N T  #  #  #  #  #  #  #######
  $sql_statement= "Select
                    trim(logbestandsname) as log_bestand, count(*) as
treffer
                     from session.result, lnk_met_log_bestand
                where tabellenname = kategorie
                   group by logbestandsname";
  $cursor = $dbh->prepare($sql_statement);
  print $sql_statement."\n" if $DO_IR_DEBUG;
```

```
  logmsg ("Fehler bei prepare: $DBI::errstr\n") if $dbh->errstr;
  @namea=@{$cursor->{NAME}};
  print join (" ", @{$cursor->{NAME}},"\n") if $DO_IR_DEBUG;
  $cursor->execute;

  if ($DBI::err) { logmsg ("Prepare Fehler: $DBI::err .... $DBI::errstr");
die; }
  if ($DO_IR_DEBUG) {
    print "Simple Vote logical domains\n";
    print "--------\n";
    print "\t";
    print join ("\t", @{$cursor->{NAME}}) ,"\n";}

  $i=0;
  $first = 1;
  while ( (@row = $cursor->fetchrow)  )
  {
          print"\t" if $DO_IR_DEBUG;
        print join ("\t", @row), "\n" if $DO_IR_DEBUG;
        # storing logical domains redundant:
        # #1 name of logical domain => key of hasharray (e.g. "DIW-
Bauvolumen60")
        $dblogdomain{$row[0]}=++$i;
        $dblogdomainid[$i]=$row[0];
  }
  print "\n" if $DO_IR_DEBUG;
  print "$i rows displayed.\n" if $DO_IR_DEBUG;
  $cursor->finish;
  $dbh->commit;

  # After we having finished the selection process
  # we can calculate the weight for each keyword voted:
  # The formula looks like this:
  # Weight (keyword(i)) = sum_of_voted_logical_domains(keyword(i) *
  #                       ( log10
(grand_total_of_items_for_all_categories_touched(keyword(i)
  #                        -
  #                         log10 (sum_of_items_for_all_voted_categories)
  #                        +1
  #                       )
  undef $totalweight;
  undef $totalweightitems;

  my (%temp_catarray, $temp_catarray_element, $tmp_weightkey);

  for ($i=0; $i<$count_titem; $i++)
  {
    if ( ($main::ptitem{$titem[$i]}{"voted"} > 0 ) &&
         ($main::ptitem{$titem[$i]}{"weight"} == undef) &&
       ($main::ptitem{$titem[$i]}{"count hits total"} > 0)  )
    { # collect all categories from "catlist" once. This is necessary
because
      # catlist contains a defined category-name twice and more often
      undef %temp_catarray;
      foreach $ii (@{$ptitem{$titem[$i]}{"catlist"}}) {
        $temp_catarray{@{$ii}[0]}=1; }
      # calculate grand_total_of_items_for_all_categories_touched
      foreach $temp_catarray_element (keys %temp_catarray) {
        $ptitem{$titem[$i]}{"grand total items all log domains"}+=
$dbcat{$temp_catarray_element}{"total number of items"};
      #print $ii.":  ".$dbcat{$temp_catarray_element}{"total number of
items"}."\n";
      }
      if ($DO_IR_DEBUG) {
        print "Weight for ".$titem[$i]." (".$ptitem{$titem[$i]}{"stemmed"}
. ") = ".
```

```
              $ptitem{$titem[$i]}{"count voted domains"}.
             " * (log10(".$ptitem{$titem[$i]}{"grand total items all log
domains"}. ") - ".
             "log10(".$ptitem{$titem[$i]}{"count hits total"}. ") + 1) = ";
      } # if $DO_IR_DEBUG
      $ptitem{$titem[$i]}{"weight"} =
               $ptitem{$titem[$i]}{"count voted domains"} *
               (log10 ($ptitem{$titem[$i]}{"grand total items all log
domains"}) -
              log10 ($ptitem{$titem[$i]}{"count hits total"}) + 1 );
      if ($ptitem{$titem[$i]}{"weight"} < 10.0) {
        $tmp_weightkey= "0". $ptitem{$titem[$i]}{"weight"}}
      elsif ($ptitem{$titem[$i]}{"weight"} < 100.0) {
        $tmp_weightkey= $ptitem{$titem[$i]}{"weight"}."0"}

      # remember weight for item in the %totalweightlist
      # for this purpose we create an hasharray which contains as the key
      # the weight followed by "_*_" (as a seperator) and the stemmed
string of the
      # item

$totalweightlist{$tmp_weightkey."_*_".$main::ptitem{$titem[$i]}{"stemmed"}}
=$ptitem{$titem[$i]}{"weight"};
      $totalweightlist{$tmp_weightkey."_*_".$main::ptitem{
$titem[$i]}{"stemmed"} }{"titem"}=$titem[$i];
      # accumulate total weight for all items
      $totalweight+=$ptitem{$titem[$i]}{"weight"};
      #count number of all weighted items
      $totalweightitems++;

      if ($DO_IR_DEBUG) {
        print $ptitem{$titem[$i]}{"weight"}. "\n";
      } # if $DO_IR_DEBUG
    } # end if

  } # end for
  # We calculate an average figure for the weight
  if ($totalweightitems > 0) {
    $totalweightaverage= $totalweight / $totalweightitems; }
  else {$totalweightaverage= $totalweight}
  # init to zero
  $totalweightlist_element1=0.0;
  $totalweightlist_element2=0.0;
  $totalweightlist_element3=0.0;
  #
  # We fix the first three heaviest weights of all ptitems
  # (which at the end of the sorted list %totalweightlist, for sure
  # there is a more effective way to do this!)
  foreach $totalweightlist_element (sort keys %totalweightlist) {
    $totalweightlist_element3= $totalweightlist_element2;
    $totalweightlist_element2= $totalweightlist_element1;
    $totalweightlist_element1= $totalweightlist_element;
    if ($DO_IR_DEBUG) {
      print "Totalweightlist_element1: $totalweightlist_element1 \n";
      print "Totalweightlist_element2: $totalweightlist_element2 \n";
      print "Totalweightlist_element3: $totalweightlist_element3 \n";
    }
  }
  #if ($totalweightlist_element3 > 0.0)
  #  {$totalweightlimit= $totalweightlist{$totalweightlist_element3};}
  #elsif ($totalweightlist_element2 > 0.0)
  #  {$totalweightlimit= $totalweightlist{$totalweightlist_element2};}
  #elsif ($totalweightlist_element1 > 0.0)
  #  {$totalweightlimit= $totalweightlist{$totalweightlist_element1};}
  #else {$totalweightlimit= 0.0}
  $totalweightlimit= $totalweightlist{$totalweightlist_element3};
```

```
  if ($totalweightlimit == undef)
    {$totalweightlimit= 0.0}

  if ($DO_IR_DEBUG) {
    print "Totalweight: $totalweight for $totalweightitems items gives an
average of $totalweightaverage.\n";
    print "Totalweightlimit: $totalweightlimit.\n";
  }
  # Now we found the three heaviest weighting items so that we can
  # set ptitem{"titem"}{"order"} for these items
  $main::ptitem{ $totalweightlist{$totalweightlist_element1}{"titem"}
}{"order"}=1;
  $main::ptitem{ $totalweightlist{$totalweightlist_element2}{"titem"}
}{"order"}=2;
  $main::ptitem{ $totalweightlist{$totalweightlist_element3}{"titem"}
}{"order"}=3;
  if ($DO_IR_DEBUG) {
    print "\$ptitem{
".$totalweightlist{$totalweightlist_element1}{"titem"}." }{order}".
          "= ".$main::ptitem{
$totalweightlist{$totalweightlist_element1}{"titem"} }{"order"}."\n";
  }

  $dbh->disconnect;
  return 1
} # end sub query_db

sub print_html_vote_data {
#####################################################################
# print out voted rows of data (kinda table view).
#####################################################################
  my($i);
  #$inputtext = $_[0];  # blettsinn
  for ( $i=0; $i < $count_titem; $i++)
  {
    if ( ($main::ptitem{$titem[$i]}{"voted"}) >= 1)
    {
    }
  }
} # sub print_html_vote_data

1;
```

### 3.10.6. Modul tag

```
#!/usr/local/bin/perl -s

# Modul tag for IR-Abilities of STATFINDER daemon
# After all IR-processes the original text will be tagged html-like


sub tag_reduced_text_html
#####################################################################
# Tag a reduced (without Satzzeichen) version of original text
# with html-Tags
#####################################################################
{
  my ($otext, $i, $tagged, $tmp_word);
  $otext = $_[0];
  $tagged = "";
  for ( $i=0; $i < $count_titem; $i++)
  {
    if ( $ptitem{$titem[$i]}{"stopped"} eq $STOPWORD)
    { $tmp_word = $tag_html{$STOPWORD};
      $tmp_word =~ s/$tag_html_replace/$titem[$i]/;
```

```
      $tagged.= $tmp_word . " ";
    }

    if ($main::ptitem{$titem[$i]}{"stemmed"} ne $NOSTEMWORD)
    { $tmp_word = $tag_html{$STEMWORD};
      print "\$stemmed:". $main::ptitem{$main::titem[$i]}{"stemmed"}." " if
$DEBUG;
      $tmp_word =~ s/$tag_html_replace/$main::titem[$i]/;
      $tmp_word.= " (". $main::ptitem{$main::titem[$i]}{"stemmed"}.") ";
      $tmp_word.= " (". $main::titempos[$i] .") ";
      $tagged.=$tmp_word;
    }
  } # end for
  #print $otext;
  return $tagged;
} # end sub tag_text

sub tag_text_html
######################################################################
# Tag original text with html-Tags
# This is function for web-use, which tags a test just with html-
# tags for webbrowser accessing data via client. It is used for
# test purposes and doesn't serve for any production environment.
######################################################################
{
  my ($otext, $i, $tagged, $tmp_word, $last_position, $last_length);
  $otext = $_[0];
  $tagged = "";
  $last_position= -1; # initial position
  $last_length= 0; # initial position
  # Look up sequential list of textitems (titem)
  # and try to reconstruct the original text version and replacing
  # relevant titems (stopwords, voted words etc.) by special tags
  for ( $i=0; $i < $count_titem; $i++)
  {
    #$tmp_tagged= " (" . ($last_position + 1 + $last_length) .", ".
($main::titempos[$i] - ($last_position + $last_length)) .") ";
    #$tagged.=$tmp_tagged;
    $tagged.= substr($otext, ($last_position + 1 + $last_length),
($main::titempos[$i] - ($last_position + $last_length)) );
    # mark item as stopword
    if ( $ptitem{$titem[$i]}{"stopped"} eq $STOPWORD)
    { $tmp_word = $tag_html{$STOPWORD};
      $tmp_word =~ s/$tag_html_replace/$titem[$i]/;
      $tagged.= $tmp_word;
    }

    # mark item that got voted
    elsif ( ($main::ptitem{$titem[$i]}{"voted"}) >= 1)
    { $tmp_word2 = $tag_html{$VOTEWORD};
      print "\$voted:". $main::ptitem{$main::titem[$i]}{"voted"}." " if
$DEBUG;
      $tmp_word2 =~ s/$tag_html_replace/$main::titem[$i]/;
      $tmp_word= "<img SRC=\"ledrobli.gif\" ALT=\"Hits for \'"
                 .$main::ptitem{$titem[$i]}{"stemmed"}."\' in text/DB: "
             .$main::ptitem{$main::titem[$i]}{"count"}."/"
             .$main::ptitem{$main::titem[$i]}{"voted"};
      # "catlist" enthält ein Array of Array
      $old_cat= "";
      foreach $ii (@{$ptitem{$titem[$i]}{"catlist"}}) {
      if ($old_cat eq @{$ii}[0]) { $tmp_word.= ",".@{$ii}[1]; }
      else {$tmp_word.= "\n".@{$ii}[0]."=".@{$ii}[1]; $old_cat=@{$ii}[0]}
      #$tmp_word.= join (" ", @{$ii})."\n";
      };
      $tmp_word.= "\" height=8 width=8>";
```

```
      #$tmp_word =~
s/$tag_html_replace/$main::titem[$i]($main::ptitem{$main::titem[$i]}{"voted
"})/;
      $tagged.=$tmp_word.$tmp_word2;
    }
    # mark item which just is stemword
    elsif ($main::ptitem{$titem[$i]}{"stemmed"} ne $NOSTEMWORD)
    { $tmp_word = $tag_html{$STEMWORD};
      print "\$stemmed:". $main::ptitem{$main::titem[$i]}{"stemmed"}." " if
$DEBUG;
      $tmp_word =~ s/$tag_html_replace/$main::titem[$i]/;
      #$tmp_word.= " (". $main::ptitem{$main::titem[$i]}{"stemmed"}.") ";
      #$tmp_word.= " (". $main::titempos[$i] .") ";
      $tagged.=$tmp_word;
    }
    else {$tagged.= $main::titem[$i]
    }
    $last_position= $main::titempos[$i];
    $last_length= length($main::titem[$i]);
  } # end for
  $tagged.= substr($otext, ($last_position + 1 + $last_length));
  #print $otext;
  return $tagged;
} # end sub tag_text_html

sub tag_text_textfreq
#####################################################################
# Tag original text with html-Tags
# bisher nicht benutzt!!!!!!!!!!!!!!!!!!!!!!!!!!
#####################################################################
{
  my ($otext, $i, $tagged, $tmp_word, $last_position, $last_length);
  $otext = $_[0];
  $tagged = "";
  $last_position= -1; # initial position
  $last_length= 0; # initial position
  # Look up sequential list of textitems (titem)
  # and try to reconstruct the original text version and replacing
  # relevant titems (stopwords, voted words etc.) by special tags
  for ( $i=0; $i < $count_titem; $i++)
  {
    #$tmp_tagged= " (" . ($last_position + 1 + $last_length) .", ".
($main::titempos[$i] - ($last_position + $last_length)) .") ";
    #$tagged.=$tmp_tagged;
    $tagged.= substr($otext, ($last_position + 1 + $last_length),
($main::titempos[$i] - ($last_position + $last_length)) );
    # mark item as stopword
    if ( $ptitem{$titem[$i]}{"stopped"} eq $STOPWORD)
    { $tmp_word = $tag_html{$STOPWORD};
      $tmp_word =~ s/$tag_html_replace/$titem[$i]/;
      $tagged.= $tmp_word;
    }

    # mark item that got voted
    elsif ( ($main::ptitem{$titem[$i]}{"voted"}) >= 1)
    { $tmp_word2 = $tag_html{$VOTEWORD};
      print "\$voted:". $main::ptitem{$main::titem[$i]}{"voted"}." " if
$DEBUG;
      $tmp_word2 =~ s/$tag_html_replace/$main::titem[$i]/;
      $tmp_word= "<img SRC=\"ledrobli.gif\" ALT=\"Hits for \'"
                 .$main::ptitem{$titem[$i]}{"stemmed"}."\' in text/DB: "
             .$main::ptitem{$main::titem[$i]}{"count"}."/"
             .$main::ptitem{$main::titem[$i]}{"voted"};
      # "catlist" enthält ein Array of Array
      $old_cat= "";
      foreach $ii (@{$ptitem{$titem[$i]}{"catlist"}}) {
```

```
      if ($old_cat eq @{$ii}[0]) { $tmp_word.= ",".@{$ii}[1]; }
      else {$tmp_word.= "\n".@{$ii}[0]."=".@{$ii}[1]; $old_cat=@{$ii}[0]}
      #$tmp_word.= join (" ", @{$ii})."\n";
      };
      $tmp_word.= "\" height=8 width=8>";
      #$tmp_word =~
s/$tag_html_replace/$main::titem[$i]($main::ptitem{$main::titem[$i]}{"voted
"})/;
      $tagged.=$tmp_word.$tmp_word2;
    }
    # mark item which just is stemword
    elsif ($main::ptitem{$titem[$i]}{"stemmed"} ne $NOSTEMWORD)
    { $tmp_word = $tag_html{$STEMWORD};
      print "\$stemmed:". $main::ptitem{$main::titem[$i]}{"stemmed"}." " if
$DEBUG;
      $tmp_word =~ s/$tag_html_replace/$main::titem[$i]/;
      #$tmp_word.= " (". $main::ptitem{$main::titem[$i]}{"stemmed"}.") ";
      #$tmp_word.= " (". $main::titempos[$i] .") ";
      $tagged.=$tmp_word;
    }
    else {$tagged.= $main::titem[$i]
    }
    $last_position= $main::titempos[$i];
    $last_length= length($main::titem[$i]);
  } # end for
  $tagged.= substr($otext, ($last_position + 1 + $last_length));
  #print $otext;
  return $tagged;
} # end sub tag_text_textfreq

sub print_reduced_tagged_html_text
{
  my ($ttext);
  $ttext = $_[0];


  print <<tagged_html_text;
  <P>
 <table width="54%" border="0" cellspacing="1" cellpadding="1"
bgcolor="#FFFFCC">
  <tr bgcolor="#C0C0C0">
    <td class="A10" bgcolor="#C0C0C0"><b>Tagged text:</td>
  </tr>
  <tr bgcolor="#C0C0C0">
    <td class="A10" bgcolor="#C0C0C0">$ttext</td>
  </tr>
  </table>
tagged_html_text


} # end sub print_tagged_html_text

sub show_html_legend
{
  print <<printshowlegend;
<P>
<table width="54%" border="0" cellspacing="0" cellpadding="0"
bgcolor="#FFFFCC">
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC"><b>legend</b></span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC"> </td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC">is in <span
class="A10stop">stopwortlist</span></td>
```

```
    <td width="64%" class="A10" bgcolor="#FFFFCC">This term was found in
stopwordfile</td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC">was <span
class="A10stem">stemmed
      (original word)</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC">This word was stemmed
(stemmed word)</td>
  </tr>
  <tr bgcolor="#FFFF66">
    <td class="A10" width="36%" bgcolor="#FFFFCC">got <span
class="A10vote">voted</span></td>
    <td width="64%" class="A10" bgcolor="#FFFFCC">This word got voted</td>
  </tr>
</table>
printshowlegend
} # end sub show_html_legend

sub tag_text_xml
#####################################################################
# Tag original text with xml-Tags
#
# This procedure is no more used becauese of trying to keep the
# the results in a variable which my burst!
#####################################################################
{
  my ($otext, $i, $ii, $tagged, $tmp_word, $last_position, $last_length);
  $otext = $_[0];
  $tagged = "
<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?>

<irdoc>
 <irheader>
";
  # put subdom list
  $tagged.= "   <subdomainlist id=\"1\">\n";
  for ($i=1; $i<=$#dblogdomainid; $i++) {
    $tagged.= "    <subdom id=\"".$i."\">\n";
    $tagged.= "     <name>".$dblogdomainid[$i]."</name>\n";
    $tagged.= "    </subdom>\n";
  }
  $tagged.= "   </subdomainlist>\n";
  # put cat list
  $tagged.= "   <catlist id=\"1\">\n";
  for ($i=1; $i<=$#dbcatid; $i++) {
    $tagged.= "    <cat id=\"".$i."\">\n";
    $tagged.= "     <name>".$dbcatid[$i]."</name>\n";
    $tagged.= "    </cat>\n";
  }
  $tagged.= "   </catlist>\n";
  # put term list
  $tagged.= "   <termlist id=\"1\">\n";
  for ($i=1; $i<$count_titem; $i++)
  {
    if ($main::ptitem{$titem[$i]}{"voted"} > 0 )
    {
      # "catlist" enthält ein Array of Array
      $old_cat= "";
      foreach $ii (@{$ptitem{$titem[$i]}{"catlist"}}) {
        $tagged.= "    <term id=\"".$main::ptitem{$titem[$i]}{"wkey"}."\"
";
        $tagged.= "subdom=\"0\" ";
        $tagged.= "cat=\"";
        #$tagged.= @{$ii}[0]."/";
      $tagged.= $dbcat{@{$ii}[0]}."\" ";
```

```
        $tagged.= "key=\"".@{$ii}[1]."\" ";
        $tagged.= "order=\"0\">";
        #$tagged.= @{$ii}[0]."=".@{$ii}[1]; $old_cat=@{$ii}[0];

        $tagged.= "\n";
        $tagged.= "      <name>".$titem[$i]."</name>\n";
        $tagged.= "    </term>\n";
      } # end foreach
    }
  }
  $tagged.= "   </termlist>\n";
  $tagged.= "  </irheader>\n";
  $tagged.= "  <taggedtext>\n";

  $last_position= -1; # initial position
  $last_length= 0; # initial position
  # Look up sequential list of textitems (titem)
  # and try to reconstruct the original text version and replacing
  # relevant titems (here: just voted words) by xml tags
  for ( $i=0; $i < $count_titem; $i++)
  {
    $tagged.= substr($otext, ($last_position + 1 + $last_length),
($main::titempos[$i] - ($last_position + $last_length)) );
    # mark item that got voted
    if ( ($main::ptitem{$titem[$i]}{"voted"}) > 0)
    {
      $tagged.= "<term id=\"".$main::ptitem{$titem[$i]}{"wkey"}."\">";
      $tagged.= $main::titem[$i]."</term>";
    }
    else {$tagged.= $main::titem[$i]
    }
    $last_position= $main::titempos[$i];
    $last_length= length($main::titem[$i]);
  } # end for
  $tagged.= substr($otext, ($last_position + 1 + $last_length));
  $tagged.= "  </taggedtext>\n";
  $tagged.= "</irdoc>\n";
  $tagged.= "";
  return $tagged;
} # end sub tag_text_xml

sub tag_text_xml_and_print
#####################################################################
# Tag original text with xml-Tags and print them immediatly
# Better flush the output continiously than crashing a caching var...
#####################################################################
{
  my ($otext, $i, $ii, $tagged, $tmp_word, $last_position, $last_length);
  $otext = $_[0];
  print "
<?xml version=\"1.0\" encoding=\"ISO-8859-1\"?>

<irdoc>
 <irheader>
";
  # put subdom list
  print "   <subdomainlist id=\"1\">\n";
  for ($i=1; $i<=$#dblogdomainid; $i++) {
    print "    <subdom id=\"".$i."\">\n";
    print "     <name>".$dblogdomainid[$i]."</name>\n";
    print "    </subdom>\n";
  }
  print "   </subdomainlist>\n";
  # put cat list
  print "   <catlist id=\"1\">\n";
  for ($i=1; $i<=$#dbcatid; $i++) {
```

```
    print "    <cat id=\"".$i."\">\n";
    print "     <name>".$dbcatid[$i]."</name>\n";
    print "    </cat>\n";
  }
  print "   </catlist>\n";
  # put term list
  print "   <termlist id=\"1\">\n";
  for ($i=0; $i<$count_titem; $i++)
  {
    # At this point we select only these items which have a weight more
than the
    # calculated level of relevance
    # if ($main::ptitem{$titem[$i]}{"weight"} >= $totalweightlimit )

    # Just check if this term got voted without checking a limit
    if ($main::ptitem{$titem[$i]}{"voted"} > 0 )
    {
      # "catlist" enthält ein Array of Array
      $old_cat= "";
      foreach $ii (@{$ptitem{$titem[$i]}{"catlist"}}) {
        print "    <term id=\"".$main::ptitem{$titem[$i]}{"wkey"}."\" ";
        print
"subdom=\"".$dblogdomain{$dbcat{@{$ii}[0]}{"logicaldomain"}}."\" ";
        print "cat=\"";
        #print @{$ii}[0]."/";
      print $dbcat{@{$ii}[0]}."\" ";
        print "key=\"".@{$ii}[1]."\" ";
      #if ( ($ptitem{$titem[$i]}{"order"} > 0) &&
($ptitem{$titem[$i]}{"order"} <= 3) )
      if ($ptitem{$titem[$i]}{"order"})
      {  print "order=\"".$ptitem{$titem[$i]}{"order"}."\">" }
        else {print "order=\"0\">";}
        #print @{$ii}[0]."=".@{$ii}[1]; $old_cat=@{$ii}[0];

        print "\n";
        print "      <name>".$titem[$i]."</name>\n";
        print "    </term>\n";
      } # end foreach
    }
  }
  print "   </termlist>\n";
  print "  </irheader>\n";
  print "  <taggedtext>\n";

