Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Entwicklung eines Data Warehouses zur Durchführung von Zitierungsanalysen

Diplomarbeit

vorgelegt von: Schnerwitzki, Tino
Studiengang: Informatik

Leipzig, Mai 2008

Betreuer: H. Köpcke, Prof. E. Rahm
Institut für Informatik
Abteilung Datenbanken

Vorwort

Zusammenfassung

In vergangenen Publikationen wurden bereits verschiedene Zitierungsanalysen durchgeführt. Jedoch stets auf unterschiedlichen Datenquellen, was einen direkten Vergleich der jeweiligen Ergebnisse verhindert. Ziel dieser Arbeit soll es sein, eine Grundlage zur flexiblen Durchführung von Zitierungsanalysen zu schaffen. Durch die Entwicklung eines Data Warehouses sollen verschiedene Datenquellen integriert und konsolidiert werden, um eine Vielfalt von Analyseperspektiven und Berechnungsverfahren auf einem einheitlichen Datenbestand zu ermöglichen. Dabei wird insbesondere auf die Besonderheiten bei der Nutzung von Webdatenquellen, als wie verschiedene Methoden zur Datenbereinigung eingegangen.

Aufbau der Arbeit

Diese Arbeit besteht neben der Einleitung aus zwei weiteren Teilen. Die *Einleitung* in Kapitel 1 geht auf die Motivation für die Arbeit, die Aufgabenstellung und die grobe Vorgehensweise ein.

Teil I beschäftigt sich mit dem Aufbau des Data Warehouses mit allen verbundenen notwendigen Komponenten und Aktivitäten. Speziell geht Kapitel 2 auf den Aufbau des ETL-Prozesses ein. Dazu gehören die Vorstellung der verschiedenen Komponenten, die zur Erstellung des Data Warehouses eine Rolle spielen, als auch deren Ausführungsreihenfolge. Kapitel 3 und 4 behandeln die Integration der genutzten Datenquellen, wobei Kapitel 4 zusätzlich die Besonderheiten von Webdatenquellen erläutert. Letztlich werden in Kapitel 5 Methoden zur Sicherstellung der Datenqualität vorgestellt.

In Teil II geht es schließlich um den praktischen Einsatz des Data Warehouses. Nachdem in Kapitel 6 auf die Besonderheiten der Ausgangsdaten eingegangen wurde, werden in Kapitel 7 verschiedene Analysen aus unterschiedlichen Blickwinkeln durchgeführt, um ein Bild über das Verhältnis der Bedeutungen ausgewählter Konferenzen und Journale zu erhalten.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Zitierungsanalysen dienen der Bewertung und des Vergleichs des Einflusses verschiedenster wissenschaftlicher Publikationen. Die verwendeten Ideen und Ergebnisse anderer Arbeiten werden innerhalb eines Papiers in Form von Zitierungen angegeben. Am Ende des Papiers erscheint eine Liste der verwendeten Arbeiten in Form eines Literaturverzeichnisses. Oftmals befinden sich innerhalb der Zitierungen auch solche Arbeiten, die sich in verwandten Themengebieten bewegen und so weiterführende Informationen bieten oder die Anwendbarkeit der dargestellten Verfahren in anderen Bereichen demonstrieren.

Die Analyse des Graphs der gegenseitigen Zitierungen stellt somit die Grundlage zum Ermitteln des Einflusses von gewissen Papieren auf die weitere wissenschaftliche Entwicklung dar. Weiter ermöglicht die Gruppierung der Papiere nach ihren Autoren oder den Konferenzen bzw. Journalen in denen sie erschienen sind, ein Messen deren wissenschaftlichen Einflusses. Eine Gruppierung nach Veröffentlichungsjahren kann zur Analyse der zeitlichen Entwicklung herangezogen werden.

Die Verwendung der Zitierungszahlen als Maß des wissenschaftlichen Einflusses wurde erstmals in [Gar55] richtig formuliert. Nachteile dieser Messung ergeben sich einerseits aus der Tatsache, dass nie alle Einflüsse in ein Papier auch durch Zitierungen gewürdigt werden. Vielmehr werden eher Quellen zitiert, die notwendig sind, um die eigenen Ergebnisse zu rechtfertigen. Außerdem werden Zweitquellen, also Arbeiten die die Ergebnisse von anderen weiterverwerten, teilweise bevorzugt zitiert, wodurch die Referenz auf den Ursprung verloren geht. Andererseits spielen zum Teil auch politische Gründe eine Rolle für Zitierungen, um so die Akzeptanz bei den Bewertern und damit die Wahrscheinlichkeit der Veröffentlichung des eigenen Papiers zu erhöhen. Dass diese Art der Bewertung von Artikeln trotzdem aussagekräftig ist, wurde unter anderen in [Gar79] und [BG85] diskutiert.