  $last_position= -1; # initial position
  $last_length= 0; # initial position
  # Look up sequential list of textitems (titem)
  # and try to reconstruct the original text version and replacing
  # relevant titems (here: just voted words) by xml tags
  for ( $i=0; $i < $count_titem; $i++)
  {
    print substr($otext, ($last_position + 1 + $last_length),
($main::titempos[$i] - ($last_position + $last_length)) );
    # mark item that has a special order number (1, 2 or 3)
    # if ( ($main::ptitem{$titem[$i]}{"order"}) > 0)
    # mark item that got voted
    if ($main::ptitem{$titem[$i]}{"voted"} > 0 )
    {
      print "<term id=\"".$main::ptitem{$titem[$i]}{"wkey"}."\">";
      print $main::titem[$i]."</term>";
      if ($main::ptitem{$titem[$i]}{"order"} )
      { print " (".$main::ptitem{$titem[$i]}{"order"}.") ";
      }
    }
    else {print $main::titem[$i]
    }
```

```
    $last_position= $main::titempos[$i];
    $last_length= length($main::titem[$i]);
  } # end for
  print substr($otext, ($last_position + 1 + $last_length));
  print "  </taggedtext>\n";
  print "</irdoc>\n";
  print "";
} # end sub tag_text_xml_and_print

1;
```

### 3.10.7. Modul main

```
#!/usr/local/bin/perl -w

# Datei ir_main.pl
# Aufrufbeispiele
#    ir_main.pl -f test.txt

# Modul main for IR-Abilities of STATFINDER daemon
#
# Comments on the whole project:
# split up parse_text in get_original_text and parse_text
#
use strict;
use File::Basename;
use Getopt::Long;
require "ir_prefs.pl";
require "ir_parse.pl";
require "ir_stem.pl";
require "ir_tag.pl";
require "ir_util.pl";
require "ir_vote.pl";

# Ein Dateiname wird als Parameter übergeben.
#$main::order = $Getopt::PERMUTE;
GetOptions('f:s');

# split filename
#($filename,$prefix,$tail) = fileparse($opt_f, ".txt");

# use file from parameters, if not asigned: STDIN
$main::Ifile   = (defined $main::opt_f ? $main::opt_f : "-");
#print $main::opt_f, " ",$main::Ifile, "\n";
$main::original_text=&parse_text ( &read_text_from_file ($main::Ifile) );

{ # just a block
  my ($i);

  $i=0;
  for ( $i; $i <= $#main::titem; $i++)
  {
    if (defined $main::titem[$i])
    { $main::ptitem{$main::titem[$i]}{"stopped"} = &stop_word
($main::titem[$i]);
    }
    #print $main::ptitem{$main::titem[$i]}{"stopped"},"\n";
    #if ((defined $main::titem[$i]) &&
($main::ptitem{$main::titem[$i]}{"stemmed"} ne $main::STOPWORD))
    { $main::ptitem{$main::titem[$i]}{"stemmed"}= &stem_word
($main::titem[$i]); }
    #print "$main::titem[$i] / $main::ptitem{$main::titem[$i]}{\"stemmed\"}
($i:$main::titempos[$i], $main::ptitem{$main::titem[$i]}{\"count\"})|";
  }
}
```

```
#&show_statistics;
#$main::tagged_text=&tag_text($main::original_text);
#print $main::original_text, "\n";
&get_dbdata;
```

## *3.11. Vollständiges Protokoll einer Dämonanfrage mit Debug-Informationen*

```
193.174.139.67 wird versucht...
Verbunden mit diwsysv
Escape-Zeichen ist '^]'.
WIDAB Server ready...
Version 1.11
SOS
Testtext:


Von der Etablierung des Binnenmarktes in der EU wurden seinerzeit
                  krõftige Impulse f³r den Handel zwischen den Partnerlõndern
                  erwartet. Diese Erwartung hat sich weitgehend erf³llt. So ist die
                  nominale Ausfuhr Deutschlands in die ³brigen EWU-Lõnder von 1993
                  bis 2000 um 73 % gestiegen. Zwar hat der Außenhandel der EWU in
                  dieser Zeit in õhnlichem Tempo wie der Intrahandel expandiert
                  (Abbildung 1), was daf³r spricht, dass der Austausch innerhalb der
                  EWU nicht stõrker zugenommen hat als die allgemeine
                  weltwirtschaftliche Verflechtung. Dabei ist aber zu ber³cksichtigen,
                  dass sich viele osteuropõische Lõnder erst in letzter Zeit in den
                  Welthandel integriert haben und sich der Handel mit ihnen daher
                  besonders dynamisch entwickelte.

                  Deutschland ist mit einem Gewicht von reichlich einem F³nftel der
                  wichtigste Handelspartner im Intra-EWU-Handel. [1] Die Exporte
                  Deutschlands in die ³brigen EWU-Lõnder sind in nominaler Rechnung
                  in der zweiten Hõlfte der 90er Jahre weniger stark gestiegen als der
                  Intrahandel der anderen EWU- Lõnder (Abbildung 2). Diese
                  Entwicklung ist zunõchst ³berraschend, denn in dieser Zeit hatte sich
                  die Wettbewerbsfõhigkeit der deutschen Unternehmen infolge deutlich
                  niedrigerer Lohnst³ckkosten als beim Durchschnitt der ³brigen
                  EWU-Lõnder erheblich verbessert (Abbildung 3). Dies kann dazu
                  gef³hrt haben, dass die Exportpreise Deutschlands weniger stark
                  gestiegen sind als bei den anderen EWU-Lõndern, so dass der Wert
                  der Ausfuhr entsprechend schwõcher zunahm. In realer Rechnung
                  k÷nnte sich ein anderes Bild ergeben. Eine weitere Erklõrung f³r den
                  schwõcheren Intrahandel der inlõndischen deutschen Unternehmen
                  kann darin gesehen werden, dass die Mõrkte in den einzelnen
                  EWU-Lõndern von deutschen Tochterunternehmen im Land selbst
                  oder von einem anderen EWU-Land aus beliefert wurden. Im
                  Extrahandel waren die deutschen Unternehmen erfolgreicher als
                  diejenigen der ³brigen EWU-Lõnder; dies d³rfte vor allem auf die
                  besonders engen Handelsverflechtungen mit den osteuropõischen
                  Lõndern zur³ckzuf³hren sein.

Usergroup:
statfinder
Language:
de
Select
                     name from met_benutzergruppe
                    where name= 'statfinder'
                     order by name
NAME
Usergroups in DB:
--------
        NAME
        statfinder

1 rows displayed.
DBMulitTermKeywords german:
Bah Post : Bahn/Post
Bank Namensschuldverschreibung : Bank-Namensschuldverschreibungen
Berli Ost : Berlin-Ost
Berli West : Berlin-West
Betriebsteile Ergebnis : Betriebsteile-Ergebnisse
Brutto Absatz : Brutto-Absatz
Bundesanstalt für Arbeit : Bundesanstalt für Arbeit
Bundesrepublik Deutschland : Bundesrepublik Deutschland
Börse Frankfurt : Börse Frankfurt
D Mark : D-Mark
DM Commercial Papi : DM-Commercial-Papier
DV Geräte : DV-Geräte
Deutsch Mark : Deutsche Mark
E Technik : E-Technik
ECU Leitkur : ECU-Leitkurs
ECU Wert : ECU-Wert
EG Haushalt : EG-Haushalt
EWRE Relation : EWRE-Relation
Frankfurt Börse : Frankfurter Börse
```

```
Frankfurt Markt : Frankfurter Markt
Futur Geschäft : Futures-Geschäft
Gesamtrechnung d Bankenstatistik : Gesamtrechnung der Bankenstatistik
Gold Parität : Gold-Parität
ISO Währungscode : ISO-Währungscode
Kaffee Ersatz : Kaffee-Ersatz
Kassa Kur : Kassa-Kurs
Leistung d Architekte : Leistungen der Architekten
Maßnahme §242s AFG : Maßnahmen §242s AFG
Maßnahme §249h AFG : Maßnahmen §249h AFG
Mecklenburg Vorpomm : Mecklenburg-Vorpommern
Mindestreserve Soll : Mindestreserve-Soll
NE Metalle : NE-Metalle
NE Metallerzbergbau : NE-Metallerzbergbau
Netto Absatz : Netto-Absatz
Null Kupo Anleihe : Null-Kupon-Anleihen
RFN Technik : RFN-Technik
SZR Leitkur : SZR-Leitkurs
Sachse Anhalt : Sachsen-Anhalt
Schleswig Holstei : Schleswig-Holstein
Term of Trade : Terms of Trade
US Dollar : US-Dollar
US Dollar Parität : US-Dollar-Parität
alte Bundesländ : alte Bundesländer
bankenstatistisch Gesamtrechnung : bankenstatistische Gesamtrechnung
bearbeiten Gewerbe : bearbeitendes Gewerbe
bilanzunwirksam Geschäfte : bilanzunwirksame Geschäfte
deutsch Bundesbank : deutsche Bundesbank
europäisch Währungssystem : europäisches Währungssystem
festverzinslich Wertpapier : festverzinsliche Wertpapiere
finanziell Entwicklung : finanzielle Entwicklung
früh Bundesgebiet : früheres Bundesgebiet
gewerblich Hochbau : gewerblichen Hochbaus
gewerblich Tiefbau : gewerblichen Tiefbaus
giral Verfügung : girale Verfügungen
im Gebiet : im Gebiet
kassenmäßig Entwicklung : kassenmäßige Entwicklung
konsolidierene Bilanz d Bankensystem : konsolidierte Bilanz des Bankensystems
leisten Arbeit : geleistete Arbeit
leisten Arbeiterstunde : geleistete Arbeiterstunden
monetär Indikator : monetäre Indikatoren
nach Sitz d Betriebe : nach Sitz der Betriebe
neue Bundesländ : neue Bundesländer
neue Land : neue Länder
offene Stell : offene Stellen
privaen Haushalt : privater Haushalt
privaen Verbrauch : privatem Verbrauch
privat Haushalt : privaten Haushalts
privat Haushalte : privaten Haushalte
privat Organisation : private Organisationen
privat Verbrauch : privaten Verbrauches
produzieren Gewerbe : produzierendes Gewerbe
produzierend Gewerb : produzierenden Gewerbes
produzierend Gewerbe : produzierenden Gewerbe
saisonbereinigene Wirtschaftszahle : saisonbereinigte Wirtschaftszahlen
verarbeiten Gewerbe : verarbeitendes Gewerbe
verarbeitend Gewerb : verarbeitenden Gewerbes
verarbeitend Gewerbe : verarbeitenden Gewerbe
vereinigen Königreich : vereinigtes Königreich
volkswirtschaftlich Gesamtrechnung : volkswirtschaftliche Gesamtrechnung
vorgezog Ruhestand : vorgezogener Ruhestand
warenproduzieren Gewerbe : warenproduzierendes Gewerbe
warenproduzierend Gewerb : warenproduzierenden Gewerbes
warenproduzierend Gewerbe : warenproduzierenden Gewerbe
zivile Zwecke : zivile Zwecke
öffentlich Finanz : öffentliche Finanzen
öffentlich Hand : öffentliche Hand
öffentlich Haushalte : öffentliche Haushalte
öffentlich Hochbau : öffentlichen Hochbaus
öffentlich Tiefbau : öffentlichen Tiefbau
öffentlich Verbrauch : öffentlichen Verbrauch
übrig Welt : übrige Welt

List of pre-voted MulitTermKeywords
MTKw item: vorgezog Ruhestand (2 items)
Joined item:  bis um
Joined item:  reichlich ein
Joined item:  deutsche Tochterunternehme
Joined item:  ein ander
Joined item:  all auf
MTKw item: öffentlich Hand (2 items)
MTKw item: volkswirtschaftlich Gesamtrechnung (2 items)
```

```
Joined item:  bis um
Joined item:  reichlich ein
Joined item:  deutsche Tochterunternehme
Joined item:  ein ander
MTKw item: nach Sitz d Betriebe (4 items)
MTKw item: leisten Arbeit (2 items)
MTKw item: neue Land (2 items)
MTKw item: Bundesrepublik Deutschland (2 items)
MTKw item: Gesamtrechnung d Bankenstatistik (3 items)
MTKw item: Null Kupo Anleihe (3 items)
MTKw item: Berli West (2 items)
MTKw item: Betriebsteile Ergebnis (2 items)
MTKw item: leisten Arbeiterstunde (2 items)
MTKw item: gewerblich Hochbau (2 items)
MTKw item: Deutsch Mark (2 items)
MTKw item: übrig Welt (2 items)
MTKw item: produzierend Gewerbe (2 items)
MTKw item: gewerblich Tiefbau (2 items)
MTKw item: bankenstatistisch Gesamtrechnung (2 items)
MTKw item: Futur Geschäft (2 items)
MTKw item: monetär Indikator (2 items)
MTKw item: Mecklenburg Vorpomm (2 items)
MTKw item: öffentlich Hochbau (2 items)
MTKw item: Kassa Kur (2 items)
MTKw item: Brutto Absatz (2 items)
MTKw item: EG Haushalt (2 items)
MTKw item: im Gebiet (2 items)
Joined item:  d EU
Joined item:  di nominal
Joined item:  di ³brig
Joined item:  dies Zeit
Joined item:  õhnlich Tempo
Joined item:  ab zu
Joined item:  letzen Zeit
Joined item:  d Welthandel
Joined item:  mit ein
Joined item:  Intra EWU
Joined item:  di ³brig
Joined item:  nominal Rechnung
Joined item:  d zweiten
Joined item:  zunõchsen ³berraschennd
Joined item:  dies Zeit
Joined item:  d einzelne
Joined item:  Land selbsen
MTKw item: öffentlich Haushalte (2 items)
MTKw item: öffentlich Tiefbau (2 items)
MTKw item: Mindestreserve Soll (2 items)
MTKw item: europäisch Währungssystem (2 items)
MTKw item: deutsch Bundesbank (2 items)
Joined item:  Etablierung d
Joined item:  Binnenmarkt i
Joined item:  EU werden
Joined item:  Handel zwisch
Joined item:  Partnerlõnd erwartet
Joined item:  Außenhandel d
Joined item:  EWU i
Joined item:  Intrahandel expandieren
Joined item:  Austausch innerhalb
Joined item:  EWU nichen
Joined item:  Welthandel integrieren
Joined item:  Handel mit
Joined item:  wichtigene Handelspartn
Joined item:  zweiten Hõlfte
Joined item:  Jahr wenig
Joined item:  Intrahandel d
Joined item:  ander EWU
Joined item:  deutsche Unternehme
Joined item:  ³brig EWU
Joined item:  ander EWU
Joined item:  Wert d
Joined item:  Ausfuhr sprechen
Joined item:  schwõch Intrahandel
Joined item:  inlõndisch deutsche
Joined item:  einzelne EWU
Joined item:  ³brig EWU
Joined item:  osteuropõisch Lõnd
MTKw item: bilanzunwirksam Geschäfte (2 items)
MTKw item: öffentlich Finanz (2 items)
MTKw item: Netto Absatz (2 items)
MTKw item: Sachse Anhalt (2 items)
MTKw item: neue Bundesländ (2 items)
MTKw item: Maßnahme §249h AFG (3 items)
```

```
MTKw item: konsolidierene Bilanz d Bankensystem (4 items)
MTKw item: Leistung d Architekte (3 items)
MTKw item: Frankfurt Börse (2 items)
MTKw item: privat Haushalte (2 items)
MTKw item: vereinigen Königreich (2 items)
MTKw item: finanziell Entwicklung (2 items)
MTKw item: verarbeitend Gewerb (2 items)
MTKw item: DM Commercial Papi (3 items)
MTKw item: offene Stell (2 items)
MTKw item: verarbeiten Gewerbe (2 items)
MTKw item: Bank Namensschuldverschreibung (2 items)
MTKw item: ISO Währungscode (2 items)
Joined item:  real Rechnung
MTKw item: warenproduzierend Gewerbe (2 items)
Joined item:  di deutsche
MTKw item: NE Metallerzbergbau (2 items)
MTKw item: zivile Zwecke (2 items)
MTKw item: Term of Trade (3 items)
MTKw item: Bundesanstalt für Arbeit (3 items)
MTKw item: Berli Ost (2 items)
MTKw item: alte Bundesländ (2 items)
MTKw item: RFN Technik (2 items)
MTKw item: früh Bundesgebiet (2 items)
MTKw item: D Mark (2 items)
MTKw item: privat Haushalt (2 items)
MTKw item: kassenmäßig Entwicklung (2 items)
MTKw item: Frankfurt Markt (2 items)
MTKw item: bearbeiten Gewerbe (2 items)
MTKw item: warenproduzierend Gewerb (2 items)
Joined item:  di deutsche
MTKw item: warenproduzieren Gewerbe (2 items)
Joined item:  di deutsche
MTKw item: Maßnahme §242s AFG (3 items)
MTKw item: produzierend Gewerb (2 items)
MTKw item: produzieren Gewerbe (2 items)
MTKw item: Kaffee Ersatz (2 items)
MTKw item: US Dollar Parität (3 items)
MTKw item: Gold Parität (2 items)
MTKw item: Schleswig Holstei (2 items)
MTKw item: privat Organisation (2 items)
MTKw item: US Dollar (2 items)
MTKw item: festverzinslich Wertpapier (2 items)
MTKw item: öffentlich Verbrauch (2 items)
MTKw item: E Technik (2 items)
MTKw item: privaen Haushalt (2 items)
MTKw item: giral Verfügung (2 items)
MTKw item: saisonbereinigene Wirtschaftszahle (2 items)
MTKw item: EWRE Relation (2 items)
MTKw item: Bah Post (2 items)
MTKw item: Börse Frankfurt (2 items)
MTKw item: DV Geräte (2 items)
MTKw item: NE Metalle (2 items)
MTKw item: privaen Verbrauch (2 items)
MTKw item: privat Verbrauch (2 items)
MTKw item: verarbeitend Gewerbe (2 items)
MTKw item: ECU Leitkur (2 items)
MTKw item: SZR Leitkur (2 items)
MTKw item: ECU Wert (2 items)
Number of prevoted MultiTermItems: 0

declare global temporary table session.votetmp
                      (eintrag varchar(100), treffer_im_text integer)
                    on commit preserve rows with norecovery, noduplicates
Insert into session.kat_usergroup
                         (kategorie, logbestand)

                      Select lmlb.tabellenname,
                             lmben.logbestandsname
                        from
                            lnk_met_log_bestand lmlb,
                            lnk_met_benutzergruppe lmben
                        where lmben.benutzergruppe = 'statfinder' and
                             lmlb.logbestandsname = lmben.logbestandsname and
                             lmlb.tabellenname like 'bam_%'

Select
                      *
                        from session.kat_usergroup
                      order by kategorie
KATEGORIE LOGBESTAND BESTAND
Content of session.kat_usergroup:
--------
```

```
        KATEGORIE      LOGBESTAND     BESTAND
        bam_bau_bereiche       DIW-Bauvolumen
        bam_bau_bereiche       DIW-Bauvolumen60
        bam_bau_laender        DIW-Bauvolumen
        bam_bau_produzenten    DIW-Bauvolumen
        bam_bau_produzenten60  DIW-Bauvolumen60
        bam_bau_themen DIW-Bauvolumen
        bam_brand_arbeitsmarkt WIDAB_Brandenburg_Arbeitsmarkt
        bam_brand_aregion      WIDAB_Brandenburg_Arbeitsmarkt
        bam_brand_bbranchen    WIDAB_Brandenburg_Bau
        bam_brand_bip  WIDAB_Brandenburg_BIP
        bam_brand_bregion      WIDAB_Brandenburg_Bau
        bam_brand_bthemen      WIDAB_Brandenburg_Bau
        bam_brand_eregion      WIDAB_Brandenburg_Erwerb
        bam_brand_erwt WIDAB_Brandenburg_Erwerb
        bam_brand_ethemen      WIDAB_Brandenburg_Erwerb
        bam_brand_iregion      WIDAB_Brandenburg_BIP
        bam_brand_nace WIDAB_Brandenburg_Industrie
        bam_brand_nregion      WIDAB_Brandenburg_Industrie
        bam_brand_themen       WIDAB_Brandenburg_Industrie
        bam_eurostat_branch    Eurostat-Linda
        bam_eurostat_branch_de Eurostat-Linda
        bam_eurostat_branch_en Eurostat-Linda
        bam_eurostat_branch_fr Eurostat-Linda
        bam_eurostat_country   Eurostat-Linda
        bam_eurostat_country_de       Eurostat-Linda
        bam_eurostat_country_en       Eurostat-Linda
        bam_eurostat_country_fr       Eurostat-Linda
        bam_eurostat_themes    Eurostat-Linda
        bam_eurostat_themes_de Eurostat-Linda
        bam_eurostat_themes_en Eurostat-Linda
        bam_eurostat_themes_fr Eurostat-Linda
        bam_ifo_kbran  ifo-KT
        bam_ifo_kthem_fapsy    ifo-KT
        bam_vgr_konten DIW-vgr_Konten
        bam_vgr_kthem  DIW-vgr_Konten
        bam_vgr_region DIW-vgr_Sektoren
        bam_vgr_region DIW-vgr_Konten
        bam_vgr_sektor DIW-vgr_Sektoren
        bam_vgr_sthem  DIW-vgr_Sektoren
        bam_vgrneu_themen      DIW-vgrneu
        bam_vgrneu_werteart    DIW-vgrneu

41 rows displayed.
Insert into session.result
                         (wkey,stemeintrag,eintrag,
                         treffer_im_text,
                         kategorie,satznummer,
                         Treffer_in_DB, logbestand)

                     Select distinct
                         mtw.wkey, mtw.stemeintrag, mtw.eintrag,
                          ses.treffer_im_text,
                         trim(sesku.kategorie), mlw.satznummer,
                         count(*) as Treffer_in_DB, sesku.bestand
                       from
                         met_tab_wort_de mtw,
                         session.votetmp ses,
                        met_lnk_wort_de mlw,
                        session.kat_usergroup sesku
                       where mlw.wkey = mtw.wkey and
                          ses.eintrag = mtw.stemeintrag and
                          sesku.kategorie = trim(mlw.kategorie)
                     group by
                         mtw.stemeintrag, mtw.eintrag, sesku.kategorie,
                         mlw.satznummer, mtw.wkey, treffer_im_text,
                         sesku.bestand
Select
                    wkey, stemeintrag, eintrag,
                   kategorie, satznummer, logbestand,
                   treffer_im_text, Treffer_in_DB
                     from session.result
                    where treffer_im_text > 0
                     group by stemeintrag, eintrag, kategorie, satznummer, logbestand, wkey,
treffer_im_text, Treffer_in_DB
                      order by treffer_im_text desc
WKEY STEMEINTRAG EINTRAG KATEGORIE SATZNUMMER LOGBESTAND TREFFER_IM_TEXT TREFFER_IN_DB
Votelist
--------
        WKEY    STEMEINTRAG    EINTRAG KATEGORIE      SATZNUMMER     LOGBESTAND
        TREFFER_IM_TEXT        TREFFER_IN_DB
        290     Deutschland    Deutschland    bam_brand_aregion      1              4       1
```

```
290     Deutschland     Deutschland     bam_brand_bregion       1               4       1
290     Deutschland     Deutschland     bam_brand_eregion       1               4       1
290     Deutschland     Deutschland     bam_brand_iregion       1               4       1
290     Deutschland     Deutschland     bam_brand_nregion       1               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  1                4       1
290     Deutschland     Deutschland     bam_brand_aregion       2               4       1
290     Deutschland     Deutschland     bam_brand_bregion       2               4       1
290     Deutschland     Deutschland     bam_brand_eregion       2               4       1
290     Deutschland     Deutschland     bam_brand_iregion       2               4       1
290     Deutschland     Deutschland     bam_brand_nregion       2               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  2                4       1
290     Deutschland     Deutschland     bam_brand_aregion       3               4       1
290     Deutschland     Deutschland     bam_brand_bregion       3               4       1
290     Deutschland     Deutschland     bam_brand_eregion       3               4       1
290     Deutschland     Deutschland     bam_brand_iregion       3               4       1
290     Deutschland     Deutschland     bam_brand_nregion       3               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  3                4       1
290     Deutschland     Deutschland     bam_brand_aregion       4               4       1
290     Deutschland     Deutschland     bam_brand_bregion       4               4       1
290     Deutschland     Deutschland     bam_brand_eregion       4               4       1
290     Deutschland     Deutschland     bam_brand_iregion       4               4       1
290     Deutschland     Deutschland     bam_brand_nregion       4               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  4                4       1
290     Deutschland     Deutschland     bam_brand_aregion       5               4       1
290     Deutschland     Deutschland     bam_brand_bregion       5               4       1
290     Deutschland     Deutschland     bam_brand_eregion       5               4       1
290     Deutschland     Deutschland     bam_brand_iregion       5               4       1
290     Deutschland     Deutschland     bam_brand_nregion       5               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  5                4       1
290     Deutschland     Deutschland     bam_brand_aregion       6               4       1
290     Deutschland     Deutschland     bam_brand_bregion       6               4       1
290     Deutschland     Deutschland     bam_brand_eregion       6               4       1
290     Deutschland     Deutschland     bam_brand_iregion       6               4       1
290     Deutschland     Deutschland     bam_brand_nregion       6               4       1
290     Deutschland     Deutschland     bam_eurostat_country_de         6               4
1
290     Deutschland     Deutschland     bam_ifo_kbran  6                4       1
290     Deutschland     Deutschland     bam_brand_aregion       7               4       1
290     Deutschland     Deutschland     bam_brand_bregion       7               4       1
290     Deutschland     Deutschland     bam_brand_eregion       7               4       1
290     Deutschland     Deutschland     bam_brand_iregion       7               4       1
290     Deutschland     Deutschland     bam_brand_nregion       7               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  7                4       1
290     Deutschland     Deutschland     bam_brand_aregion       8               4       1
290     Deutschland     Deutschland     bam_brand_bregion       8               4       1
290     Deutschland     Deutschland     bam_brand_eregion       8               4       1
290     Deutschland     Deutschland     bam_brand_iregion       8               4       1
290     Deutschland     Deutschland     bam_brand_nregion       8               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  8                4       1
290     Deutschland     Deutschland     bam_brand_aregion       9               4       1
290     Deutschland     Deutschland     bam_brand_bregion       9               4       1
290     Deutschland     Deutschland     bam_brand_eregion       9               4       1
290     Deutschland     Deutschland     bam_brand_iregion       9               4       1
290     Deutschland     Deutschland     bam_brand_nregion       9               4       1
290     Deutschland     Deutschland     bam_ifo_kbran  9                4       1
290     Deutschland     Deutschland     bam_brand_aregion       10              4       1
290     Deutschland     Deutschland     bam_brand_bregion       10              4       1
290     Deutschland     Deutschland     bam_brand_eregion       10              4       1
290     Deutschland     Deutschland     bam_brand_iregion       10              4       1
290     Deutschland     Deutschland     bam_brand_nregion       10              4       1
290     Deutschland     Deutschland     bam_ifo_kbran  10               4       1
290     Deutschland     Deutschland     bam_brand_aregion       11              4       1
290     Deutschland     Deutschland     bam_brand_bregion       11              4       1
290     Deutschland     Deutschland     bam_brand_eregion       11              4       1
290     Deutschland     Deutschland     bam_brand_iregion       11              4       1
290     Deutschland     Deutschland     bam_brand_nregion       11              4       1
290     Deutschland     Deutschland     bam_ifo_kbran  11               4       1
290     Deutschland     Deutschland     bam_brand_aregion       12              4       1
290     Deutschland     Deutschland     bam_brand_bregion       12              4       1
290     Deutschland     Deutschland     bam_brand_eregion       12              4       1
290     Deutschland     Deutschland     bam_brand_iregion       12              4       1
290     Deutschland     Deutschland     bam_brand_nregion       12              4       1
290     Deutschland     Deutschland     bam_ifo_kbran  12               4       1
290     Deutschland     Deutschland     bam_brand_aregion       13              4       1
290     Deutschland     Deutschland     bam_brand_bregion       13              4       1
290     Deutschland     Deutschland     bam_brand_eregion       13              4       1
290     Deutschland     Deutschland     bam_brand_iregion       13              4       1
290     Deutschland     Deutschland     bam_brand_nregion       13              4       1
290     Deutschland     Deutschland     bam_ifo_kbran  13               4       1
290     Deutschland     Deutschland     bam_brand_aregion       14              4       1
290     Deutschland     Deutschland     bam_brand_bregion       14              4       1
290     Deutschland     Deutschland     bam_brand_eregion       14              4       1
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 290 | Deutschland | Deutschland | bam_brand_iregion | 14 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_brand_nregion | 14 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 14 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_brand_aregion | 15 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_brand_bregion | 15 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_brand_eregion | 15 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_brand_iregion | 15 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_brand_nregion | 15 | | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 15 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 16 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 17 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 18 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 19 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 20 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 21 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 22 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 23 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 24 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 25 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 26 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 27 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 28 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 29 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 30 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 31 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 32 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 33 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 34 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 35 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 36 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 37 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 38 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 39 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 40 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 41 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 42 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 43 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 44 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 45 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 46 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 47 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 48 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 49 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 50 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 51 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 52 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 53 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 54 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 55 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 56 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 57 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 58 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 59 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 60 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 61 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 62 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 63 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 64 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 65 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 66 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 67 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 68 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 69 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 70 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 71 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 72 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 73 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 74 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 75 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 76 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 78 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 79 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 80 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 81 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 82 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 83 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 84 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 85 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 86 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 87 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 88 | 4 | 1 | |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 89 | 4 | 1 | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 90 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 91 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 92 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 93 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 94 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 95 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 96 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 97 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 98 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 99 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 100 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 101 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 102 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 103 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 104 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 105 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 106 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 107 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 108 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 110 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 111 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 112 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 113 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 114 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 115 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 116 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 117 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 118 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 119 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 120 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 121 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 122 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 123 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 124 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 125 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 126 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 127 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 128 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 129 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 130 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 131 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 132 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 133 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 134 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 135 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 136 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 137 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 138 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 139 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 140 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 141 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 142 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 143 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 144 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 145 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 146 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 147 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 148 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 149 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 150 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 151 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 152 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 153 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 154 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 155 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 156 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 157 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 158 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 159 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 160 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 161 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 162 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 163 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 164 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 165 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 166 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 167 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 168 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 169 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 170 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 171 | 4 | 1 |
| 290 | Deutschland | Deutschland | bam_ifo_kbran | 172 | 4 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 290 | Deutschland | Deutschland | bam_vgr_region | 9 | | 4 | 2 |
| 290 | Deutschland | Deutschland | bam_vgr_region | 1001 | | 4 | 2 |
| 290 | Deutschland | Deutschland | bam_vgr_region | 1002 | | 4 | 2 |
| 1449 | Handel Handel | bam_brand_bip | 6 | 3 | 1 | | |
| 1449 | Handel Handel | bam_brand_erwt | 6 | 3 | 1 | | |
| 1449 | Handel Handel | bam_brand_bip | 7 | 3 | 1 | | |
| 1449 | Handel Handel | bam_brand_erwt | 7 | 3 | 1 | | |
| 91 | Unternehme | Unternehmen | bam_brand_bip | 13 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_brand_erwt | 13 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 14 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_brand_bip | 15 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 15 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 16 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_brand_bip | 17 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 17 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 17 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_brand_bip | 18 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 18 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 19 | | 3 | 1 |
| 1449 | Handel Handel | bam_vgr_konten | 20 | 3 | 1 | | |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 20 | | 3 | 1 |
| 1449 | Handel Handel | bam_vgr_konten | 21 | 3 | 1 | | |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 21 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 22 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 23 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 23 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 24 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 25 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 26 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 38 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 42 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 47 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 49 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 50 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 57 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 61 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 64 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 75 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 76 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 79 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 90 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 91 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 92 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 93 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 94 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 95 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 96 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 97 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 101 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 101 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 103 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 104 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 105 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 106 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 107 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 108 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 109 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 110 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 111 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 112 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 113 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 114 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 115 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 116 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 117 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 118 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 119 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 120 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 121 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 122 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 122 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 123 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 123 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 124 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 124 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 125 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 126 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 127 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 127 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 128 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_sektor | 129 | | 3 | 1 |
| 91 | Unternehme | Unternehmen | bam_vgr_konten | 141 | | 3 | 1 |

```
        91      Unternehme      Unternehmen     bam_vgr_sektor 144             3       1
        91      Unternehme      Unternehmen     bam_vgr_sektor 150             3       1
        91      Unternehme      Unternehmen     bam_vgr_sektor 154             3       1
        312     Ausfuhr Ausfuhr bam_eurostat_themes_de 16              2       1
        312     Ausfuhr Ausfuhr bam_eurostat_themes_de 17              2       1
        312     Ausfuhr Ausfuhr bam_vgrneu_themen      23              2       1
        312     Ausfuhr Ausfuhr bam_vgr_sektor 25              2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 79              2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 80              2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 81              2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 82              2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 104             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 105             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 106             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 107             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 130             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 131             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 132             2       1
        312     Ausfuhr Ausfuhr bam_vgr_konten 133             2       1
        377     Binnenmarkt     Binnenmarkt     bam_eurostat_themes_de 5               1       1
        377     Binnenmarkt     Binnenmarkt     bam_eurostat_themes_de 9               1       1
        377     Binnenmarkt     Binnenmarkt     bam_eurostat_themes_de 27              1       1
        377     Binnenmarkt     Binnenmarkt     bam_eurostat_themes_de 28              1       1

351 rows displayed.
Select
                    *
                      from session.votetmp
                    order by eintrag
EINTRAG TREFFER_IM_TEXT
Content of session.votetmp: all keywords found in querying text
--------
        EINTRAG TREFFER_IM_TEXT
        Abbildung       3
        Ausfuhr 2
        Austausch       1
        Außenhandel     1
        Bild    1
        Binnenmarkt     1
        Deutschland     4
        Die     1
        Durchschnitt    1
        EU      1
        EWU     11
        Entwicklung     1
        Erklõrung       1
        Erwartung       1
        Etablierung     1
        Exporte 1
        Exportpreise    1
        Extrahandel     1
        F³nftel 1
        Gewicht 1
        Handel  3
        Handelspartn    1
        Handelsverflechtung     1
        Hõlfte  1
        Impulse 1
        Intra   1
        Intrahandel     3
        Jahr    1
        Land    2
        Lohnst³ckkoste  1
        Lõnd    9
        Mõrkte  1
        Partnerlõnd     1
        Rechnung        2
        Tempo   1
        Tochterunternehme       1
        Unternehme      3
        Verflechtung    1
        Welthandel      1
        Wert    1
        Wettbewerbsfõhigkeit    1
        Zeit    3
        ander   1
        ber³cksichtig   1
        daf³r   1
        dass    5
        deutsche        4
        d³rfte  1
        enge    1
```

```
        erfolgreich    1
        erf³llen       1
        f³hren  1
        f³r     2
        inlõndisch     1
        innerhalb      1
        krõftig 1
        k÷nnte  1
        nominal 2
        osteuropõisch  2
        real    1
        schwõch 2
        seinerzeit     1
        stõrk   1
        weltwirtschaftlich     1
        zunõchsen      1
        zur³ckzuf³hre  1
        ³berraschennd  1
        ³brig   4
        õhnlich 1

69 rows displayed.
Select
                      kategorie, satznummer, eintrag
                        from session.result
                        group by kategorie, satznummer, eintrag
                      order by kategorie, satznummer desc(get_dbdata) $cursor->{NAME}:
KATEGORIE SATZNUMMER EINTRAG
Vote categories grouped by categorie/Satznummer
--------
Sql-Statement:
Select
                      kategorie, satznummer, eintrag
                        from session.result
                        group by kategorie, satznummer, eintrag
                      order by kategorie, satznummer desc
        KATEGORIE       SATZNUMMER      EINTRAG
        bam_brand_aregion       15      Deutschland
        bam_brand_aregion       14      Deutschland
        bam_brand_aregion       13      Deutschland
        bam_brand_aregion       12      Deutschland
        bam_brand_aregion       11      Deutschland
        bam_brand_aregion       10      Deutschland
        bam_brand_aregion       9       Deutschland
        bam_brand_aregion       8       Deutschland
        bam_brand_aregion       7       Deutschland
        bam_brand_aregion       6       Deutschland
        bam_brand_aregion       5       Deutschland
        bam_brand_aregion       4       Deutschland
        bam_brand_aregion       3       Deutschland
        bam_brand_aregion       2       Deutschland
        bam_brand_aregion       1       Deutschland
        bam_brand_bip   18      Unternehmen
        bam_brand_bip   17      Unternehmen
        bam_brand_bip   15      Unternehmen
        bam_brand_bip   13      Unternehmen
        bam_brand_bip   7       Handel
        bam_brand_bip   6       Handel
        bam_brand_bregion       15      Deutschland
        bam_brand_bregion       14      Deutschland
        bam_brand_bregion       13      Deutschland
        bam_brand_bregion       12      Deutschland
        bam_brand_bregion       11      Deutschland
        bam_brand_bregion       10      Deutschland
        bam_brand_bregion       9       Deutschland
        bam_brand_bregion       8       Deutschland
        bam_brand_bregion       7       Deutschland
        bam_brand_bregion       6       Deutschland
        bam_brand_bregion       5       Deutschland
        bam_brand_bregion       4       Deutschland
        bam_brand_bregion       3       Deutschland
        bam_brand_bregion       2       Deutschland
        bam_brand_bregion       1       Deutschland
        bam_brand_eregion       15      Deutschland
        bam_brand_eregion       14      Deutschland
        bam_brand_eregion       13      Deutschland
        bam_brand_eregion       12      Deutschland
        bam_brand_eregion       11      Deutschland
        bam_brand_eregion       10      Deutschland
        bam_brand_eregion       9       Deutschland
        bam_brand_eregion       8       Deutschland
        bam_brand_eregion       7       Deutschland
```

```
bam_brand_eregion      6       Deutschland
bam_brand_eregion      5       Deutschland
bam_brand_eregion      4       Deutschland
bam_brand_eregion      3       Deutschland
bam_brand_eregion      2       Deutschland
bam_brand_eregion      1       Deutschland
bam_brand_erwt 13      Unternehmen
bam_brand_erwt 7       Handel
bam_brand_erwt 6       Handel
bam_brand_iregion      15      Deutschland
bam_brand_iregion      14      Deutschland
bam_brand_iregion      13      Deutschland
bam_brand_iregion      12      Deutschland
bam_brand_iregion      11      Deutschland
bam_brand_iregion      10      Deutschland
bam_brand_iregion      9       Deutschland
bam_brand_iregion      8       Deutschland
bam_brand_iregion      7       Deutschland
bam_brand_iregion      6       Deutschland
bam_brand_iregion      5       Deutschland
bam_brand_iregion      4       Deutschland
bam_brand_iregion      3       Deutschland
bam_brand_iregion      2       Deutschland
bam_brand_iregion      1       Deutschland
bam_brand_nregion      15      Deutschland
bam_brand_nregion      14      Deutschland
bam_brand_nregion      13      Deutschland
bam_brand_nregion      12      Deutschland
bam_brand_nregion      11      Deutschland
bam_brand_nregion      10      Deutschland
bam_brand_nregion      9       Deutschland
bam_brand_nregion      8       Deutschland
bam_brand_nregion      7       Deutschland
bam_brand_nregion      6       Deutschland
bam_brand_nregion      5       Deutschland
bam_brand_nregion      4       Deutschland
bam_brand_nregion      3       Deutschland
bam_brand_nregion      2       Deutschland
bam_brand_nregion      1       Deutschland
bam_eurostat_country_de        6       Deutschland
bam_eurostat_themes_de 28      Binnenmarkt
bam_eurostat_themes_de 27      Binnenmarkt
bam_eurostat_themes_de 17      Ausfuhr
bam_eurostat_themes_de 16      Ausfuhr
bam_eurostat_themes_de 9       Binnenmarkt
bam_eurostat_themes_de 5       Binnenmarkt
bam_ifo_kbran  172     Deutschland
bam_ifo_kbran  171     Deutschland
bam_ifo_kbran  170     Deutschland
bam_ifo_kbran  169     Deutschland
bam_ifo_kbran  168     Deutschland
bam_ifo_kbran  167     Deutschland
bam_ifo_kbran  166     Deutschland
bam_ifo_kbran  165     Deutschland
bam_ifo_kbran  164     Deutschland
bam_ifo_kbran  163     Deutschland
bam_ifo_kbran  162     Deutschland
bam_ifo_kbran  161     Deutschland
bam_ifo_kbran  160     Deutschland
bam_ifo_kbran  159     Deutschland
bam_ifo_kbran  158     Deutschland
bam_ifo_kbran  157     Deutschland
bam_ifo_kbran  156     Deutschland
bam_ifo_kbran  155     Deutschland
bam_ifo_kbran  154     Deutschland
bam_ifo_kbran  153     Deutschland
bam_ifo_kbran  152     Deutschland
bam_ifo_kbran  151     Deutschland
bam_ifo_kbran  150     Deutschland
bam_ifo_kbran  149     Deutschland
bam_ifo_kbran  148     Deutschland
bam_ifo_kbran  147     Deutschland
bam_ifo_kbran  146     Deutschland
bam_ifo_kbran  145     Deutschland
bam_ifo_kbran  144     Deutschland
bam_ifo_kbran  143     Deutschland
bam_ifo_kbran  142     Deutschland
bam_ifo_kbran  141     Deutschland
bam_ifo_kbran  140     Deutschland
bam_ifo_kbran  139     Deutschland
bam_ifo_kbran  138     Deutschland
bam_ifo_kbran  137     Deutschland
```

```
bam_ifo_kbran  136     Deutschland
bam_ifo_kbran  135     Deutschland
bam_ifo_kbran  134     Deutschland
bam_ifo_kbran  133     Deutschland
bam_ifo_kbran  132     Deutschland
bam_ifo_kbran  131     Deutschland
bam_ifo_kbran  130     Deutschland
bam_ifo_kbran  129     Deutschland
bam_ifo_kbran  128     Deutschland
bam_ifo_kbran  127     Deutschland
bam_ifo_kbran  126     Deutschland
bam_ifo_kbran  125     Deutschland
bam_ifo_kbran  124     Deutschland
bam_ifo_kbran  123     Deutschland
bam_ifo_kbran  122     Deutschland
bam_ifo_kbran  121     Deutschland
bam_ifo_kbran  120     Deutschland
bam_ifo_kbran  119     Deutschland
bam_ifo_kbran  118     Deutschland
bam_ifo_kbran  117     Deutschland
bam_ifo_kbran  116     Deutschland
bam_ifo_kbran  115     Deutschland
bam_ifo_kbran  114     Deutschland
bam_ifo_kbran  113     Deutschland
bam_ifo_kbran  112     Deutschland
bam_ifo_kbran  111     Deutschland
bam_ifo_kbran  110     Deutschland
bam_ifo_kbran  108     Deutschland
bam_ifo_kbran  107     Deutschland
bam_ifo_kbran  106     Deutschland
bam_ifo_kbran  105     Deutschland
bam_ifo_kbran  104     Deutschland
bam_ifo_kbran  103     Deutschland
bam_ifo_kbran  102     Deutschland
bam_ifo_kbran  101     Deutschland
bam_ifo_kbran  100     Deutschland
bam_ifo_kbran  99      Deutschland
bam_ifo_kbran  98      Deutschland
bam_ifo_kbran  97      Deutschland
bam_ifo_kbran  96      Deutschland
bam_ifo_kbran  95      Deutschland
bam_ifo_kbran  94      Deutschland
bam_ifo_kbran  93      Deutschland
bam_ifo_kbran  92      Deutschland
bam_ifo_kbran  91      Deutschland
bam_ifo_kbran  90      Deutschland
bam_ifo_kbran  89      Deutschland
bam_ifo_kbran  88      Deutschland
bam_ifo_kbran  87      Deutschland
bam_ifo_kbran  86      Deutschland
bam_ifo_kbran  85      Deutschland
bam_ifo_kbran  84      Deutschland
bam_ifo_kbran  83      Deutschland
bam_ifo_kbran  82      Deutschland
bam_ifo_kbran  81      Deutschland
bam_ifo_kbran  80      Deutschland
bam_ifo_kbran  79      Deutschland
bam_ifo_kbran  78      Deutschland
bam_ifo_kbran  76      Deutschland
bam_ifo_kbran  75      Deutschland
bam_ifo_kbran  74      Deutschland
bam_ifo_kbran  73      Deutschland
bam_ifo_kbran  72      Deutschland
bam_ifo_kbran  71      Deutschland
bam_ifo_kbran  70      Deutschland
bam_ifo_kbran  69      Deutschland
bam_ifo_kbran  68      Deutschland
bam_ifo_kbran  67      Deutschland
bam_ifo_kbran  66      Deutschland
bam_ifo_kbran  65      Deutschland
bam_ifo_kbran  64      Deutschland
bam_ifo_kbran  63      Deutschland
bam_ifo_kbran  62      Deutschland
bam_ifo_kbran  61      Deutschland
bam_ifo_kbran  60      Deutschland