Grundlage für Zitierungsanalysen bilden so genannte Bibliographiedatenbanken. Allerdings sind diese meist nicht vollständig in Bezug auf die gewünschten Informationen und bieten oft keine direkte Zugriffmöglichkeit, sondern erfordern das Extrahieren der erforderten Daten mittels verschiedener Abfragetechniken. Quellen, die schon aggregierte Werte und Einflusskennzahlen anbieten, wie etwa die Journal Citation Reports [oK06],

berücksichtigen oft nur Journale und ignorieren Konferenzen. Dabei wird sich zeigem, dass gerade Konferenzen in der Informatik einen großen Einfluss besitzen.

Die Bestimmung des Einflusses von Journalen, Instituten, einzelnen Papieren oder ähnlichem, hat deshalb eine große Bedeutung, weil die Kennzahlen oft genutzt werden, um Forschungsgelder zu verteilen. Eine höhere Anzahl von Zitierungen resultiert daher teilweise in einem höheren zur Verfügung stehenden Budgets. Umso wichtiger ist eine qualitativ hochwertige Datengrundlage zur Bestimmung der Kennzahlen, da sonst finanzielle und strategische Entscheidungen fälschlich beeinflusst werden. Eine solche falsche Entscheidung wäre beispielsweise Papiere abzulehnen, die nur aufgrund ihres speziellen Themas nur geringere Zitierungschancen besitzen. Das hätte wiederum zur Folge dass nur noch Mainstreamarbeiten geschrieben und veröffentlicht würden.

Verschiedene Zitierungsanalysen für bestimmte Journale und Konferenzen wurden unter anderen in [RT05] ausgeführt. Eine grundlegende Schwierigkeit besteht immer im Aufbau des Datenbestandes. Darin inbegriffen sind die Integration verschiedenster Datenquellen, meist Bibliographiedatenbanken, und die notwendigen Maßnahmen zur Herstellung einer zufriedenstellenden Datenqualität. Für jede Analyse müssen diese Schritte ausgeführt werden. Damit bauen unabhängige Analysen zum Teil auf unterschiedlichen Datenquellen, auf jeden Fall aber auf unterschiedlich aktuellen Ständen der Datenquellen auf. Die Analysen sind daher nicht miteinander vergleichbar. Eine erneute Ausführung der alten Analyse ist zudem meist schwierig, wenn sich die Datenstruktur geändert hat. Aus diesem Grund bietet sich die *Erstellung eines Data Warehouses zur Durchführung von Zitierungsanalyse* an.

1.2 Aufgabenstellung

Im Mittelpunkt der Arbeit steht die Entstehung eines Data Warehouses, welches die Durchführung verschiedener Anfragen im Rahmen der Zitierungsanalyse ermöglicht. Dabei sollen insbesondere Auswertungen über Zitierungszahlen und Einflusskennzahlen von Publikationen, Publikationsorten wie Konferenzen und Journale, Autoren und weiteren Publikationsattributen durchführbar sein. Daneben sind sowohl gesamtheitliche Betrachtungen, als auch zeitliche Entwicklungen gefordert.

Da die für die Auswertungen notwendigen Informationen über verschiedene Bibliographiedatenbanken verteilt sind, ist eine Integration dieser Datenquellen erforderlich. Besonderes Augenmerk liegt dabei auf Webdatenquellen wie *Google Scholar* [Goo]. Diese unterstützen keinen wahlfreien Zugriff, sondern erfordern die Nutzung spezieller APIs oder die Verarbeitung entsprechender Webseiten. Die Auswahl der Daten erfolgt teilweise über Suchfunktionen, was eine Untersuchung von Suchstrategien, welche die erforderten Daten mit minimalen Kostenaufwand ermitteln, notwendig macht.