bam_ifo_kbran  59      Deutschland
bam_ifo_kbran  58      Deutschland
bam_ifo_kbran  57      Deutschland
bam_ifo_kbran  56      Deutschland
bam_ifo_kbran  55      Deutschland
bam_ifo_kbran  54      Deutschland
bam_ifo_kbran  53      Deutschland
```

```
bam_ifo_kbran  52      Deutschland
bam_ifo_kbran  51      Deutschland
bam_ifo_kbran  50      Deutschland
bam_ifo_kbran  49      Deutschland
bam_ifo_kbran  48      Deutschland
bam_ifo_kbran  47      Deutschland
bam_ifo_kbran  46      Deutschland
bam_ifo_kbran  45      Deutschland
bam_ifo_kbran  44      Deutschland
bam_ifo_kbran  43      Deutschland
bam_ifo_kbran  42      Deutschland
bam_ifo_kbran  41      Deutschland
bam_ifo_kbran  40      Deutschland
bam_ifo_kbran  39      Deutschland
bam_ifo_kbran  38      Deutschland
bam_ifo_kbran  37      Deutschland
bam_ifo_kbran  36      Deutschland
bam_ifo_kbran  35      Deutschland
bam_ifo_kbran  34      Deutschland
bam_ifo_kbran  33      Deutschland
bam_ifo_kbran  32      Deutschland
bam_ifo_kbran  31      Deutschland
bam_ifo_kbran  30      Deutschland
bam_ifo_kbran  29      Deutschland
bam_ifo_kbran  28      Deutschland
bam_ifo_kbran  27      Deutschland
bam_ifo_kbran  26      Deutschland
bam_ifo_kbran  25      Deutschland
bam_ifo_kbran  24      Deutschland
bam_ifo_kbran  23      Deutschland
bam_ifo_kbran  22      Deutschland
bam_ifo_kbran  21      Deutschland
bam_ifo_kbran  20      Deutschland
bam_ifo_kbran  19      Deutschland
bam_ifo_kbran  18      Deutschland
bam_ifo_kbran  17      Deutschland
bam_ifo_kbran  16      Deutschland
bam_ifo_kbran  15      Deutschland
bam_ifo_kbran  14      Deutschland
bam_ifo_kbran  13      Deutschland
bam_ifo_kbran  12      Deutschland
bam_ifo_kbran  11      Deutschland
bam_ifo_kbran  10      Deutschland
bam_ifo_kbran  9       Deutschland
bam_ifo_kbran  8       Deutschland
bam_ifo_kbran  7       Deutschland
bam_ifo_kbran  6       Deutschland
bam_ifo_kbran  5       Deutschland
bam_ifo_kbran  4       Deutschland
bam_ifo_kbran  3       Deutschland
bam_ifo_kbran  2       Deutschland
bam_ifo_kbran  1       Deutschland
bam_vgr_konten 141     Unternehmen
bam_vgr_konten 133     Ausfuhr
bam_vgr_konten 132     Ausfuhr
bam_vgr_konten 131     Ausfuhr
bam_vgr_konten 130     Ausfuhr
bam_vgr_konten 127     Unternehmen
bam_vgr_konten 124     Unternehmen
bam_vgr_konten 123     Unternehmen
bam_vgr_konten 122     Unternehmen
bam_vgr_konten 107     Ausfuhr
bam_vgr_konten 106     Ausfuhr
bam_vgr_konten 105     Ausfuhr
bam_vgr_konten 104     Ausfuhr
bam_vgr_konten 101     Unternehmen
bam_vgr_konten 82      Ausfuhr
bam_vgr_konten 81      Ausfuhr
bam_vgr_konten 80      Ausfuhr
bam_vgr_konten 79      Ausfuhr
bam_vgr_konten 76      Unternehmen
bam_vgr_konten 61      Unternehmen
bam_vgr_konten 50      Unternehmen
bam_vgr_konten 49      Unternehmen
bam_vgr_konten 47      Unternehmen
bam_vgr_konten 26      Unternehmen
bam_vgr_konten 25      Unternehmen
bam_vgr_konten 24      Unternehmen
bam_vgr_konten 23      Unternehmen
bam_vgr_konten 22      Unternehmen
bam_vgr_konten 21      Handel
bam_vgr_konten 21      Unternehmen
```

```
        bam_vgr_konten 20      Handel
        bam_vgr_konten 20      Unternehmen
        bam_vgr_konten 19      Unternehmen
        bam_vgr_konten 18      Unternehmen
        bam_vgr_konten 17      Unternehmen
        bam_vgr_konten 16      Unternehmen
        bam_vgr_konten 15      Unternehmen
        bam_vgr_konten 14      Unternehmen
        bam_vgr_region 1002    Deutschland
        bam_vgr_region 1001    Deutschland
        bam_vgr_region 9       Deutschland
        bam_vgr_sektor 154     Unternehmen
        bam_vgr_sektor 150     Unternehmen
        bam_vgr_sektor 144     Unternehmen
        bam_vgr_sektor 129     Unternehmen
        bam_vgr_sektor 128     Unternehmen
        bam_vgr_sektor 127     Unternehmen
        bam_vgr_sektor 126     Unternehmen
        bam_vgr_sektor 125     Unternehmen
        bam_vgr_sektor 124     Unternehmen
        bam_vgr_sektor 123     Unternehmen
        bam_vgr_sektor 122     Unternehmen
        bam_vgr_sektor 121     Unternehmen
        bam_vgr_sektor 120     Unternehmen
        bam_vgr_sektor 119     Unternehmen
        bam_vgr_sektor 118     Unternehmen
        bam_vgr_sektor 117     Unternehmen
        bam_vgr_sektor 116     Unternehmen
        bam_vgr_sektor 115     Unternehmen
        bam_vgr_sektor 114     Unternehmen
        bam_vgr_sektor 113     Unternehmen
        bam_vgr_sektor 112     Unternehmen
        bam_vgr_sektor 111     Unternehmen
        bam_vgr_sektor 110     Unternehmen
        bam_vgr_sektor 109     Unternehmen
        bam_vgr_sektor 108     Unternehmen
        bam_vgr_sektor 107     Unternehmen
        bam_vgr_sektor 106     Unternehmen
        bam_vgr_sektor 105     Unternehmen
        bam_vgr_sektor 104     Unternehmen
        bam_vgr_sektor 103     Unternehmen
        bam_vgr_sektor 101     Unternehmen
        bam_vgr_sektor 97      Unternehmen
        bam_vgr_sektor 96      Unternehmen
        bam_vgr_sektor 95      Unternehmen
        bam_vgr_sektor 94      Unternehmen
        bam_vgr_sektor 93      Unternehmen
        bam_vgr_sektor 92      Unternehmen
        bam_vgr_sektor 91      Unternehmen
        bam_vgr_sektor 90      Unternehmen
        bam_vgr_sektor 79      Unternehmen
        bam_vgr_sektor 75      Unternehmen
        bam_vgr_sektor 64      Unternehmen
        bam_vgr_sektor 57      Unternehmen
        bam_vgr_sektor 42      Unternehmen
        bam_vgr_sektor 38      Unternehmen
        bam_vgr_sektor 25      Ausfuhr
        bam_vgr_sektor 23      Unternehmen
        bam_vgr_sektor 17      Unternehmen
        bam_vgrneu_themen      23      Ausfuhr

351 rows displayed.
Select
                     trim(kategorie) as tkategorie, count(*) as treffer, eintrag
                       from session.result
                     group by kategorie, eintrag
                      order by tkategorie
TKATEGORIE TREFFER EINTRAG
Number of hits for each keyword in each category:
--------
        TKATEGORIE     TREFFER EINTRAG
        bam_brand_aregion      15      Deutschland
        bam_brand_bip  4       Unternehmen
        bam_brand_bip  2       Handel
        bam_brand_bregion      15      Deutschland
        bam_brand_eregion      15      Deutschland
        bam_brand_erwt 1       Unternehmen
        bam_brand_erwt 2       Handel
        bam_brand_iregion      15      Deutschland
        bam_brand_nregion      15      Deutschland
        bam_eurostat_country_de        1       Deutschland
        bam_eurostat_themes_de 4       Binnenmarkt
```

```
        bam_eurostat_themes_de 2       Ausfuhr
        bam_ifo_kbran  170     Deutschland
        bam_vgr_konten 2       Handel
        bam_vgr_konten 12      Ausfuhr
        bam_vgr_konten 24      Unternehmen
        bam_vgr_region 3       Deutschland
        bam_vgr_sektor 47      Unternehmen
        bam_vgr_sektor 1       Ausfuhr
        bam_vgrneu_themen      1       Ausfuhr

20 rows displayed.
Select
                     trim(kategorie) as tkategorie, count(*) as treffer, logbestand
                       from session.result
                     group by kategorie, logbestand
                      order by tkategorie, treffer desc
TKATEGORIE TREFFER LOGBESTAND
Simple Vote categories, hits all together
--------
        TKATEGORIE     TREFFER LOGBESTAND
        bam_brand_aregion      15
countsnrcat for bam_brand_aregion: 12
        bam_brand_bip  6
countsnrcat for bam_brand_bip: 17
        bam_brand_bregion      15
countsnrcat for bam_brand_bregion: 12
        bam_brand_eregion      15
countsnrcat for bam_brand_eregion: 16
        bam_brand_erwt 3
countsnrcat for bam_brand_erwt: 16
        bam_brand_iregion      15
countsnrcat for bam_brand_iregion: 16
        bam_brand_nregion      15
countsnrcat for bam_brand_nregion: 15
        bam_eurostat_country_de        1
countsnrcat for bam_eurostat_country_de: 23
        bam_eurostat_themes_de 6
countsnrcat for bam_eurostat_themes_de: 30
        bam_ifo_kbran  170
countsnrcat for bam_ifo_kbran: 170
        bam_vgr_konten 38
countsnrcat for bam_vgr_konten: 158
        bam_vgr_region 3
countsnrcat for bam_vgr_region: 3
        bam_vgr_sektor 48
countsnrcat for bam_vgr_sektor: 154
        bam_vgrneu_themen      1
countsnrcat for bam_vgrneu_themen: 27

14 rows displayed.
Select
                    eintrag, trim(logbestandsname) as log_bestand, trim(kategorie) as
tkategorie,count(*) as treffer
                     from session.result, lnk_met_log_bestand
                   where tabellenname = kategorie
                    group by eintrag, logbestandsname, kategorie
EINTRAG LOG_BESTAND TKATEGORIE TREFFER
Voted words and belonging logical domains/ categories
--------
        EINTRAG LOG_BESTAND    TKATEGORIE     TREFFER
        Ausfuhr DIW-vgr_Konten bam_vgr_konten 12
No. of hits in total for Ausfuhr: 12
        Ausfuhr DIW-vgr_Sektoren       bam_vgr_sektor 1
No. of hits in total for Ausfuhr: 13
        Ausfuhr DIW-vgrneu     bam_vgrneu_themen      1
No. of hits in total for Ausfuhr: 14
        Ausfuhr Eurostat-Linda bam_eurostat_themes_de 2
No. of hits in total for Ausfuhr: 16
        Binnenmarkt    Eurostat-Linda bam_eurostat_themes_de 4
No. of hits in total for Binnenmarkt: 4
        Deutschland    DIW-vgr_Konten bam_vgr_region 3
No. of hits in total for Deutschland: 3
        Deutschland    DIW-vgr_Sektoren       bam_vgr_region 3
No. of hits in total for Deutschland: 6
        Deutschland    Eurostat-Linda bam_eurostat_country_de       1
No. of hits in total for Deutschland: 7
        Deutschland    WIDAB_Brandenburg_Arbeitsmarkt       bam_brand_aregion      15
No. of hits in total for Deutschland: 22
        Deutschland    WIDAB_Brandenburg_BIP  bam_brand_iregion      15
No. of hits in total for Deutschland: 37
        Deutschland    WIDAB_Brandenburg_Bau  bam_brand_bregion      15
No. of hits in total for Deutschland: 52
```

```
        Deutschland    WIDAB_Brandenburg_Erwerb      bam_brand_eregion      15
No. of hits in total for Deutschland: 67
        Deutschland    WIDAB_Brandenburg_Industrie   bam_brand_nregion      15
No. of hits in total for Deutschland: 82
        Deutschland    ifo-KT  bam_ifo_kbran  170
No. of hits in total for Deutschland: 252
        Handel  DIW-vgr_Konten bam_vgr_konten 2
No. of hits in total for Handel: 2
        Handel  WIDAB_Brandenburg_BIP  bam_brand_bip  2
No. of hits in total for Handel: 4
        Handel  WIDAB_Brandenburg_Erwerb      bam_brand_erwt 2
No. of hits in total for Handel: 6
        Unternehmen    DIW-vgr_Konten bam_vgr_konten 24
No. of hits in total for Unternehmen: 24
        Unternehmen    DIW-vgr_Sektoren       bam_vgr_sektor 47
No. of hits in total for Unternehmen: 71
        Unternehmen    WIDAB_Brandenburg_BIP  bam_brand_bip  4
No. of hits in total for Unternehmen: 75
        Unternehmen    WIDAB_Brandenburg_Erwerb      bam_brand_erwt 1
No. of hits in total for Unternehmen: 76

21 rows displayed.
Select
                    trim(logbestand) as tbestand, count(*) as treffer
                     from session.result
                   where treffer_im_text > 0
                    group by logbestand
                     order by tbestand desc
TBESTAND TREFFER
Simple Vote domains
--------
        TBESTAND       TREFFER
                351

1 rows displayed.
Select
                    trim(logbestandsname) as log_bestand, count(*) as treffer
                     from session.result, lnk_met_log_bestand
                   where tabellenname = kategorie
                    group by logbestandsname
LOG_BESTAND TREFFER
Simple Vote logical domains
--------
        LOG_BESTAND    TREFFER
        DIW-vgr_Konten 41
        DIW-vgr_Sektoren       51
        DIW-vgrneu     1
        Eurostat-Linda 7
        WIDAB_Brandenburg_Arbeitsmarkt        15
        WIDAB_Brandenburg_BIP  21
        WIDAB_Brandenburg_Bau  15
        WIDAB_Brandenburg_Erwerb      18
        WIDAB_Brandenburg_Industrie   15
        ifo-KT  170

10 rows displayed.
Weight for Handel (Handel) = 3 * (log10(191) - log10(6) + 1) = 7.50864635059225
Weight for Ausfuhr (Ausfuhr) = 4 * (log10(369) - log10(16) + 1) = 9.45162553401254
Weight for Unternehmen (Unternehme) = 4 * (log10(345) - log10(76) + 1) = 6.62802201116993
Totalweightlist_element1: 06.62802201116993_*_Unternehme
Totalweightlist_element2: 0
Totalweightlist_element3: 0
Totalweightlist_element1: 07.50864635059225_*_Handel
Totalweightlist_element2: 06.62802201116993_*_Unternehme
Totalweightlist_element3: 0
Totalweightlist_element1: 09.45162553401254_*_Ausfuhr
Totalweightlist_element2: 07.50864635059225_*_Handel
Totalweightlist_element3: 06.62802201116993_*_Unternehme
Totalweight: 23.5882938957747 for 3 items gives an average of 7.86276463192491.
Totalweightlimit: 6.62802201116993.
$ptitem{ Ausfuhr }{order}= 1

<?xml version="1.0" encoding="ISO-8859-1"?>

<irdoc>
 <irheader>
   <subdomainlist id="1">
    <subdom id="1">
     <name>DIW-vgr_Konten</name>
    </subdom>
    <subdom id="2">
     <name>DIW-vgr_Sektoren</name>
```

```
 </subdom>
 <subdom id="3">
  <name>DIW-vgrneu</name>
 </subdom>
 <subdom id="4">
  <name>Eurostat-Linda</name>
 </subdom>
 <subdom id="5">
  <name>WIDAB_Brandenburg_Arbeitsmarkt</name>
 </subdom>
 <subdom id="6">
  <name>WIDAB_Brandenburg_BIP</name>
 </subdom>
 <subdom id="7">
  <name>WIDAB_Brandenburg_Bau</name>
 </subdom>
 <subdom id="8">
  <name>WIDAB_Brandenburg_Erwerb</name>
 </subdom>
 <subdom id="9">
  <name>WIDAB_Brandenburg_Industrie</name>
 </subdom>
 <subdom id="10">
  <name>ifo-KT</name>
 </subdom>
</subdomainlist>
<catlist id="1">
 <cat id="1">
  <name>bam_brand_aregion</name>
 </cat>
 <cat id="2">
  <name>bam_brand_bip</name>
 </cat>
 <cat id="3">
  <name>bam_brand_bregion</name>
 </cat>
 <cat id="4">
  <name>bam_brand_eregion</name>
 </cat>
 <cat id="5">
  <name>bam_brand_erwt</name>
 </cat>
 <cat id="6">
  <name>bam_brand_iregion</name>
 </cat>
 <cat id="7">
  <name>bam_brand_nregion</name>
 </cat>
 <cat id="8">
  <name>bam_eurostat_country_de</name>
 </cat>
 <cat id="9">
  <name>bam_eurostat_themes_de</name>
 </cat>
 <cat id="10">
  <name>bam_ifo_kbran</name>
 </cat>
 <cat id="11">
  <name>bam_vgr_konten</name>
 </cat>
 <cat id="12">
  <name>bam_vgr_region</name>
 </cat>
 <cat id="13">
  <name>bam_vgr_sektor</name>
 </cat>
 <cat id="14">
  <name>bam_vgrneu_themen</name>
 </cat>
</catlist>
<termlist id="1">
 <term id="1449" subdom="6" cat="2" key="6" order="2">
   <name>Handel</name>
 </term>
 <term id="1449" subdom="8" cat="5" key="6" order="2">
   <name>Handel</name>
 </term>
 <term id="1449" subdom="6" cat="2" key="7" order="2">
   <name>Handel</name>
 </term>
 <term id="1449" subdom="8" cat="5" key="7" order="2">
   <name>Handel</name>
```

```
</term>
<term id="1449" subdom="1" cat="11" key="20" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="1" cat="11" key="21" order="2">
  <name>Handel</name>
</term>
<term id="312" subdom="4" cat="9" key="16" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="4" cat="9" key="17" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="3" cat="14" key="23" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="2" cat="13" key="25" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="79" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="80" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="81" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="82" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="104" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="105" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="106" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="107" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="130" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="131" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="132" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="133" order="1">
  <name>Ausfuhr</name>
</term>
<term id="1449" subdom="6" cat="2" key="6" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="8" cat="5" key="6" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="6" cat="2" key="7" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="8" cat="5" key="7" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="1" cat="11" key="20" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="1" cat="11" key="21" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="6" cat="2" key="6" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="8" cat="5" key="6" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="6" cat="2" key="7" order="2">
  <name>Handel</name>
</term>
```

```
<term id="1449" subdom="8" cat="5" key="7" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="1" cat="11" key="20" order="2">
  <name>Handel</name>
</term>
<term id="1449" subdom="1" cat="11" key="21" order="2">
  <name>Handel</name>
</term>
<term id="91" subdom="6" cat="2" key="13" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="8" cat="5" key="13" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="14" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="15" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="15" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="16" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="18" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="18" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="19" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="20" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="21" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="22" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="23" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="23" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="24" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="25" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="26" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="38" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="42" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="47" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="49" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="50" order="3">
```

```
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="57" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="61" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="64" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="75" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="76" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="79" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="90" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="91" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="92" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="93" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="94" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="95" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="96" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="97" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="101" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="101" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="103" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="104" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="105" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="106" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="107" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="108" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="109" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="110" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="111" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="112" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="113" order="3">
  <name>Unternehmen</name>
```

```
</term>
<term id="91" subdom="2" cat="13" key="114" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="115" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="116" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="117" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="118" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="119" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="120" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="121" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="122" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="122" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="123" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="123" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="124" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="124" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="125" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="126" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="127" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="127" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="128" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="129" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="141" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="144" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="150" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="154" order="3">
  <name>Unternehmen</name>
</term>
<term id="312" subdom="4" cat="9" key="16" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="4" cat="9" key="17" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="3" cat="14" key="23" order="1">
  <name>Ausfuhr</name>
</term>
```

```
<term id="312" subdom="2" cat="13" key="25" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="79" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="80" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="81" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="82" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="104" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="105" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="106" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="107" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="130" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="131" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="132" order="1">
  <name>Ausfuhr</name>
</term>
<term id="312" subdom="1" cat="11" key="133" order="1">
  <name>Ausfuhr</name>
</term>
<term id="91" subdom="6" cat="2" key="13" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="8" cat="5" key="13" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="14" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="15" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="15" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="16" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="18" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="18" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="19" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="20" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="21" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="22" order="3">
```

```
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="23" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="23" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="24" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="25" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="26" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="38" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="42" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="47" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="49" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="50" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="57" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="61" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="64" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="75" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="76" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="79" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="90" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="91" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="92" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="93" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="94" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="95" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="96" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="97" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="101" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="101" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="103" order="3">
  <name>Unternehmen</name>
```

```
</term>
<term id="91" subdom="2" cat="13" key="104" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="105" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="106" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="107" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="108" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="109" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="110" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="111" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="112" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="113" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="114" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="115" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="116" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="117" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="118" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="119" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="120" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="121" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="122" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="122" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="123" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="123" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="124" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="124" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="125" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="126" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="127" order="3">
  <name>Unternehmen</name>
</term>
```

```
<term id="91" subdom="2" cat="13" key="127" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="128" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="129" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="141" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="144" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="150" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="154" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="13" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="8" cat="5" key="13" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="14" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="15" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="15" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="16" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="17" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="6" cat="2" key="18" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="18" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="19" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="20" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="21" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="22" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="23" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="23" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="24" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="25" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="26" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="38" order="3">
```

```
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="42" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="47" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="49" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="50" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="57" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="61" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="64" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="75" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="76" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="79" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="90" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="91" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="92" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="93" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="94" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="95" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="96" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="97" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="101" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="101" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="103" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="104" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="105" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="106" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="107" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="108" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="109" order="3">
  <name>Unternehmen</name>
```

```
</term>
<term id="91" subdom="2" cat="13" key="110" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="111" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="112" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="113" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="114" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="115" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="116" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="117" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="118" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="119" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="120" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="121" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="122" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="122" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="123" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="123" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="124" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="124" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="125" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="126" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="127" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="127" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="128" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="129" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="1" cat="11" key="141" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="144" order="3">
  <name>Unternehmen</name>
</term>
<term id="91" subdom="2" cat="13" key="150" order="3">
  <name>Unternehmen</name>
</term>
```

```
    <term id="91" subdom="2" cat="13" key="154" order="3">
      <name>Unternehmen</name>
    </term>
   </termlist>
  </irheader>
  <taggedtext>


Von der Etablierung des Binnenmarktes in der EU wurden seinerzeit
                krõftige Impulse f³r den <term id="1449">Handel</term> zwischen den
Partnerlõndern
                erwartet. Diese Erwartung hat sich weitgehend erf³llt. So ist die
                nominale <term id="312">Ausfuhr</term> Deutschlands in die ³brigen EWU-
Lõnder von 1993
                bis 2000 um 73 % gestiegen. Zwar hat der Außenhandel der EWU in
                dieser Zeit in õhnlichem Tempo wie der Intrahandel expandiert
                (Abbildung 1), was daf³r spricht, dass der Austausch innerhalb der
                EWU nicht stõrker zugenommen hat als die allgemeine
                weltwirtschaftliche Verflechtung. Dabei ist aber zu ber³cksichtigen,
                dass sich viele osteuropõische Lõnder erst in letzter Zeit in den
                Welthandel integriert haben und sich der <term id="1449">Handel</term> mit
ihnen daher
                besonders dynamisch entwickelte.

                Deutschland ist mit einem Gewicht von reichlich einem F³nftel der
                wichtigste Handelspartner im Intra-EWU-<term id="1449">Handel</term>. [1]
Die Exporte
                Deutschlands in die ³brigen EWU-Lõnder sind in nominaler Rechnung
                in der zweiten Hõlfte der 90er Jahre weniger stark gestiegen als der
                Intrahandel der anderen EWU- Lõnder (Abbildung 2). Diese
                Entwicklung ist zunõchst ³berraschend, denn in dieser Zeit hatte sich
                die Wettbewerbsfõhigkeit der deutschen <term id="91">Unternehmen</term>
infolge deutlich
                niedrigerer Lohnst³ckkosten als beim Durchschnitt der ³brigen
                EWU-Lõnder erheblich verbessert (Abbildung 3). Dies kann dazu
                gef³hrt haben, dass die Exportpreise Deutschlands weniger stark
                gestiegen sind als bei den anderen EWU-Lõndern, so dass der Wert
                der <term id="312">Ausfuhr</term> entsprechend schwõcher zunahm. In realer
Rechnung
                k÷nnte sich ein anderes Bild ergeben. Eine weitere Erklõrung f³r den
                schwõcheren Intrahandel der inlõndischen deutschen <term
id="91">Unternehmen</term>
                kann darin gesehen werden, dass die Mõrkte in den einzelnen
                EWU-Lõndern von deutschen Tochterunternehmen im Land selbst
                oder von einem anderen EWU-Land aus beliefert wurden. Im
                Extrahandel waren die deutschen <term id="91">Unternehmen</term>
erfolgreicher als
                diejenigen der ³brigen EWU-Lõnder; dies d³rfte vor allem auf die
                besonders engen Handelsverflechtungen mit den osteuropõischen
                Lõndern zur³ckzuf³hren sein.
   </taggedtext>
</irdoc>
EOLABER
EOT
Goodbye from WIDAB Server
EOS
```