Die Qualität der Ergebnisse hängt maßgeblich von der Qualität der Ausgangsdaten ab. Aus diesem Grund sind massive Datenbereinigungen erforderlich. Speziell gilt das für das Auffinden und Beseitigen von Duplikaten innerhalb der einzelnen Datenquellen, als auch von durch die Integration entstandenen Duplikate. Die Beseitigung der Duplikate muss dabei ein Maximum an Informationen erhalten und widersprüchliche Informationen derart auflösen, dass die Qualität erhalten bleibt.

Bei der Konzipierung des Data Warehouses ist darauf zu achten, dass es eine flexible Struktur besitzt. Es soll möglich sein das Datenbankschema zu erweitern, um neue Anfragen zu erlauben. Weiter sollen Änderungen am Erstellungsmechanismus des Data Warehouses durch einen flexiblen modularen Aufbau unterstützt werden. Dies betrifft speziell das Einbinden neuer Datenquellen und den Austausch von Datenqualitätssicherungskomponenten. Außerdem muss das System eine regelmäßige Aktualisierung des Datenbestandes unterstützen, um Analysen stets mit aktuellen Daten durchzuführen.

1.3 Abgrenzung

Verfahren zur Herstellung der Datenqualität werden in dieser Arbeit nur soweit behandelt, wie es für das eigentliche Ziel der Zitierungsanalyse zwingend notwendig ist. Insbesondere werden keine Untersuchungen und Vergleiche zum Thema Duplication Matching angestellt, sondern ein einfaches aber effektives, wenn auch nicht optimales, Verfahren ausgewählt und beispielhaft behandelt.

Ziel ist es auch nicht eine umfassende Untersuchung von Suchstrategien für Webseiten mit Suchfunktion darzustellen. Die genutzten Strategien werden so angelegt sein, eine möglichst hohe Abdeckung bei moderatem Aufwand zu gewährleisten. Verbesserungen in beide Richtungen, sowohl Abdeckung als auch Aufwand, sind noch möglich und werden nur am Rande angesprochen.

Weiterhin wird für das Data Warehouse keine Historie gepflegt werden. Vielmehr wird das Data Warehouse eine regelmäßige Komplettaktualisierung aus dem Datenbestand einer Basisdatenbank unterstützen. Untersuchungen über Änderungen des Datenbestandes werden damit nicht möglich sein.

1.4 Vorteile eines Data Warehouses

Für die gegebene Aufgabe bietet ein Data Warehouse die folgenden Vorteile:

- Aktualität der Daten

- Analysen können ständig mit den aktuellsten Daten wiederholt werden. (Auch automatisiert)
- Integration verschiedenster Datenquellen ist möglich
- Saubere Daten, Vorausgesetzt es wurden entsprechende Säuberungsverfahren bei der Befüllung verwendet.
- performante Ausführung von Analyseabfragen
- einheitliche Struktur für verschiedenste Analysen
- *Single Point of Truth*[1] - Prinzip → ermöglicht konsistente Vergleiche zwischen den Analysen
- Offlineverarbeitung der Anfragen → kein Zugriff auf Webseiten oder ähnliches nach der Datenintegration mehr notwendig

Die *Aktualität der Daten* bezieht sich dabei auf die Möglichkeit, das Data Warehouse ständig aktualisieren zu können. Dafür müssen die Datenquellen, bei Änderungen oder in Zeitintervallen, einfach neu importiert werden. An dieser Stelle wird deutlich, dass der ganze oder zumindest allergrößte Teil des Erstellungs- und Aktualisierungsprozesses automatisiert stattfinden sollte. Ansonsten wäre eine ständige manuelle Pflege der geänderten Daten notwendig. Nach Abschluss eines Aktualisierungsvorgangs können theoretisch alle Analyseanfragen automatisiert erneut ausgeführt und die Ergebnisse an eine geeignete Stelle abgelegt werden. Damit würden Analysen der Ergebnisse stets auf den aktuellsten Daten erfolgen.

Die Möglichkeit verschiedenste Datenquellen integrieren zu können, erfordert einen möglichst modularen Aufbau des Erstellung- bzw. Aktualiserungsprozesses. Nur so ist es möglich nachträglich beliebige Datenquellen beliebiger Struktur zu nutzen. Besonderes Augenmerk liegt dabei auf der Nutzung von Webdatenquellen, da Webdienste wie zum Beispiel `Google Scholar` meist frei verfügbar und nutzbar sind und über einen umfangreichen Datenbestand verfügen. Als Datenbasis soll die `DBLP` Datenbank dienen, welche eine hohe Qualität bietet und für die relevanten Daten nahezu vollständig ist.

Die Nutzung von `DBLP` sorgt zusätzlich für einen hohen Anteil sauberer Daten in den Grunddaten. Weitere Säuberungsverfahren sind daher nur auf den zusätzlichen Datenquellen anzuwenden. Dies umfasst vor allem die Duplikatbeseitigung, da zum einen die

[1] Im Gegensatz zu verteilten Datenbanken, bei der jede ihre eigenen Wahrheiten in Bezug auf bestimmte Anfragen besitzt und so zwangsläufig Widersprüche zwischen den Ergebnissen auftreten, bietet die Integration aller Quellen und Bereinigung der Daten im Data Warehouse eine zentrale Anfragestelle. Diese wird als die einzige Wahrheit angesehen, alle Anfragen müssen damit auf dem Data Warehouse ausgeführt werden.

Datenquellen selbst nicht duplikatfrei, zum anderen die Quellen wechselseitig nicht disjunkt sind.

Die *performante Ausführung von Analyseabfragen* setzt ein geeignetes Schema des Data Warehouse voraus. Dieses muss statt auf transaktionale Nutzung auf Analyseverarbeitung ausgerichtet sein, was eine denormalisierte Haltung der Daten beinhaltet. Außerdem werden mittels gewisser OLAP Tools Daten schon voraggregiert gehalten, um die Berechung der Teilergebnisse während der Abfrageverarbeitung zusätzlich zu beschleunigen. Zusätzlich müssen hier abgeleitete Werte aus den Daten berechnet werden. Dazu zählt beispielsweise, die Eigenzitierungen als solche zu markieren, um diese später bei Analysen wahlweise herauszurechnen, da sie den Einfluss von Papieren nicht widerspiegeln.

Das Data Warehouse soll so konzipiert sein, dass verschiedenste Anfragen möglich sind. Zum einen entfällt damit das jeweilige Sammeln und Integrieren von Daten für eine bestimmte Analyseart, die teilweise für die nächste Analyse unbrauchbar sind. Beispielsweise eignet sich eine Datenbank mit Zitierungen eines Autors nicht zur Einflussanalyse von bestimmten Konferenzen. Das Data Warehouse beinhaltet alle relevanten Daten und ermöglicht die Auswertungen die gerade gewünschte Richtung. Zudem Arbeiten die verschiedenen Abfragen auf dem gleichen zugrunde liegenden Datenbestand und somit untereinander vergleichbar.

Die Nutzung der Webdatenquellen setzt meist eine Onlineabfrage der benötigten Daten und Onlineverarbeitung derselben voraus. Dies ist zum einen recht unflexibel und kann zum andern länger dauern, wenn viele verstreute Daten benötigt werden. Das Data Warehouse beinhaltet die Daten vom Integrationszeitpunkt und ermöglicht damit eine *Offlineverarbeitung*. Ein Nachteil ergibt sich allerdings aus der Methode. Es muss zum Integrationszeitpunkt feststehen, welche Webdaten für die Anfragen benötigt werden, da diese zum Abfragezeitpunkt nicht mehr beeinflusst werden können. Aber auch hier kommt der modulare Aufbau des Aktualisierungsprozesses ins Spiel. Schließlich kann dieser so angepasst werden, dass die notwendigen Daten abgerufen werden. Nach der nächsten Aktualisierung stehen die notwendigen Daten dann zu Verfügung. Allerdings muss dieser Punkt im Kopf behalten werden, um keine statistischen Verzerrungen in den Ergebnissen zu produzieren. So könnte eine Auswertung über alle Autoren verfälscht werden, wenn nur die Webdaten bestimmter Konferenzen und Journale berücksichtigt wurden, einzelne Autoren aber eher in anderen Bereichen aktiv sind und daher schlechte Kennzahlen erhalten. Diese Abfragen müssen dann zwingend in dem Kontext betrachtet werden, d.h. speziell „Wer sind die Top 10 Autoren in den Konferenzen ...?“, statt „Wer sind die Top 10 Autoren?“.