### *3.12. Der STATFINDER in Aktion*

Die Funktionsweise des DIW-STATFINDER (**F**lexible **IN**ternet **D**elivery of **E**conomic **R**elated **STAT**istics) soll am Beispiel des öffentlich zugänglichen Internet-Angebots „Regionalstatistische Datenbanken“ verdeutlicht werden. STATFINDER als http-Client funktioniert mit jedem Rechner, der einen neueren Webbrowser installiert und Zugang zum Internet hat.

Der STATFINDER ist layoutmäßig vollständig in den Internetauftritt des DIW Berlin integriert und profitiert daher zusätzlich von dem großen Umfang an Zugriffen, die das umfangreiche DIW Webangebot generiert. Ausgangspunkt ist die Internetadresse http://www.diw.de/deutsch/service/statfinder/regiostat/ ,
nach deren Aufruf folgende Einstiegsseite mit den im Rahmen dieses Projekts öffentlich verfügbar gemachten regionalstatistischen Datenbanken erscheint (Abbildung 40).

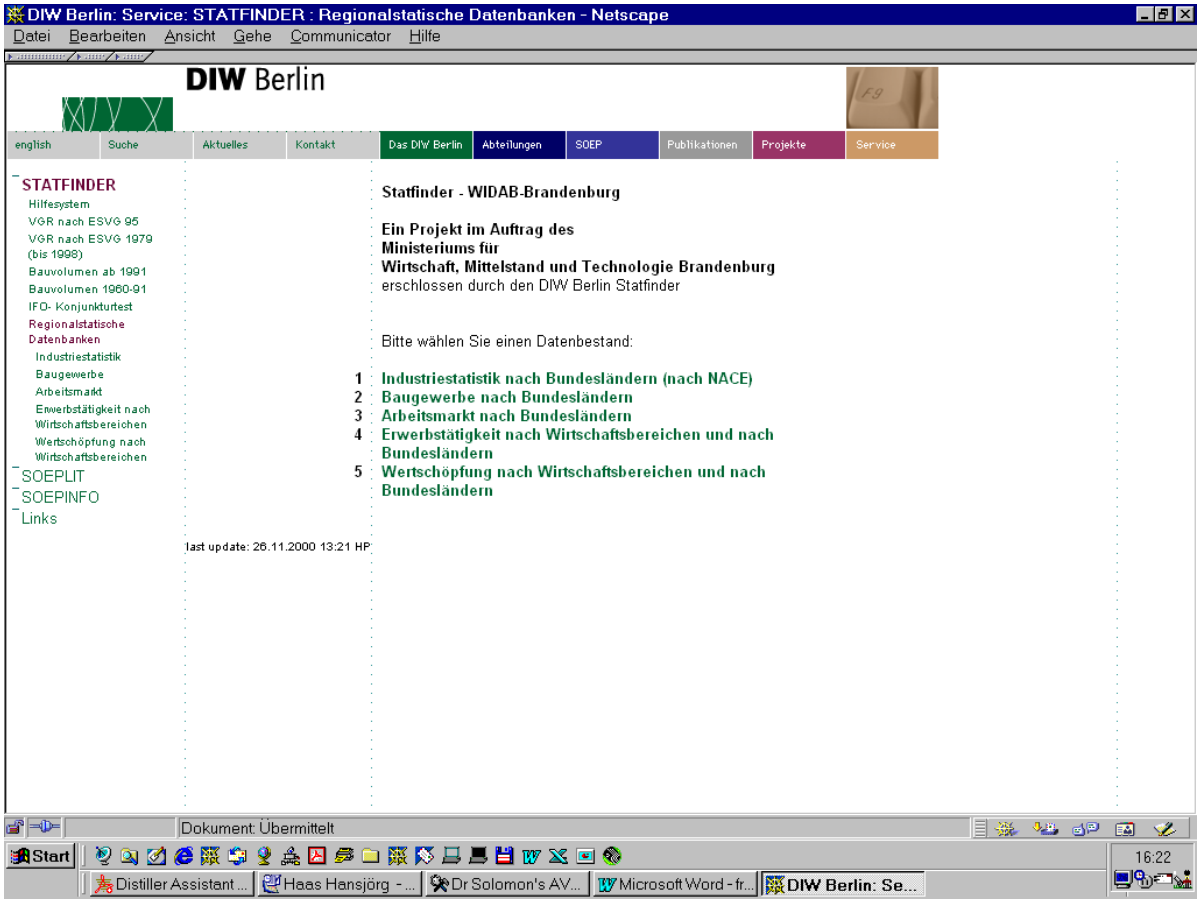



**Abbildung 40: STATFINDER in Aktion, Einstieg**

Besonderen Wert wurde auf einfache Suchmöglichkeiten, die sich weniger an den Datenbankstrukturen als an den Interessen des Nutzers orientieren, gelegt. Dies wird an folgendem Menü für die Industriestatistik illustriert, in dem drei Kategorien Regionen, Themen und Branchen angeboten werden, aus denen eines oder mehrere Merkmale ausgewählt werden können.

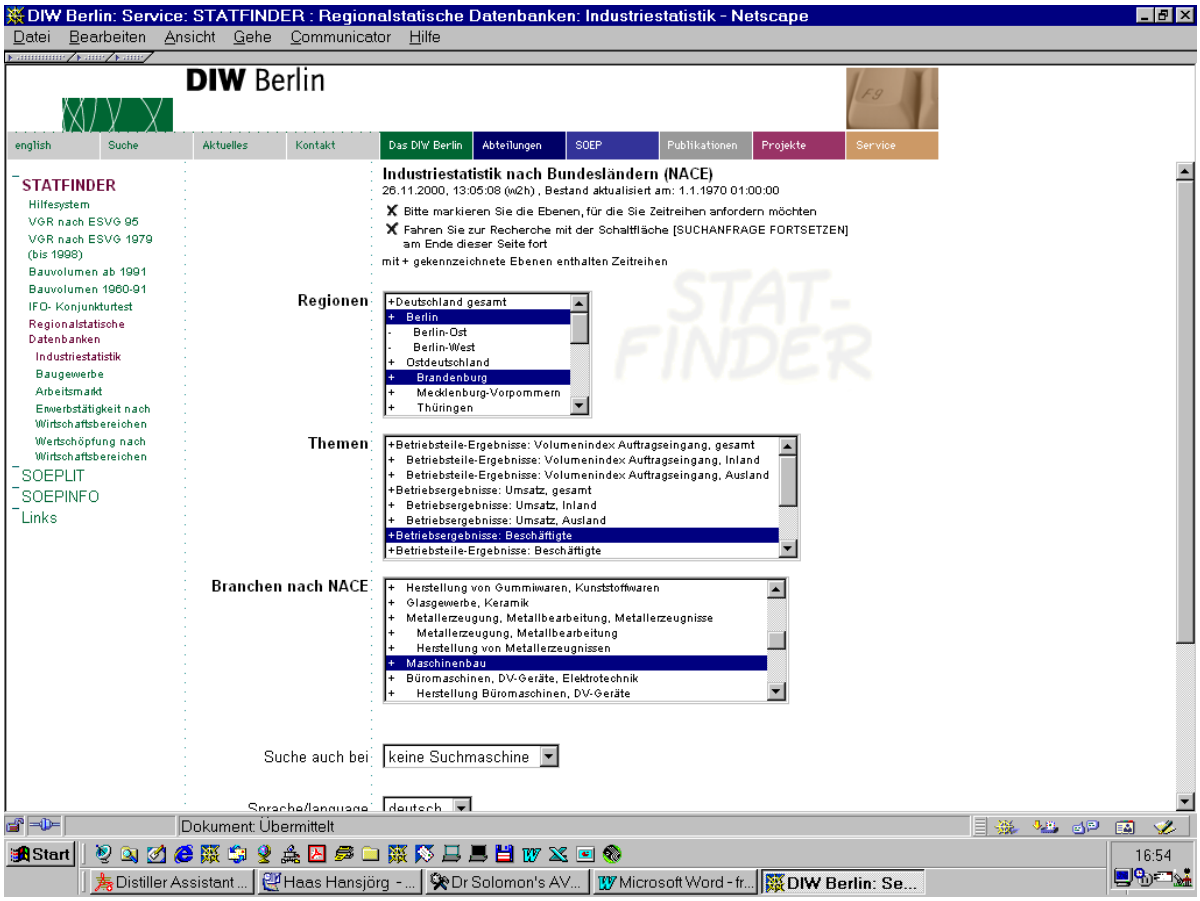



**Abbildung 41: STATFINDER in Aktion, Kategorien**

Als Ergebnis der oben gezeigten Abfrage erhält man eine strukturierte Übersicht mit den in den Zeitreihendatenbanken verfügbaren Zeitreihen.

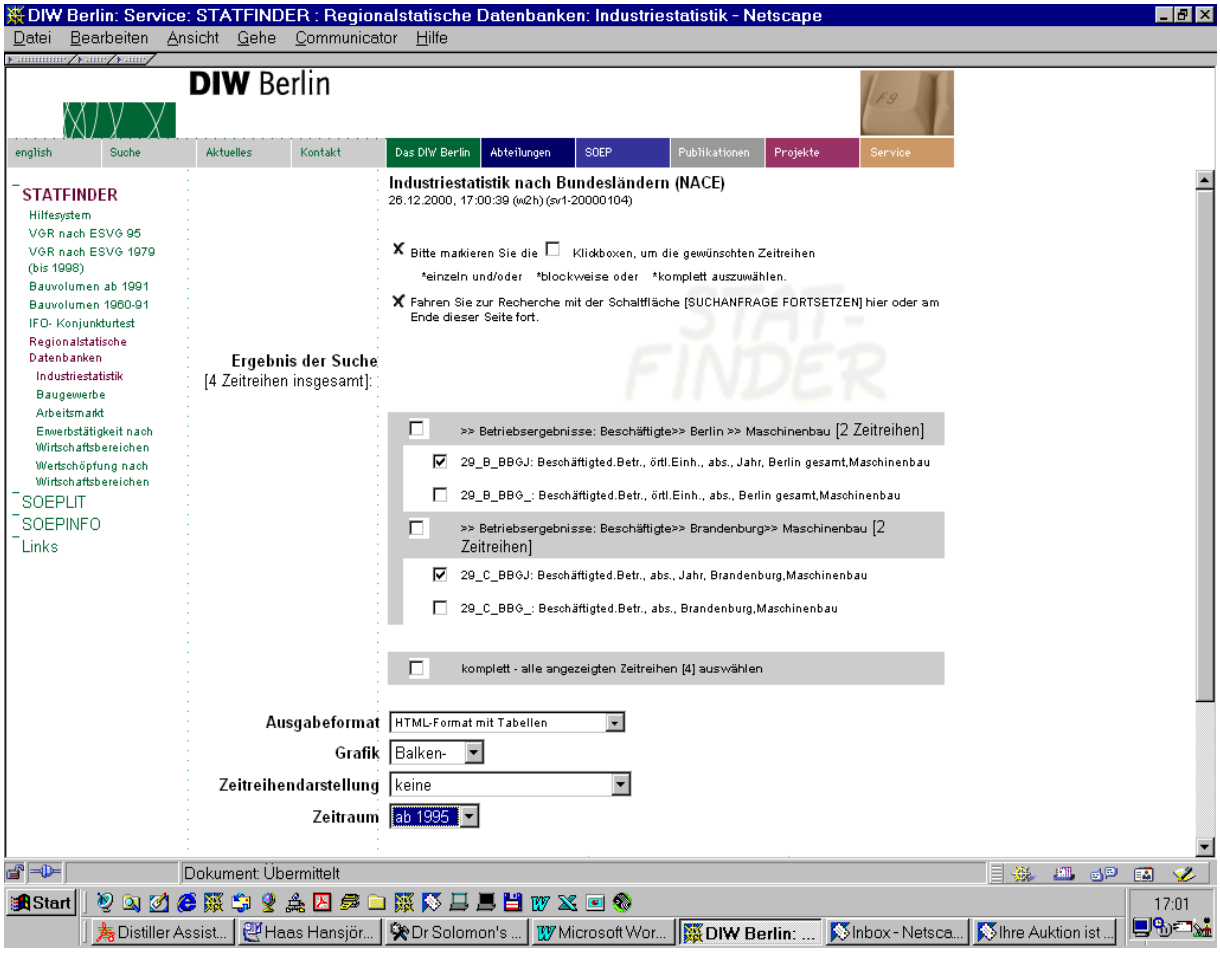

DIW Berlin: Service: STATFINDER : Regionalstatische Datenbanken: Industriestatistik - Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
DIW Berlin
english
Suche
Aktuelles
Kontakt
Das DIW Berlin
Abteilungen
SOEP
Publikationen
Projekte
Service
STATFINDER
Hilfesystem
VGR nach ESVG 95
VGR nach ESVG 1979 (bis 1998)
Bauvolumen ab 1991
Bauvolumen 1960-91
IFO- Konjunkturtest
Regionalstatische Datenbanken
Industriestatistik
Baugewerbe
Arbeitsmarkt
Erwerbstätigkeit nach Wirtschaftsbereichen
Wertschöpfung nach Wirtschaftsbereichen
SOEPLIT
SOEPINFO
Links
Industriestatistik nach Bundesländern (NACE)
26.12.2000, 17:00:39 (w2h) (sv1-20000104)
Bitte markieren Sie die Klickboxen, um die gewünschten Zeitreihen
*einzeln und/oder *blockweise oder *komplett auszuwählen.
Fahren Sie zur Recherche mit der Schaltfläche [SUCHANFRAGE FORTSETZEN] hier oder am Ende dieser Seite fort.
Ergebnis der Suche
[4 Zeitreihen insgesamt]:
>> Betriebsergebnisse: Beschäftigte>> Berlin >> Maschinenbau [2 Zeitreihen]
29_B_BBGJ: Beschäftigted.Betr., örtl.Einh., abs., Jahr, Berlin gesamt,Maschinenbau
29_B_BBG_: Beschäftigted.Betr., örtl.Einh., abs., Berlin gesamt,Maschinenbau
>> Betriebsergebnisse: Beschäftigte>> Brandenburg>> Maschinenbau [2 Zeitreihen]
29_C_BBGJ: Beschäftigted.Betr., abs., Jahr, Brandenburg,Maschinenbau
29_C_BBG_: Beschäftigted.Betr., abs., Brandenburg,Maschinenbau
komplett - alle angezeigten Zeitreihen [4] auswählen
Ausgabeformat
HTML-Format mit Tabellen
Grafik
Balken-
Zeitreihendarstellung
keine
Zeitraum
ab 1995
Dokument: Übermittelt
Start
17:01
Distiller Assist...
Haas Hansjör...
Dr Solomon's ...
Microsoft Wor...
DIW Berlin: ...
Inbox - Netsca...
Ihre Auktion ist ...

**Abbildung 42: STATFINDER in Aktion, Zeitreihenauswahl**

Ergebnisse können grafisch, tabellarisch oder als Download in verschiedensten Formaten abgefragt werden. Hier ein Beispiel für eine grafische Darstellung.

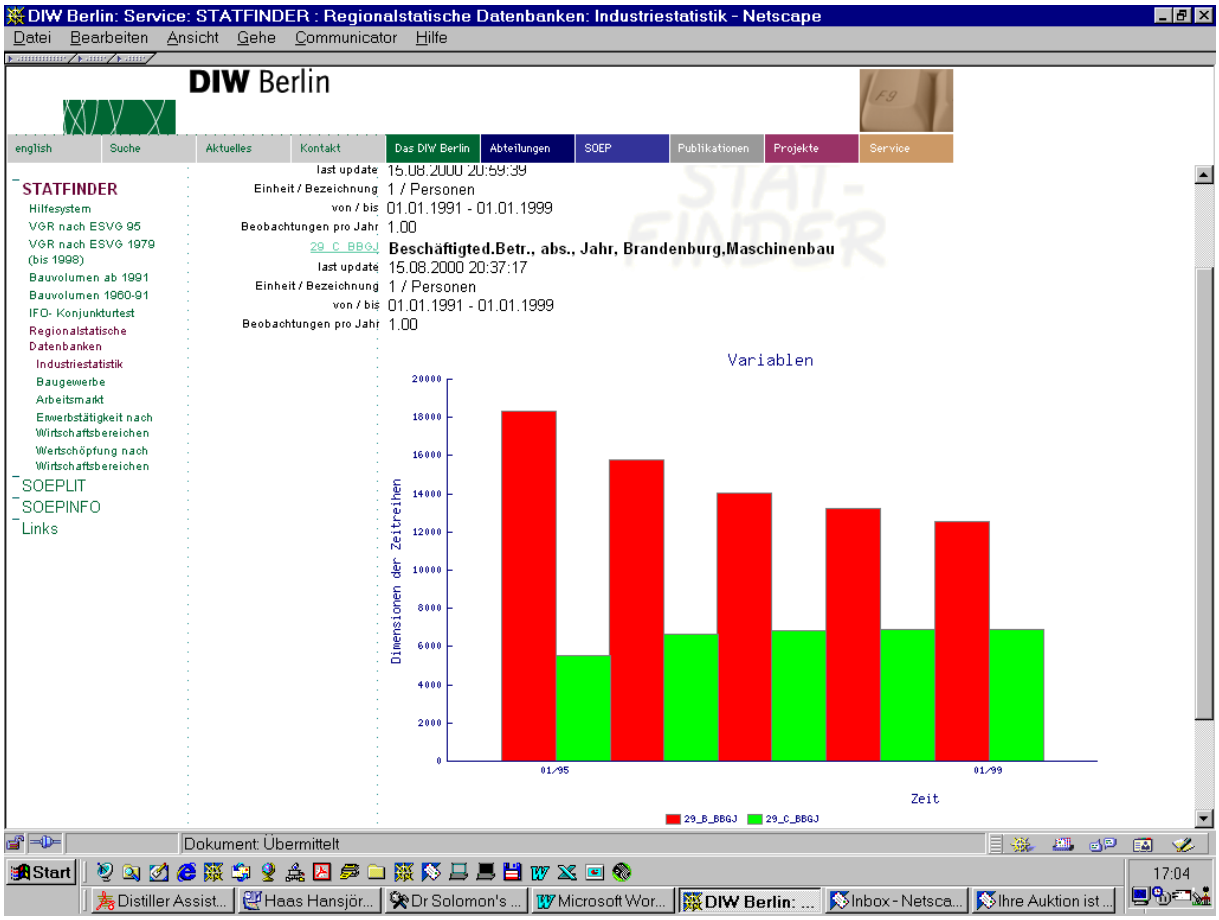

DIW Berlin: Service: STATFINDER : Regionalstatische Datenbanken: Industriestatistik - Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
DIW Berlin
english Suche Aktuelles Kontakt Das DIW Berlin Abteilungen SOEP Publikationen Projekte Service
STATFINDER
Hilfesystem
VGR nach ESVG 95
VGR nach ESVG 1979 (bis 1998)
Bauvolumen ab 1991
Bauvolumen 1960-91
IFO- Konjunkturtest
Regionalstatische Datenbanken
Industriestatistik
Baugewerbe
Arbeitsmarkt
Erwerbstätigkeit nach Wirtschaftsbereichen
Wertschöpfung nach Wirtschaftsbereichen
SOEPLIT
SOEPINFO
Links
last update 15.08.2000 20:59:39
Einheit / Bezeichnung 1 / Personen
von / bis 01.01.1991 - 01.01.1999
Beobachtungen pro Jahr 1.00
29_C_BBGJ Beschäftigted.Betr., abs., Jahr, Brandenburg,Maschinenbau
last update 15.08.2000 20:37:17
Einheit / Bezeichnung 1 / Personen
von / bis 01.01.1991 - 01.01.1999
Beobachtungen pro Jahr 1.00
Variablen
Dimensionen der Zeitreihen
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
01/95
01/99
Zeit
29_B_BBGJ
29_C_BBGJ
Dokument: Übermittelt
Start
17:04
Distiller Assist...
Haas Hansjör...
Dr Solomon's ...
Microsoft Wor...
DIW Berlin: ...
Inbox - Netsca...
Ihre Auktion ist ...

**Abbildung 43: STATFINDER in Aktion, Ergebnis (Grafik)**

Ein interessantes und gern genutztes Feature bei der Suchanfrage ist die (im Beispiel in Abbildung 41 nicht genutzte) Möglichkeit, die Begriffe aus Ergebnislisten mit Hyperlinks zu verschiedenen Suchmaschinen versehen zu lassen. Diese Begriffe können dann angeklickt werden, und eine Suche mit dem ausgewählten Begriff z.B. in Altavista wird angestoßen.

Wir verweisen für weitere Beispiele auf das im Rahmen dieses Projekts erarbeitete Online-Lern- und Hilfesystem zu STATFINDER. Es bietet nicht nur die Einführung in die Technik der Nutzung des Systems, sondern auch weitgefasste inhaltliche Erläuterungen zu den über STATFINDER und STATLINK zugänglichen Datenbanken.

## *3.13. Nutzung der STATFINDER-Technologie in weiteren Web-basierten Lösungen*

### 3.13.1. Die Teleforum-Marktzugangsinitiative

Die breite Nutzbarkeit der im Rahmen des STATLINK-Projekts zur Reife entwickelten STAFINDER-CGI-Technologie wird durch eine weitere Anwendung deutlich, die im Rahmen „Tele-Forum Marktzugangsinitiative“ des Brandenburgischen Ministeriums für Wirtschaft, Mittelstand und Technologie (MWMT) von der Postdamer Firma MIDAT entwickelt wurde. Hier handelt es sich natürlich nicht um die im Rahmen des Projekts vordringlich zu bedienende Klientel aus Universitäten, Fachhochschulen und Schulen, die vom MWMT angesprochen wird. Aber die hier gezeigten Möglichkeiten können ohne weiteres auch für entsprechende Anwendungen durch unsere Klientel genutzt werden, wie die weiter unten gezeigte Anwendung im Rahmen der Infoschul3-Initiative beweist.

Bei Teleforum handelt es sich um eine Art „Electronic Mall“ oder „Internet-Portal“. Interessierte brandenburgische Unternehmen können sich dort mit ihren Unternehmenskennziffern, die in der Industrie-Datenbank des MWMT enthalten sind, präsentieren. (http://www.brandenburg.de/land/mw/midat/index.htm). Hier wird Benchmarking in dem Sinne ermöglicht, daß individuelle Unternehmenskennziffern durch einen Mausklick um Branchendaten für die brandenburgischen Unternehmen insgesamt ergänzt werden können. Zusätzlich soll zukünftig auch der Durchgriff auf die in den DIW-Datenbanken vorhandenen Branchendaten für Deutschland und und die Neuen Länder aus STATISBUND möglich sein. Damit erlaubt das Benchmarking nicht nur einen Vergleich der individuellen Unternehmensentwicklung mit der Entwicklung der Branche in Brandenburg, sondern auch mit derjenigen in den Neuen und in den Alten Bundesländern sowie in Deutschland insgesamt.