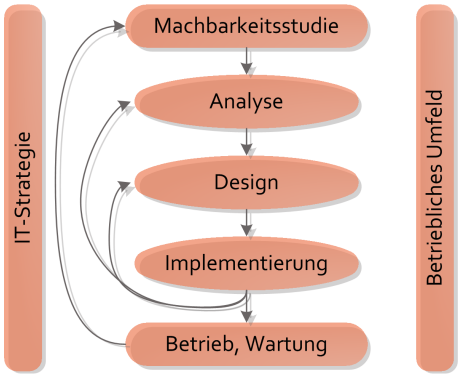


Abbildung 1.1: Projektphasen für Data Warehouse-Projekte, Quelle: [Dit99]

1.5 Vorgehensweise

Wie in allen IT-Projekten, benötigt die auch Erstellung eines Data Warehouses ein geplantes Vorgehen, um der gewünschte Resultat erreichen zu können. Da ein Data Warehouseprojekt in der Regel höchst dynamisch ist, das bedeutet vorläufige Resultate haben einen großen Einfluss auf das Design, wird im Projektverlauf ein hoher Grad an Rückkopplung benötigt. Dies wird durch ein iteratives Projektmodell ermöglicht.

Abbildung 1.1 zeigt ein in [Dit99] vorgeschlagenes Projektphasenmodell für Data Warehouse-Projekte im wirtschaftlichen Umfeld. Dieses soll in leicht abgewandelter Form auch für dieses Projekt verwendet werden. So spielen die IT-Strategie und das Betriebliche Umfeld an dieser Stelle nur eine untergeordnete Rolle. Auf Betrieb und Wartung wird nur minimal eingegangen, zum Beispiel wenn es um Erweiterbarkeit oder Austausch von Teilkomponenten geht.

1.5.1 Machbarkeitsstudie

Zur Machbarkeitsstudie gehört zum einen die Suche nach geeigneter Software zur Verarbeitung der Daten und Ausgabe der Ergebnisse, zum anderen die Suche nach geeigneten Datenquellen, hinsichtlich der Anforderungen.

Softwareseitig bietet sich der *Microsoft SQL Server 2005* als Komplettpaket an, bestehend aus einem SQL Datenbank Server zur Datenhaltung, den Integration Services (SSIS) als ETL-Tool, den Analysis Services (SSAS) als OLAP Tool zur performanten Abfrageverarbeitung und den Reporting Services (SSRS) zur Darstellung und Export der Abfrageergebnisse. Weiteres zur verwendeten Software in Anhang A.

Aus den Anforderungen heraus, steht schon fest, dass als Datenquelle zum einen `DBLP` verwendet werden soll. Diese Datenbank ist frei im Xmlformat verfügbar und wodurch die Nutzung relativ einfach ist. Weiter werden Datenquellen benötigt, die auch Zitierungsinformationen beinhalten. Da kommt zum Beispiel `Google Scholar` als Webseite in Frage, welche entsprechend verarbeitet werden muss. Näher eingegangen wird auf Datenquellen in Abschnitt 2.1, auf die Verarbeitung der Webseiten in Kapitel 4.

1.5.2 Analyse, Design und Implementierung

Die Trennung von Analyse, Design und Implementierung kann oft nur schwer vollzogen werden. So müssen beim Design der Data Warehouses schon Implementierungsdetails beachtet werden, um die Implementierung mit Hilfe der verwendeten Software überhaupt zu ermöglichen. Beispielsweise setzt das verwendete OLAP-Tool bestimmte Eigenschaften des Data Warehouse-Schemas voraus. Obwohl das klassische Data Warehouse-Design für sich korrekt wäre, wäre die Nutzung letzten Endes nicht möglich. Weiter können Erkenntnisse der Design oder Implementierungsphase Auswirkungen auf die Ergebnisse der Analysephase haben, so dass diese teilweise überarbeitet werden müssen. Wenn zum Beispiel beim Verarbeiten der Webseiten, einzelne Attribute gar nicht oder zu mindest nur sehr schwer zu extrahieren sind, müssen Alternativen gefunden und neu analysiert werden. Eventuell müssen sogar die Anforderungen abgeschwächt werden. Umgekehrt gilt natürlich auch, dass wenn sich neue unerwartet leicht zu extrahierende Attribute auftun, können die Anforderungen erweitert und das Design entsprechend angepasst werden.