Technisch basiert die Lösung auf dem oben beschriebenen DIW-STATFINDER. Die Anwendung wird aus dem Web-Browser heraus durch eine post-Operation realisiert, durch die eine Datenbankabfrage über das Internet am DIW initiiert wird. Deren Ergebnisse (die dann immer die aktuellen Statistiken beinhalten) werden dann in einer dynamisch erzeugten HTML-Seite zusammen mit den Unternehmensdaten aus der IDAB-Datenbank des MWMT dargestellt.

Der Vorteil des Internet, thematisch zusammenhängende Inhalte und Dokumente unabhängig vom Ort, wo diese tatsächlich vorgehalten werden, über Browsertechnik zusammenführen zu können, schafft hier einen echten Mehrwert gegenüber isolierten Anwendungen. Das MWMT, das im Rahmen der Tele-Forums-Initiative die

Bündelung dezentral vorhandener Informationen zu einem qualitativ hochwertigeren Informationsgut veranlaßt, trifft damit die Bedürfnisse von Internetanwendern, die weniger an einzelnen Details, die sie sich selber zusammen suchen müssen, als an kompletten Problemlösungen interessiert sind. Von einer einzigen Internetseite aus kann der Großteil der verfügbaren Informationen zu einem Thema erschlossen werden.

### 3.13.2. Das DIW als Partner für "InfoSchul 3"- Projekte

Das DIW ist daran interessiert, die im Institut verfügbaren Datenbanken mit stabilen Internetschnittstellen für Arbeiten im Bereich Lehre und Forschung anzubieten, wo innerhalb dort entwickelter Anwendungen der Internetzugriff auf aktuelle Wirtschaftsstatistiken erforderlich ist. Als Beispiel seien hier die Kontakte mit der Kaufmännischen Schule II in Villingen-Schwenningen genannt. Diese Schule entwickelt im Rahmen der BMFT-Initiative "InfoSchul 3" eine Internet-Lehreinheit "Wirtschaftskreislauf". Lehrer und Schüler haben zusammen mit dem STATLINK-Team eine grafische Darstellung des Wirtschaftskreislauf erarbeitet, deren Komponenten (Sektoren) und Variablen (z.B. privater Konsum) angeklickt werden können.

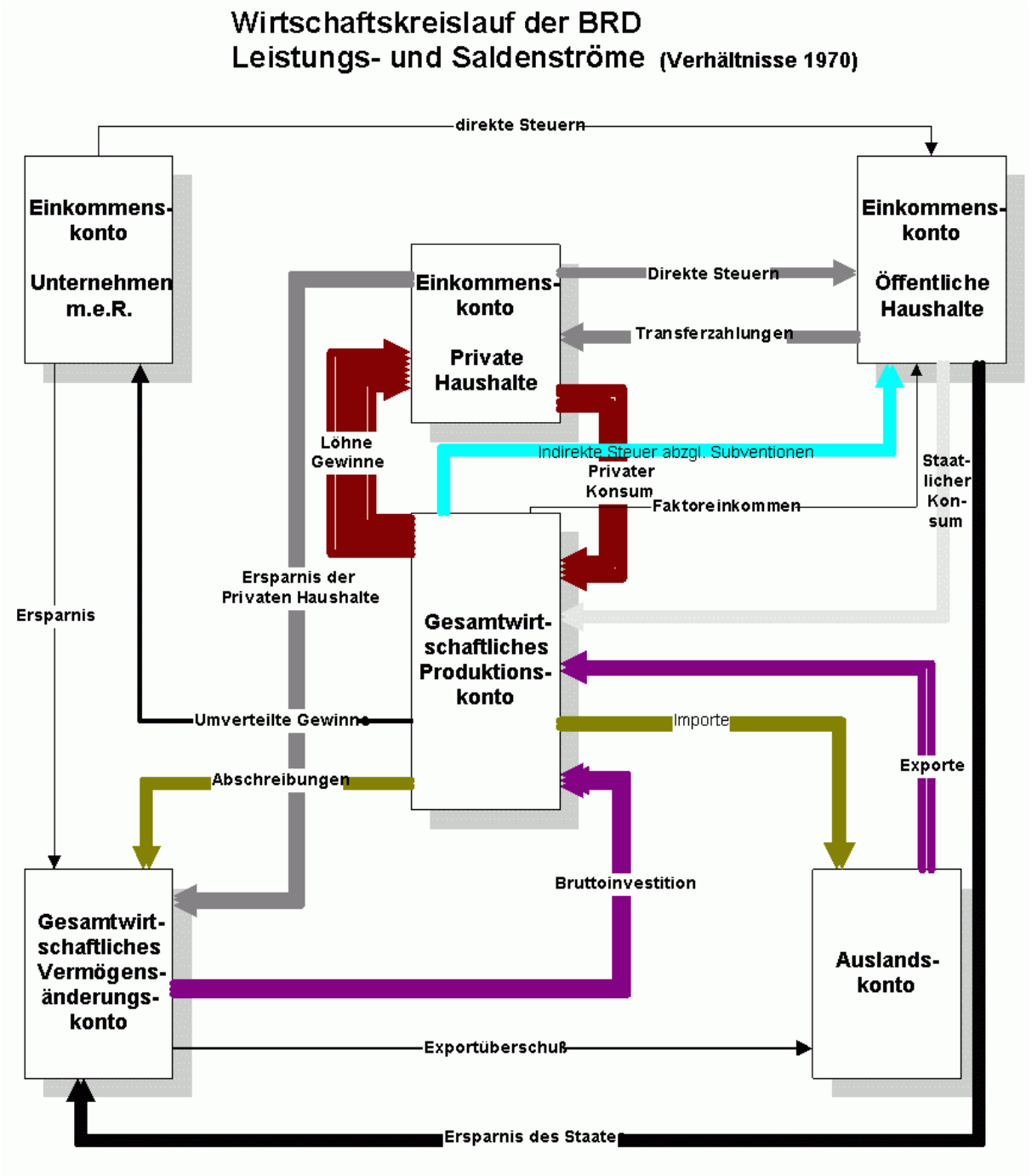



**Abbildung 44: Wirtschaftskreislauf der BRD**

Die zugehörigen Variablen werden dann über das Internet ganz aktuell aus den DIW-Datenbanken gezogen und in Form von Tabellen oder Grafiken präsentiert. Solche "lebenden" Unterrichtshilfen kommen bei den Schülern sehr gut an und erleichtern so die Vermittlung des doch etwas trockenen Lehrstoffes.

Es wird die selbe STATFINDER-Technologie eingesetzt, die auch für die oben skizzierte Teleforum-Anwendung verwendet wurde.

Das DIW Berlin bietet Projekten aus den Bereichen Ökonometrie, quantitative Wirtschaftspolitik oder Wirtschaftsstatistik mit ähnlichen Anforderungen bezüglich des Zugangs zu Wirtschaftsdaten gerne Hilfe bei der Realisierung entsprechender Lösungen an.

# 4. Weiterentwicklungen der Internetlösungen: STATLINK

Der Zugang zu Wirtschaftsdatenbanken über Internettechnik - sei es im Intranet oder im Weitverkehrsnetz - wird zukünftig immer mehr an Bedeutung gewinnen. Im Rahmen dieses Projekts sollte eine Internetlösung entwicklelt werden, die interaktive grafische Darstellungsmöglichkeiten anbietet, der Auswahlprozess ergonomischer gestaltet und mathematisch-statistische Methoden zur Nachbearbeitung der Rechercheergebnisse mit integriert werden müssen. Weiterhin werden neue Retrievaltechniken, die einen komfortableren und schnelleren Zugang zu den gesuchten Informationen erlauben, wie z.B. über Freitextanfragen, implementiert.

Zu diesem Zweck wurde eine völlig neue Applikation auf JAVA-Basis entwickelt, die zum einen einen klassischen Zugang zu Wirtschaftsdaten über geführte Suche mit Hilfe von Kategorien (eine Weiterentwicklugn der der oben gezeigten STATFINDER-Lösung) anbietet, zum anderen die innovative Technik der Freitext-Suchfrage unterstützt.

## *4.1. Die STATLINK-Technik der kategorisierten Suche*

Ausgangspunkt ist das STATLINK-Ausgangsmenü, das im oberen Menübalken die Buttons für die beiden Suchoptionen, Kategorisierte Suche bzw. Freitextsuche sowie für die Ergebnisanzeige anbietet, und das in der linken Leiste eine Reihe von Buttons für die verfügbaren Datenbanken enthält. Zusätzlich steht ein Popup Fenster für die Sprachauswahl bereit - ein Feature, das überall da eingesetzt werden kann, wo die Datenbanken auch oder nur in englischer Sprache gehalten werden.

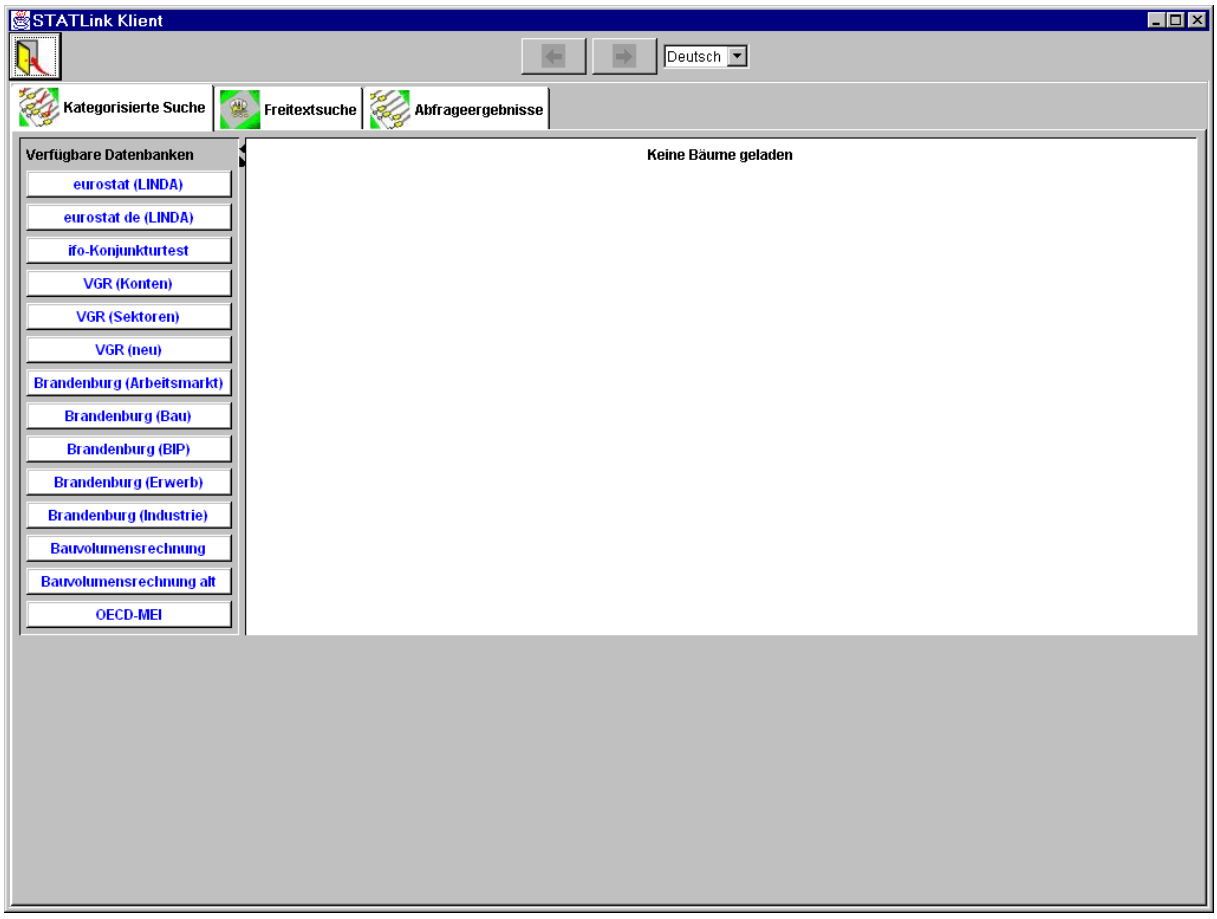



**Abbildung 45: STATLINK Client- Hauptfenster**

Voreingestellt ist die Kategorisierte Suche. Nach Auswahl einer Datenbank, z.B. ifo Konjunkturtest, erhält man im bisher freien Anzeigefenster die verfügbaren Kategorien angezeigt und die Merkmale, die abgefragt werden können. Dabei wird die Folder on/off Technik angeboten, um hierarchische Strukturen (wie Branchensystematiken oder Warengruppen) vom Aggregat zum Detail hin aufklappen zu können:

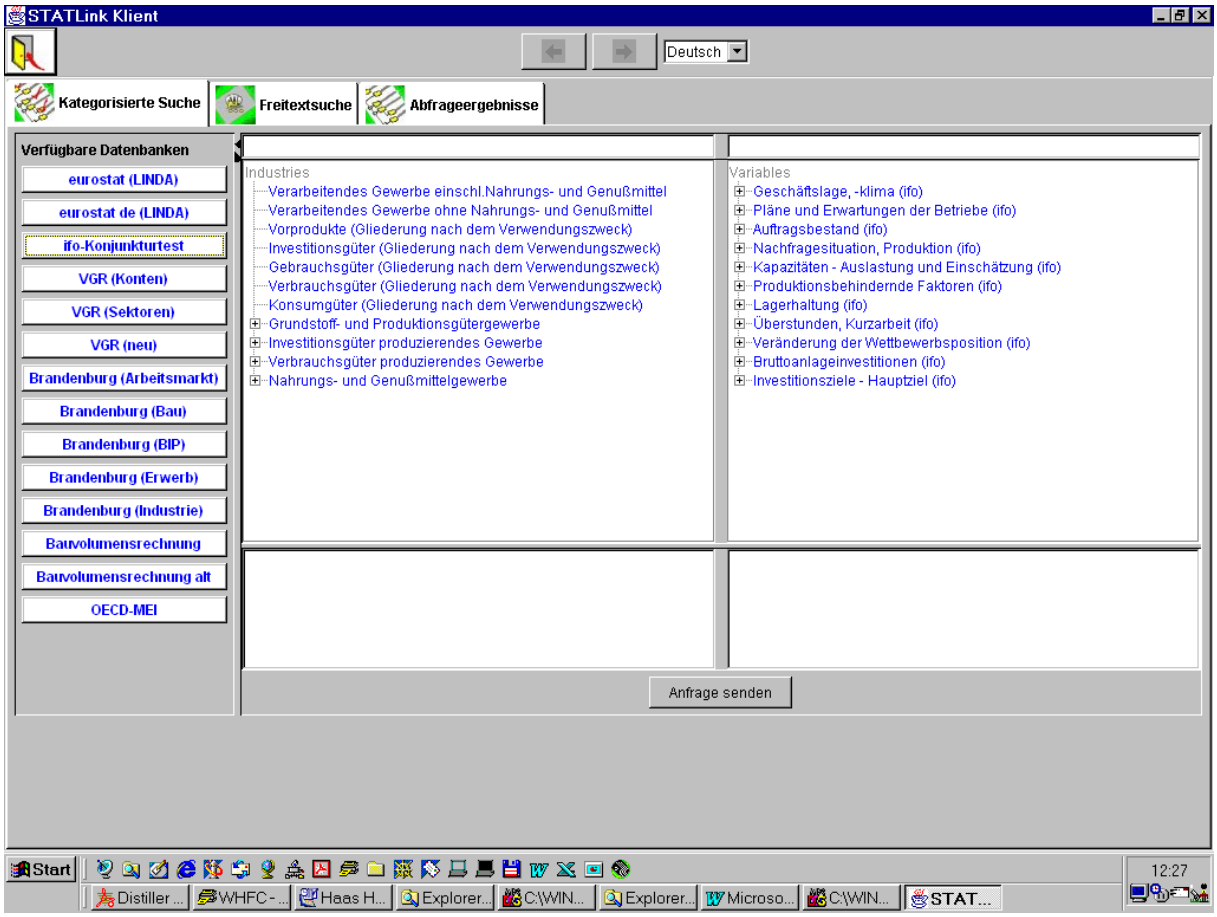



**Abbildung 46: STATLINK Client- Kategorisierte Suche**

Das Auswahlfenster enthält immer soviel Spalten wie die Datenbank Kategorien hat (hier die Kategorien Wirtschaftszweige und Themen) und zwei Zeilenblöcke.

Der obere Teil des Auswahlfensters bietet die Sicht auf die gesamte Datenbank ifo Konjunkturtest. Überall, wo vor einem Eintrag ein Pluszeichen steht, kann das Merkmal weiter disaggregiert werden, z.B. für das Nahrungs- und Genussmittelgewerbe (letzte Zeile) in Ernährungsgewerbe und Tabakgewerbe, wobei das Ernährungsgewerbe noch weiter untergliedert werden kann. Durch Anklicken eines oder mehrerer Begriffe in diesen Auswahlfenstern wird eine Suchanfrage erstellt, die im darunter leigenden Protokollfenster notiert wird. Aus den Auswahlfenstern können nur Merkmale angeklickt werden, während die Protokollfenster editiert werden können, d.h. Eointräge können durch Doppelklick wieder entfernt werden.

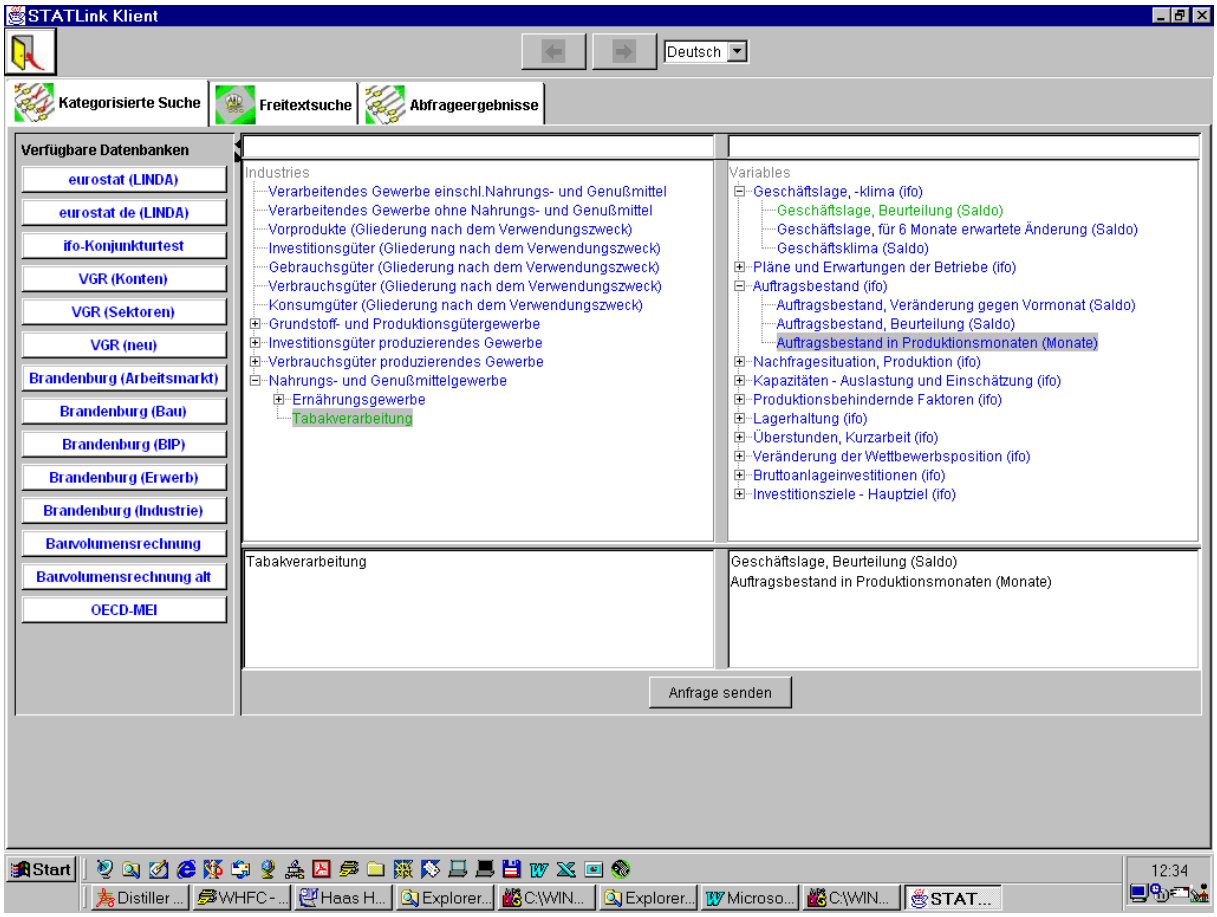



**Abbildung 47: STATLINK Client- Kategorisierte Suche mit Auswahl**

Im oben gezeigten Beispiel interessiert man sich, wie aus dem Protokollfenster zu ersehen ist, für die Branche Tabakverarbeitung, und zwar für die Beurteilung der Geschäftslage und für den Auftragsbestand in Produktionsmonaten.

Nach Anklicken des Buttons "Anfrage senden" wird die Verbindung zum DIW-Datenbankserver aufgenommen, der dem STATLINK-Client die Ergebnisse zurückgibt:

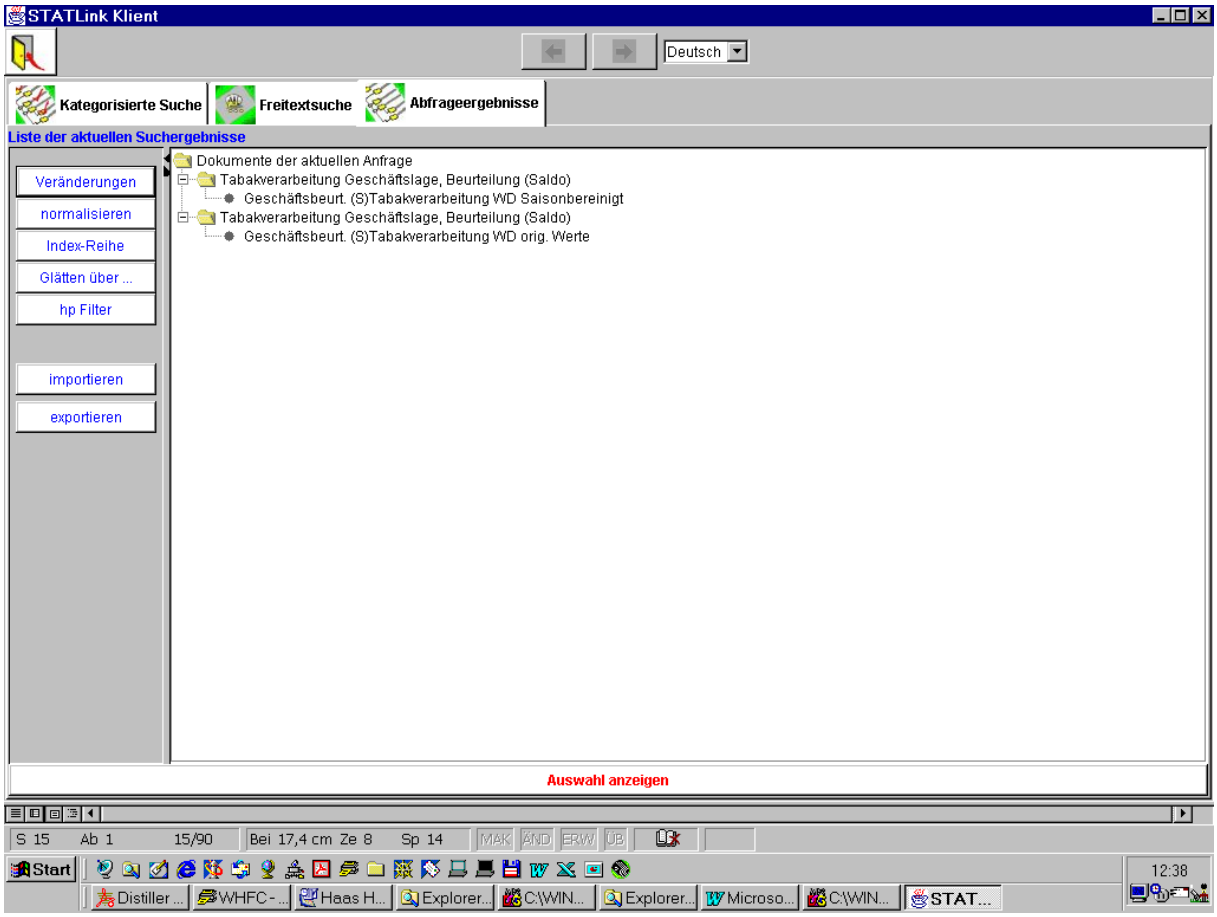



**Abbildung 48: STATLINK Client- Retrieval-Ergebnis, Zeitreihen**

Wie zu sehen ist, wurden nur Zeitreihen für die Beurteilung der Geschäftslage, nicht auch für die Auslastung zurückgegeben. Letzte existieren damit in der ausgewählten Datenbank für den Wirtschaftszweig Tabakverarbeitung gar nicht.

Die Retrievalergebnisse können nun grafisch oder tabellarisch dargestellt werden. Dazu werden eine oder mehrere der offerierten Suchergebnisse markiert, und dann der Button "Auswahl anzeigen" gedrückt:

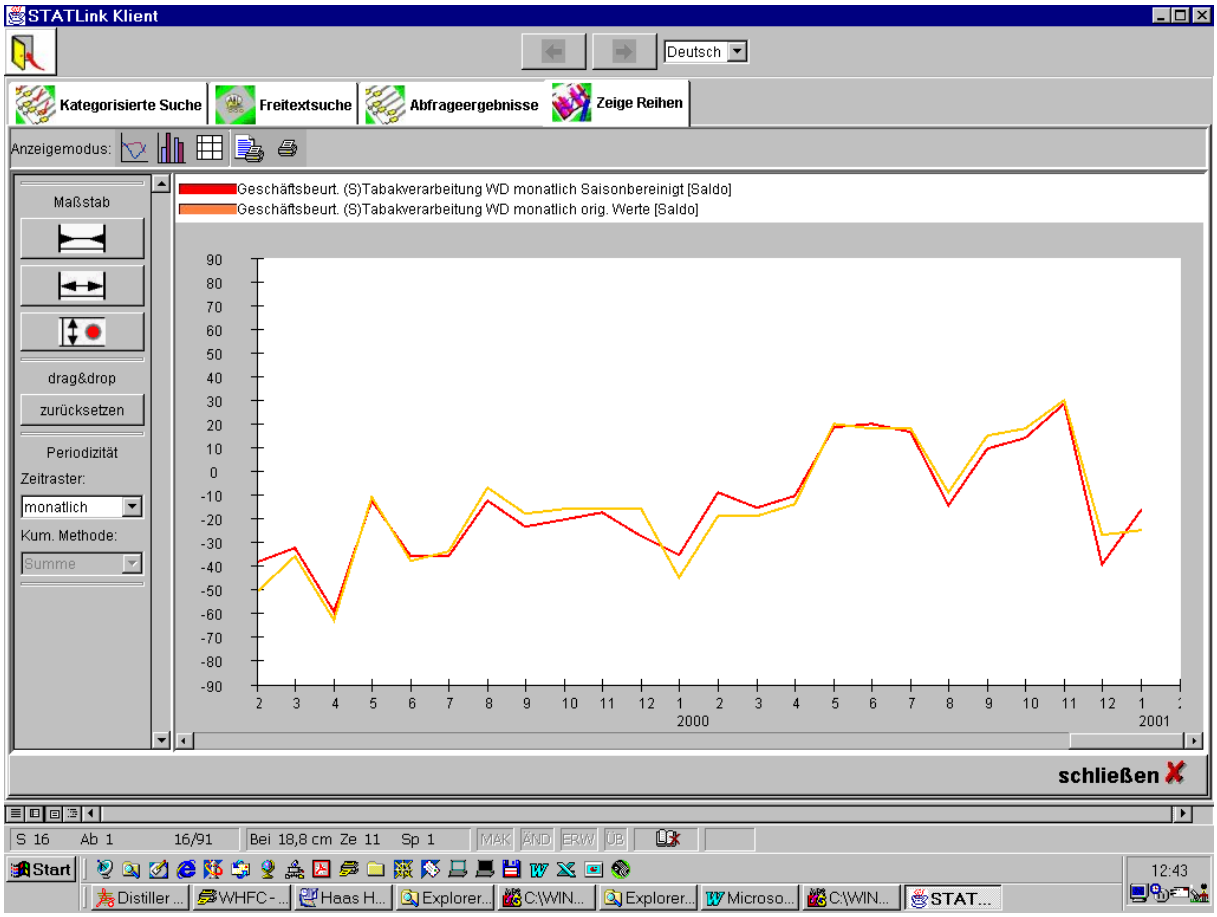



**Abbildung 49: STATLINK Client- Zeitreihendarstellung I**

Oder als Balkengrafik:

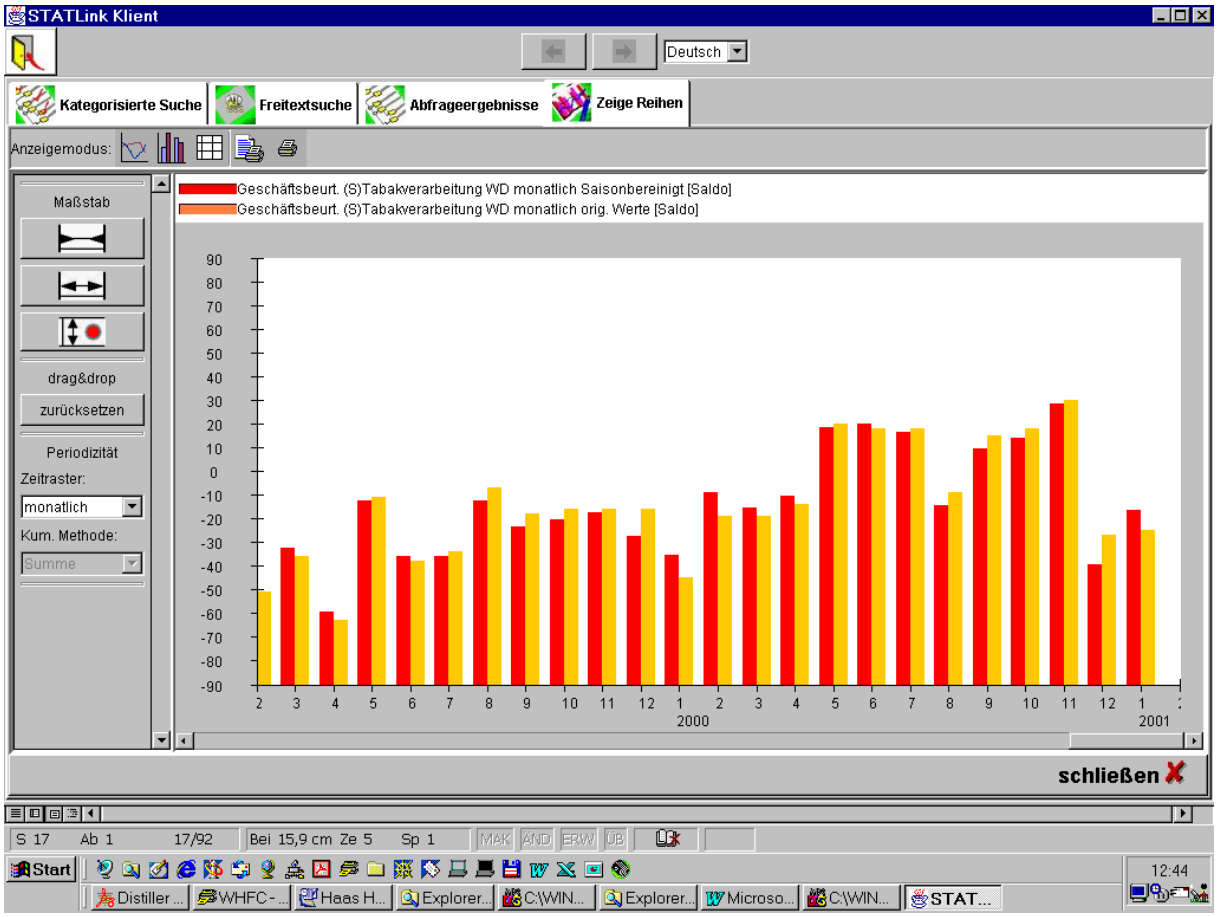



**Abbildung 50: STATLINK Client- Zeitreihendarstellung II**

Es stehen eine ganze Reihe einfacherer Methoden (Skalierung, Differenzenbildung, Aggregation über die Zeit) zur Verfügung, die auf die heruntergeladenen Zeitreihen angewandt werden können. Eine genauere Datstellung findet sich hierzu in den Online-Hilfen des Systems.

## *4.2. Verbesserung der Retrievalmöglichkeiten durch Freitextsuche*

Neben dem Ziel, unmittelbar die Versorgung des Bereichs Lehre und Forschung mit Wirtschaftsinformationen zu verbessern, werden im Rahmen des Projekts auch Fortschritte bei der Retrievaltechnik und der Visualisierung der Ergebnisse angestrebt. Dabei ist die Weiterentwicklung der statischen CGI-Lösung zu einer leistungsfähigen Java-Applikation, wie sie oben am Beispiel der kategorisierten Suche gezeigt wurde, nur ein Meilenstein, der zu erreichen ist. Darüber hinaus geht es zentral um die Verbesserung der Retrievaltechniken durch die Einführung der Freitextsuche.

Zielsetzung ist es, die Komplexität eines Suchvorgangs für den Nutzer auf ein Minimum zu reduzieren und wenig Vorwissen über vorhandenen Datenbanken und deren mehr oder weniger komplexe Struktur vorauszusetzen. Natürlich kommt diese Fortentwicklung der Suchtechnologie auch dem fachlich versierten Studenten oder Wirtschaftswissenschaftler zugute, da er seine Anfragen einfacher formulieren kann und seinen Recherchen dann nur noch innerhalb eines stark eingeschränkten Suchraums durchführen muss.

Jeder kennt und nutzt heute die Internetsuchmaschinen (Yahoo, Altavista, Google, Lycos usw.), die aber keinen Durchgriff auf Informationen haben, die in

wirtschaftsstatistischen Datenbanken abgespeichert sind. Suchergebnisse sind weitgehend Referenzen zu Textdokumenten. Im Rahmen des STATLINK-Projekts wird jetzt eine Schnittstelle angeboten, die es erlaubt, beliebige Texte als Freitextanfrage ins System zu geben. Als Ergebnis erhält der Benutzer am DIW verfügbare Datenbanken angeboten, die als geeignet angesehen werden, Informationen zu liefern, die mit dem Text im Zusammenhang stehen. Und zwar mit einer Beschränkung des Suchraumes auf die Bereiche der Datenbanken, die im Kontext der als Freitext eingegebene Suchanfrage eine Relevanz haben können.

Wie kann diese Technologie genutzt werden? Z.B. kann eine Suche im Internet zur Wirtschaftslage in Deutschland einen der DIW-Wochenberichte erbringen. Man möchte diese Informationen gerne mit aktuellen Wirtschaftsstatistiken unterfüttern. STATLINK ermöglich es nun, die aus dem Internet gezogenen Text oder besser relevante Teile davon als Anfrage einzugeben. Dazu sucht man direkt unter Verwendung des Wochenberichtes,

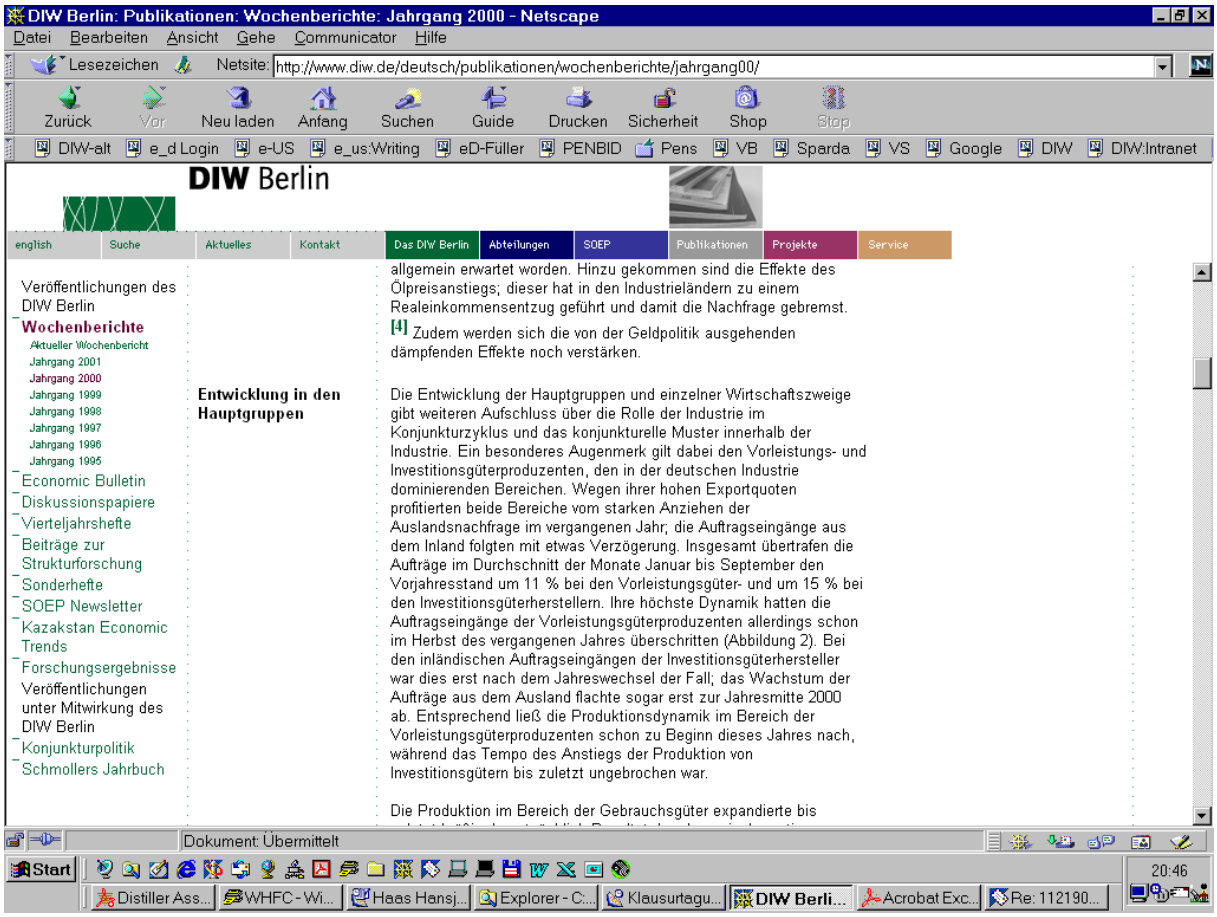



**Abbildung 51: Webseite des DIW Berlin mit DIW-Wochenbericht Nr. 49/2000**

indem man diesen (oder wie hier geschehen Teile davon, der erste Absatz des Abschnitts "Entwicklung in den Hauptgruppen") via Zwischenablage in die Freitextsuche-Funktion des STATLINK- Clients eingibt, siehe Abbildung 52.

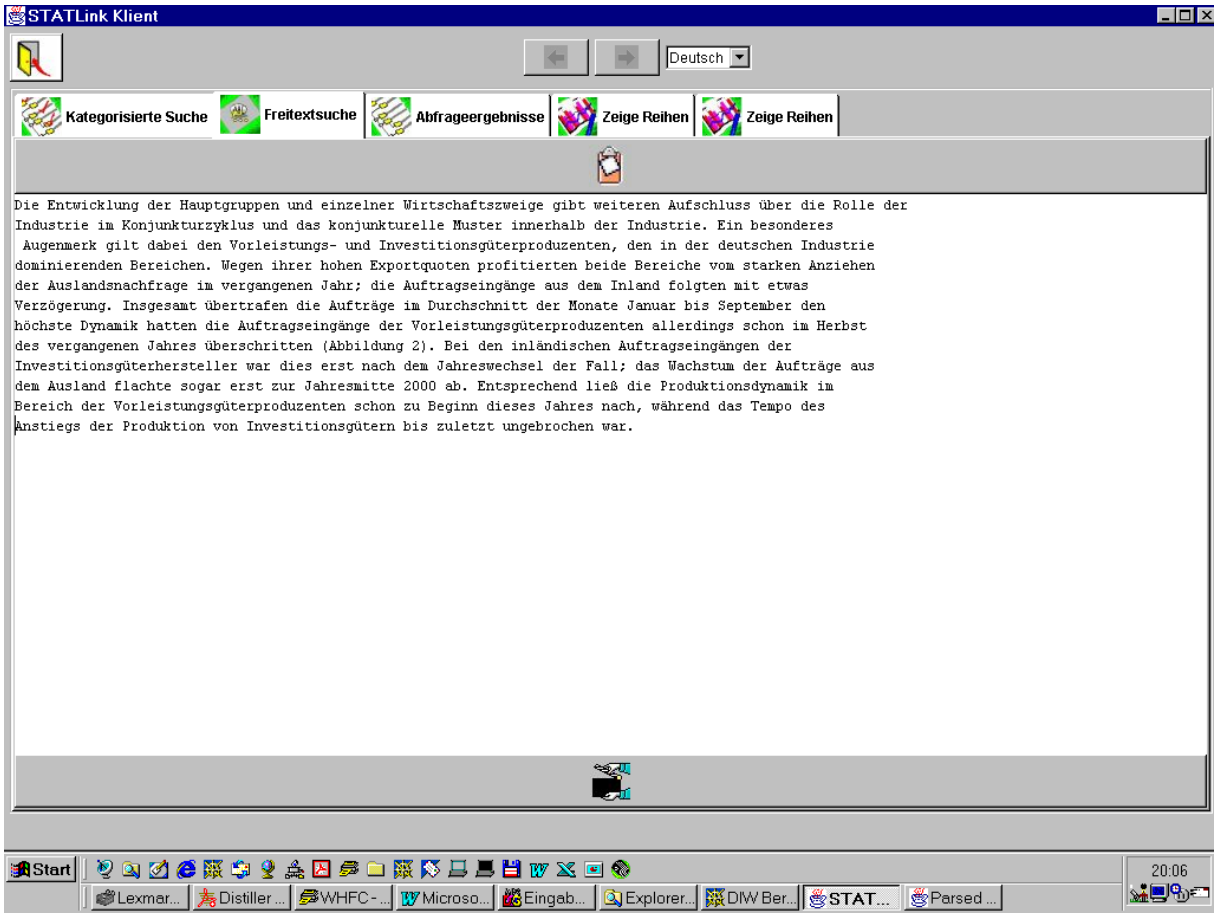



**Abbildung 52: STATLINK- Client, Freitextsuche mit dem DIW-Wochenbericht Nr. 49/2000**

Dieses Fenster ist editierbar, d.h. der Benutzer kann bei Bedarf weitere ihn interessierende Begriffe eintragen.

Nach Abschicken der Freitextanfrage wird im ersten Schritt der Freitext mit markierten Begriffen, für die Datenbankinhalte vorhanden sind, zurückgegeben.

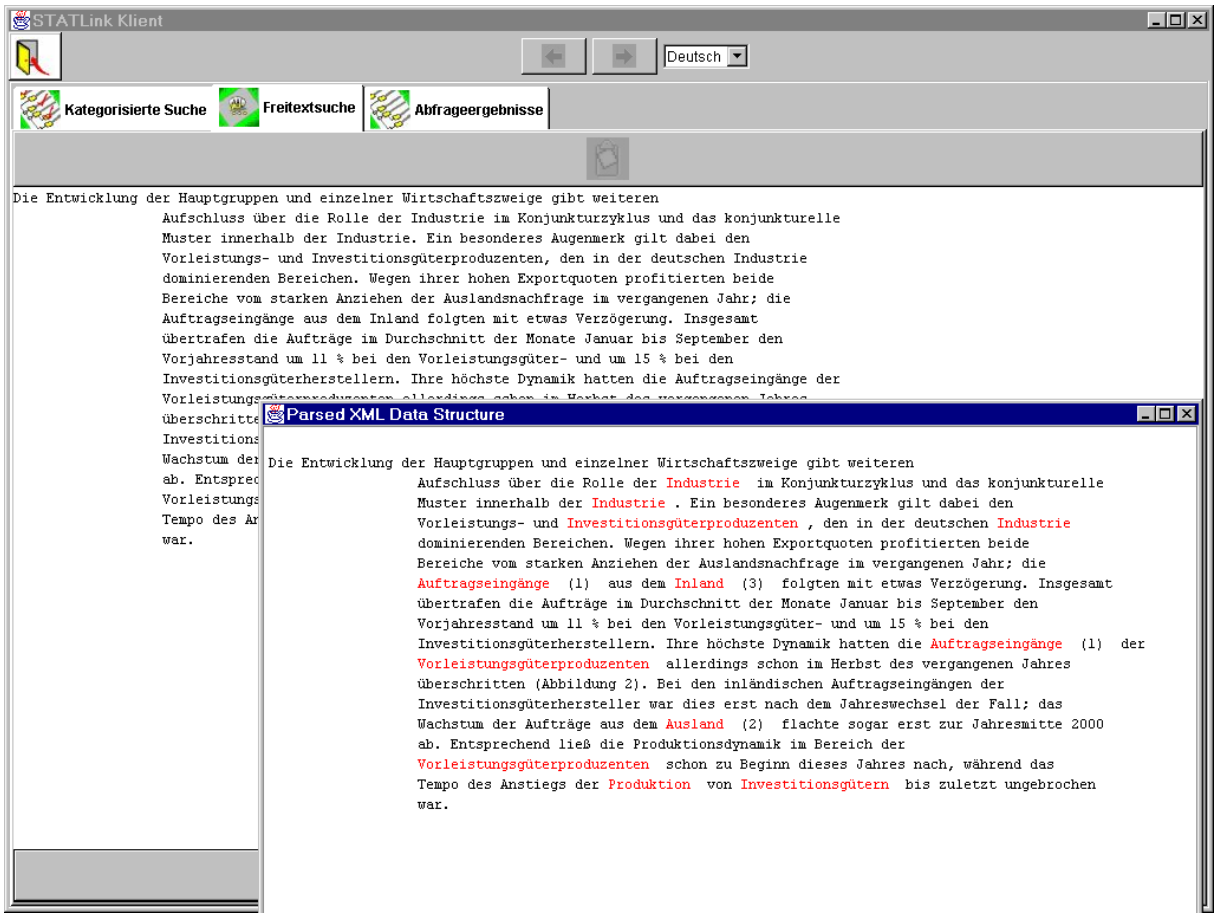



**Abbildung 53: STATLINK- Client, Ergebnis der Freitextsuche**

Wird einer dieser mit Hyperlinks vergleichbaren markierten Begriffe angeklickt (z.B. der Begriff "Produktion"), erhält man ein Auswahlmenü mit der Anzeige, in welchen Datenbanken sich Informationen zum Thema Produktion befinden. Sollte nur eine geeignete Datenbank vorhanden sein, wird das Retrieval unmittelbar angestoßen.

In dem verwendeten Beispiel mussür eine der zwei angebotenen Datenbanken muss eine Entscheidung getroffen werden.

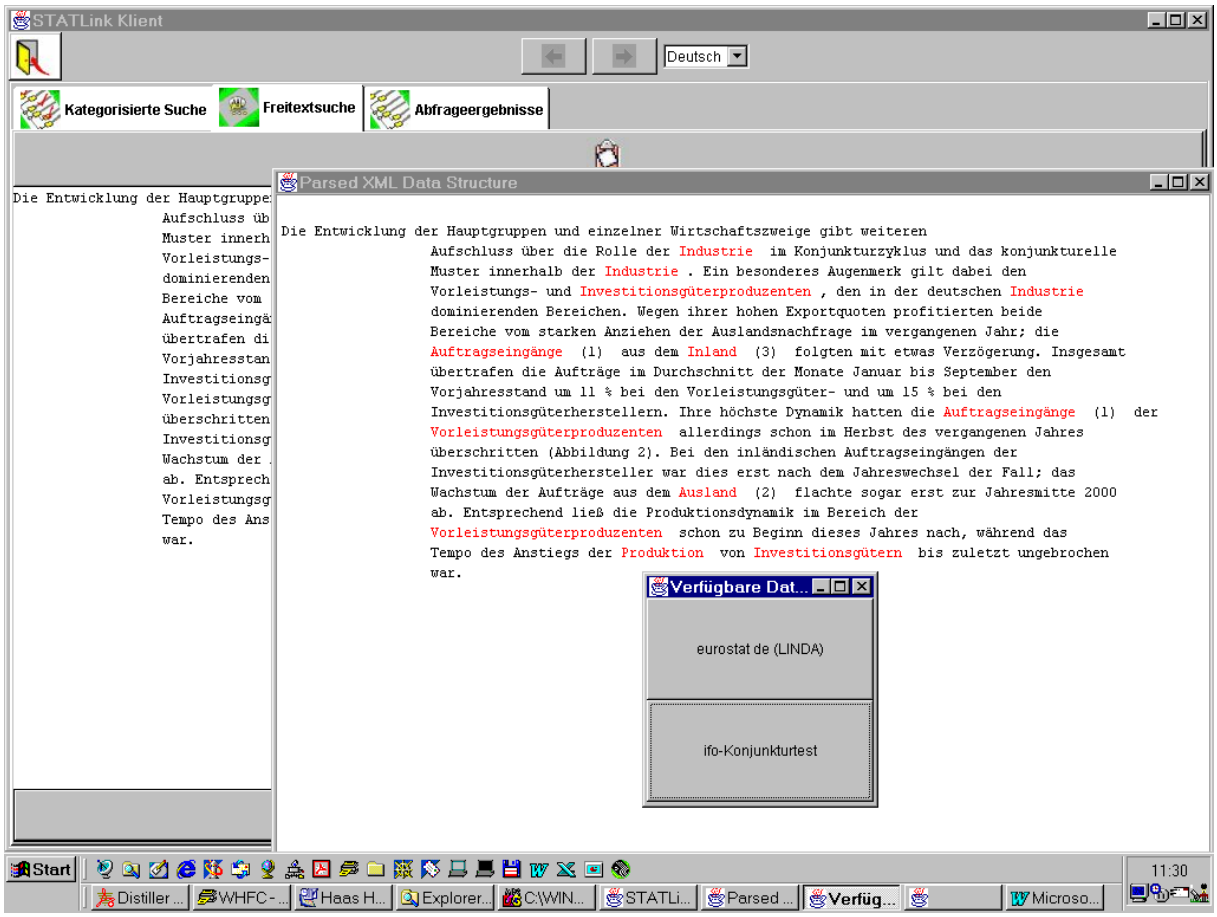



**Abbildung 54: STATLINK-Client, Ergebnis der Freitextsuche mit Datenbankauswahlfenster**

Entscheidet man sich für den ifo-Konjunkturtest, beziehen sich alle folgenden Schritte auf diese Datenbank:

Im ersten Schritt wird untersucht, wie der Suchraum unter Berücksichtigung des gewählten Begriffs Produktion und der anderen im Freitext befindlichen Begriffe eingeschränkt werden kann:

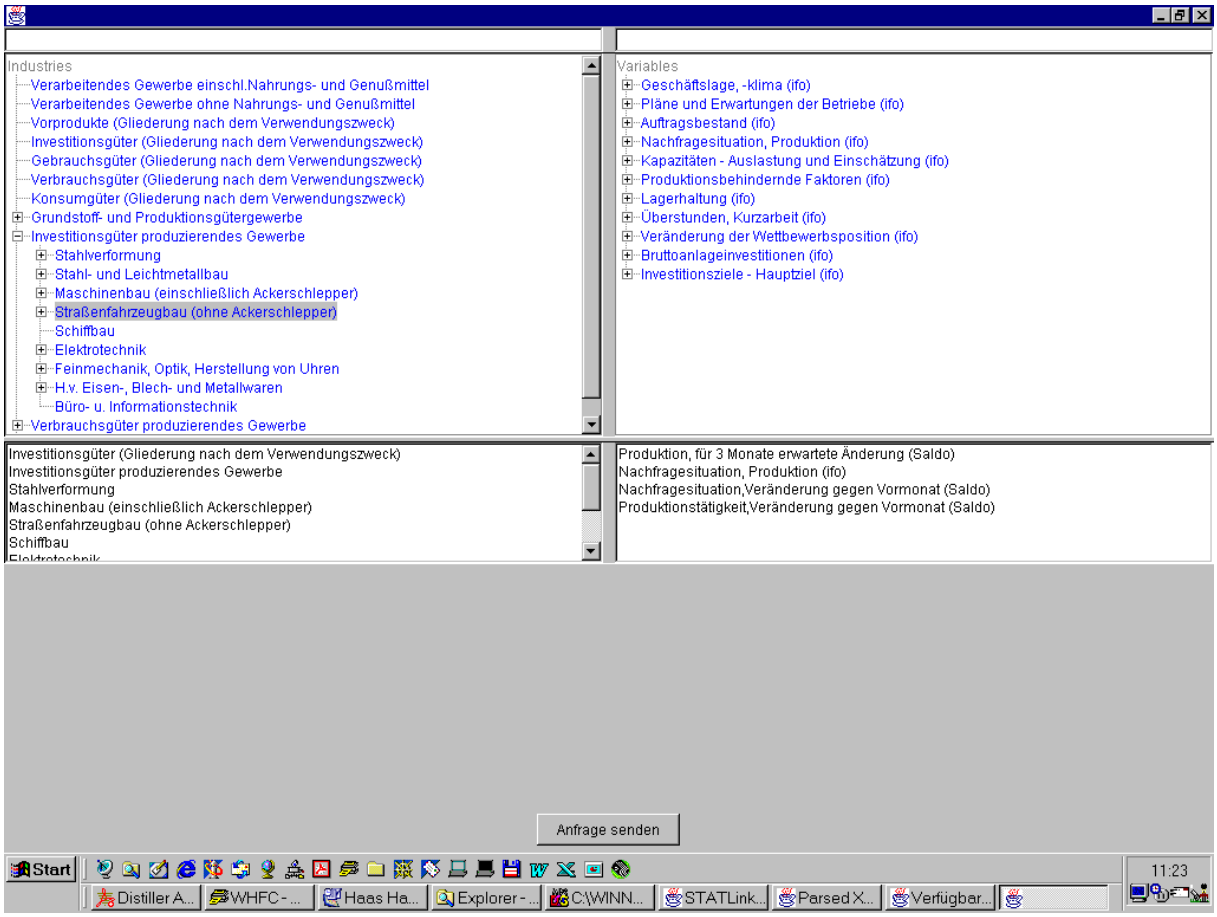



**Abbildung 55: STATLINK- Client, Ergebnis der Freitextsuche mit automatisch erstellter Suchanfrage in den Protokollfenstern**

Wie zu sehen ist, bietet die Datenbank den Suchraum in den beiden Kategorien Branchen und Themen an. Für beide Kategorien wurden im Protokollfenster schon Auswahlen getroffen: bei den Branchen die übergeordneten Bereich zum Investitionsgütergewerbe und die darunter liegenden Aggregate, bei den Themen diejenigen Merkmale, die in einem Zusammenhang mit dem Begriff "Produktion", der ja Ausgangspunkt der Recherche war.

Als Ergebnis der Recherche erhält man eine Auflistung mit Kombinationen der Protokollfenstereinträge, zu denen Zeitreihen vorhanden sind. Eine Kombination wäre z.B. Stahlverformung aus der Branchenkategorie und Nachfragesituation aus der Themenkategorie. Sind für eine Kombination mehrere Reihen verfügbar (z.B. als Original- und als saisonbereinigte Reihe), kann die Kombination entsprechend häufig angezeigt werden:

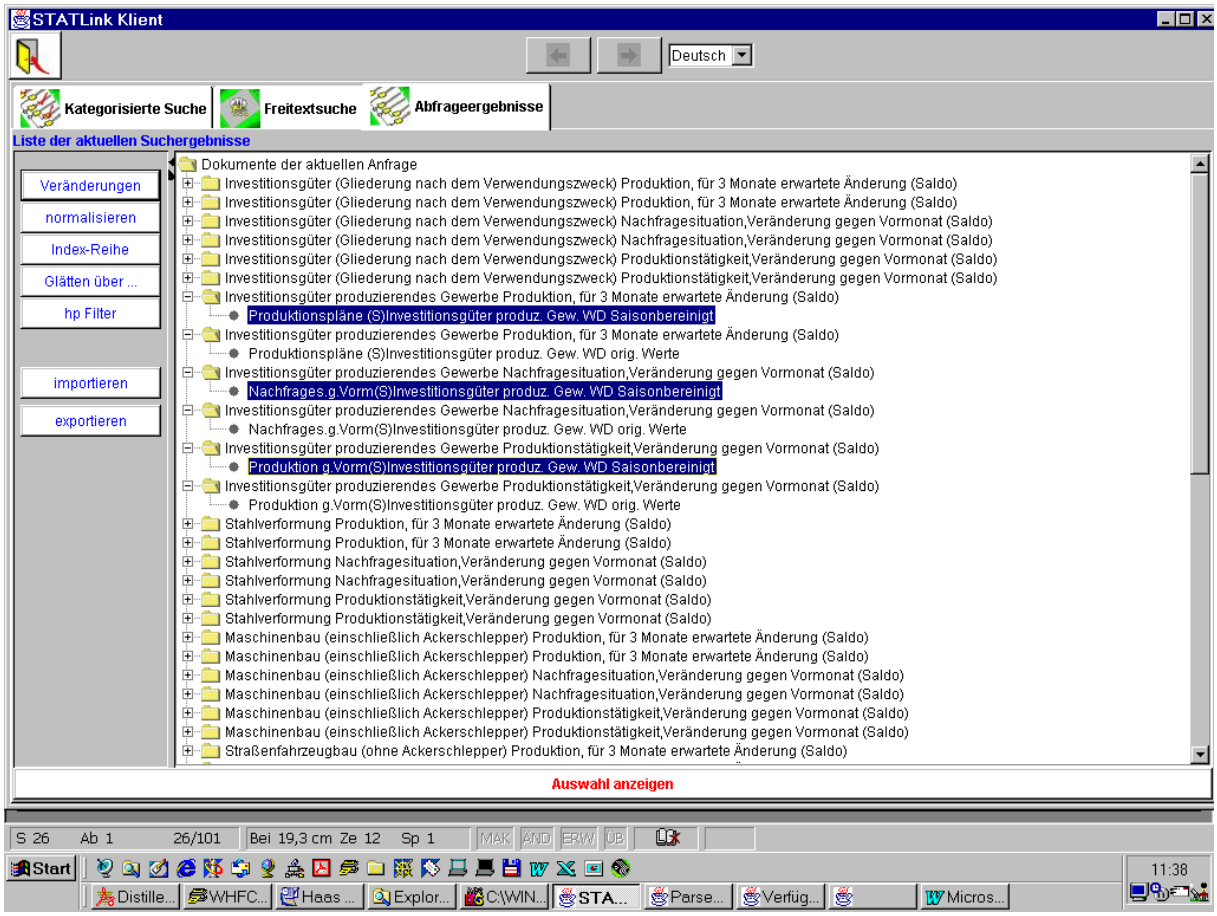



**Abbildung 56: STATLINK- Client, Zeitreihen in der lokalen Datenbasis des Clients**

Aus der Zeitreihenergebnisliste sind von uns drei Reihen ausgewählt worden, eine graphische Darstellung folgt:

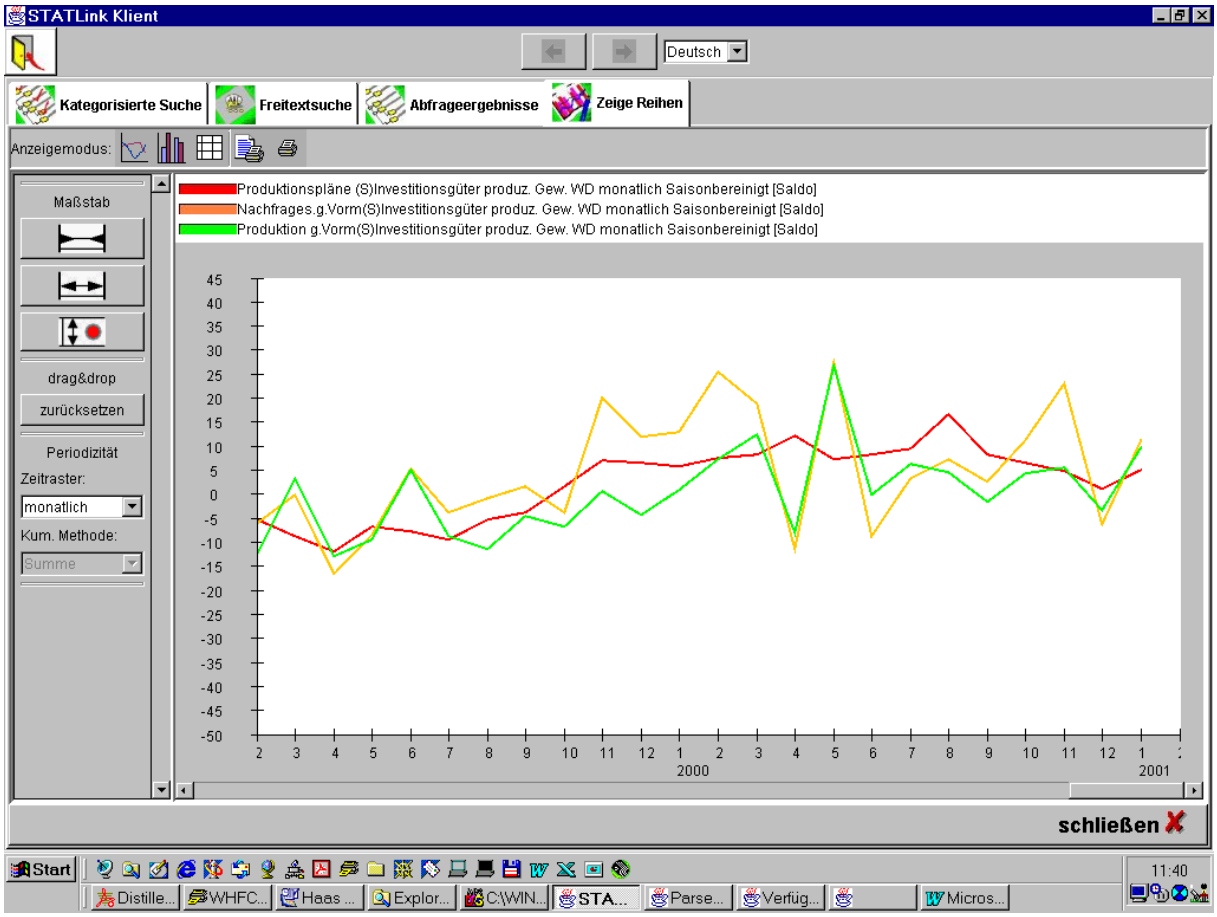



**Abbildung 57: STATLINK- Client, Grafische Zeitreihendarstellung (kombiniert)**

Die sonstigen Visualisierungsmethoden sowie die verfügbaren Methoden entsprechen denen bei der im vorhergehenden Kapitel dargestellten kategorisierten Suche und sollen hier deshalb nicht noch einmal gezeigt werden.

Wichtig ist der Hinweis, dass keine Festlegung auf eine bestimmt Datenbank erfolgt: Jedem der als relevant erkannten Begriff aus der Freitextanfrage wird eine oder mehrere passende Datenbanken zugeordnet. Bei mehreren in Frage kommenden Datenbanken muss der Benutzer sich für das nachfolgende Retrieval für eine davon entscheiden. Er kann aber jederzeit den Suchbegriff nochmals anklicken und dann auf eine der anderen angebotenen Datenbanken (im obigen Beispiel Eurostat-Linda) wechseln

Von Begriff zu Begriff ist also die Zuordnung von Datenbanken unterschiedlich. Hätte man anstelle des Begriffs "Produktion" den Begriff "Auftragseingänge" angeklickt, wäre eine andere Datenbankauswahl angeboten worden.

Es besteht auch jederzeit die Möglichkeit, den Suchraum zu vergrößern, und beispielsweise aus den Auswahlfenstern weitere Merkmale für ein Retrieval zu selekieren. Dies ist wichtig, da der Textkörper, der als Freitextanfrage übergeben wird, oft nicht alle relevanten Begriffe mitbringt. Zum Beispiel enthalten die Pressemitteilungen des Statistischen Bundesamtes, mit denen wir das System getestet haben, fast nie regionale Begriffe, beziehen sich aber meist auf Deutschland. Nichtvorhandene Hinweise können natürlich nicht für die automatische Erstellung der Anfrage verwendet werden. Aber durch die Tatsache, dass der Nutzer aus dem Protokollfenster die Einschränkung des Suchraums ersehen kann,

gleichzeitig aber in den Auswahlfenstern den kompletten Suchraum angeboten bekommt, kann er ihn interessierende weitere Elemente der Sicht problemlos in die Suchanfrage einfügen, so z.B. in dem genannten Beispiel den Regionenbegriff Deutschland.

## 5. Literaturnachweise

[Belkin 1992] Belkin, N.J.; Croft, W.B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin? In: Communications of the ACM, 35 (12) 1992, S. 29-38.

[Broglio 1994] Broglio, J.; Callan, J.P.; Croft, W.B.: INQUERY System Overview. In: Proceedings of the TIPSTER Text Programm (Phase1), San Francisco 1994, S. 47-67.

[Callan 1991] Callan, J.P.; Croft, W.B.; Harding. S.M.: The INQUERY Retrieval System. In: Proceedings of the Third International Conference on Database and Expert Systems, 1991.

[Croft 1995] W.B. Croft: What Do People Want From Information Retrieval? In: d-lib Magazine. Nov. 1995.

[Croft 1995a] Croft, W.B.: Integrating Text and Database Systems: Possibilities and Challenges. In: Proceedings of the Second East/West Database Workshop. Klagenfurt, Austria, 1994. Workshop in Computing, Springer-Verlag, 1995.

[Drosdowski 1984] Der Duden in 10 Bänden: Standardwerk zur deutschen Sprache, Bd. 4 „Grammatik der deutschen Gegenwartssprache“, Bibliographisches Institut Mannheim, 1984

[Englmeier 1998] Englmeier, K., Haas, H., Lindlbauer, J., Pauer, B. (1993): WIDAB-Projekt: Aufbau einer benutzerfreundlichen Wirtschaftsdatenbank WIDAB - Praxisgerechter Ausbau und Übergabe an privatwirtschaftliche Hosts.. Endbericht. Berlin und München 1993.

[Fapsy-Consortium 1998] Fapsy-Consortium. (1998): FAPSY - Towards a New Approach in Presenting Statistical Information Spaces (ESPRIT IP 20343), Final Report.

[Fuhr 1994] 5 Fuhr, Norbert: Probabilistic Models in Information Retrieval. In: Computer Journal 35 (3) 1992, S. 243-255.

[Luhn 1959] “Auto-Encoding of Documents for Information Retrieval Systems”. In M. Boaz (editor). Modern Trends in Documentation. London: Pergamon Press, S. 45-58

[Matthes 1993] Matthes, Florian: Persistente Objektsysteme: Integrierte Datenbankentwicklung und Programmerstellung. Springer-Verlag, 1993.

[Matthes 1994] Matthes, Florian; Müßig, Sven; Schmidt, Joachim W.: Persistent Polymorphic Programming in Tycoon: An Introduction. Technical Report FIDE/94/106, Glasgow, 1994.

[Pfeiffer 1995] Pfeiffer, U.: HTTPs älterer Bruder: WAIS: Inhaltsorientierte Suche im Internet. In: iX (1) 1995, S. 120.

[Salton 1983] Salton, G.; McGill M. J.: Introduction to Modern Information Retrieval

[Scheinost 1998] Scheinost,U., Haas, H., Krause,J., Lindlbauer,J.(Ed.)(1998): Marktanalyse und Marktprognose: das ZVEI Verbandsinformationssystem ELVIRA. Forschungsberichte des Informationszentrum Sozialwissenschaften Band 2. Bonn 1998.

[Sparck Jones 1998] Sparck Jones, K. et al.: A Probabilistic Model of Information Retrieval: Development and Status, Computer Laboratory, University of Cambridge (1998).

[Steffens 1996] Steffens, Ulrike: Integration von Information Retrieval Funktionalität in eine offene, persistente Programmierumgebung. Informatik Mitteilung FBI-HH-M-257/96. Fachbereich Informatik, Universität Hamburg, 1996.

[Steffens 1997] Steffens, Ulrike: Information Retrieval: Grundlagen, Systeme und Integration in: Bibliothek 21/1 1997

[Turtle 1992] Turtle, H.R.; Croft, W.B.: A Comparison of Text Retrieval Models. In: Computer Journal 35 (3) 1992, S. 279-290.

## 6. Internetadressen (URLs)

Die URL des STATLINK-Projekts ist:
http://www.diw-berlin.de/deutsch/projekte/home/statlink/

Die URL für die Internetversion des STATLINK-Web-Clients Statfinder ist:
http://www.diw-berlin.de/deutsch/service/statfinder/

Die Url für den Download des Windows-Clients FAPSY ist
http://www.diw-berlin.de/FAPSY/fapsy.html

Die Url für den Download des Java-Statlink-Clients ist
http://www.diw-berlin.de/deutsch/projekte/home/iuk_statlink/

Teilnahme an der ETK 1999 in Prag:

http://europa.eu.int/comm/eurostat/research/index.htm?http://europa.eu.int/en/comm/eurostat/research/conferences/etk-99/participants/piper_holger.htm&1

http://europa.eu.int/comm/eurostat/research/index.htm?http://europa.eu.int/en/comm/eurostat/research/conferences/etk-99/participants/pauer_bernd.htm&1

Die Url für Informationen zum Datenbestand der Bauvolumensrechnung ist
http://www.diw-berlin.de/deutsch/projekte/home/stt_bauvol/

Die Url für Informationen zum Datenbestand der neuen VGR ist
http://www.diw-berlin.de/deutsch/publikationen/diskussionspapiere/docs/papers/dp206.pdf

Informationen zum Projekt IRAIA:
http://iraia.diw.de