Diese Rückkopplung geschieht noch innerhalb des äußeren Iterationsprozesses, welcher von einem einfachen Prototypen mit Teilergebnissen zum vollständigen Data Warehouse führt. Aus diesem Grund werden die drei Phasen auch nicht getrennt betrachtet werden. Hauptaugenmerk liegt auf dem Design zum einen des Data Warehouses und seiner Bestandteile selbst, zum anderen des Prozesses zur Erstellung und Aktualisierung desselben. Auf Analyse- und Implementierungsdetails wird an den Stellen eingegangen, an denen dies notwendig ist.

1.5.3 Bottom-Up Vorgehen

Beim Design des Data Warehouses wurde ein *Bottom-Up* Vorgehen [BG04] verwendet. Dies bedeutet zuerst wurde ausgehend von den zur Verfügung stehenden Datenquellen und den Anforderungen an das System die Basisdatenbank entworfen. Erst anschließend das Data Warehouse selbst. Im Gegensatz zum *Top-Down*, bei dem zuerst das Data Warehouse selbst designt wird und dadurch die Basisdatenbank sehr eng auf dieses konzentriert ist, ermöglicht ein *Bottom-Up* Vorgehen das Entwerfen eines allgemeinen modularen und damit leicht erweiterbaren Gesamtsystems. Unter diesem Gesichtspunkt könnte das Vor-

gehen auch mit der *Think Big - Start small* Variante gleichgesetzt werden, da das Design zwar sofort auf das zu erzielende Endprodukt ausgerichtet ist, aber einer späteren Erweiterung nichts im Wege steht.

Das Prinzip des *Bottom-Up* Vorgehens zieht sich natürlich über alle drei Entwicklungsphasen und steht damit als ständiges Prinzip über dem iterativen Entwicklungsprozess. Innerhalb einer Iteration muss der Unterbau noch nicht vollständig implementiert werden, solange nur der Designfluss entsprechend von Basisdatenbank in Richtung Data Warehouse zeigt. Trotzdem können zwischen Iterationen einzelne Designbestandteile verfeinert, Komponenten hinzugefügt, ersetzt oder gar wieder entfernt werden. In dieser Arbeit wird nur das Ergebnis der letzten Iteration veranschaulicht werden.

Teil I

Aufbau des Datewarehouses

2 Der ETL-Prozess

Da das Erstellen eines Data Warehouses mit vielen verschiedenen Schritten verbunden ist, wird die Aufbereitung der verschiedenen Datenquellen in einem Arbeitsbereich (engl. staging area) vorgenommen. Dort wird für jede Datenquelle je eine Extraktions-, eine Transformations- und eine Ladekomponente benötigt, um die Daten für das Zielsystem aufzubereiten [BG04].

Die Extraktionskomponente ist für das Extrahieren der Daten aus der Datenquelle in eine einfach zu verarbeitende Form verantwortlich. Dies kann das Auslesen von Xml- oder CSV-Dateien sein oder das Abrufen von Daten aus anderen Datenbanken mittels SQL-Abfragen.

Die Transformationskomponente mappt unterschiedliche Attribute in Quelle und Ziel aufeinander. Meist kann dies nicht eins-zu-eins geschehen. So müssen zum Beispiel zusammengesetzte Namen in Vor- und Zunamen zerteilt und entsprechend abgelegt werden. Außerden müssen unterschiedliche Kodierungen und Datentypen in Quelle und Ziel ausgeglichen werden.

Die Ladekomponente importiert die aufbereiteten Daten in die Datenbank des Arbeitsbereichs, wo später die zur Analyse erforderlichen Daten extrahiert und ins Data Warehouse geladen werden.

Die drei beschriebenen Komponenten müssen nicht immer so klar getrennt sein. So bezeichnen sich die Microsoft SQL Server Integration Services als ETL-Tool, womit alle drei Schritte zusammen realisiert werden können. Die Trennung ist vielmehr logischer Natur. Daher wird das ganze Vorgehen auch als ETL-Prozess bezeichnet.

2.1 Auswahl der Datenquellen

Da es zur Zitierungsanalyse keine einzelne Datenquelle gibt, die schon alle benötigten Informationen beinhaltet, ist es notwendig verschiedene Datenquellen zu integrieren. Zur Auswahl der richtigen Datenquellen werden in [BG04] vier wichtige Kriterien aufgelistet:

- Der Zweck des Data Warehouses

- Die Qualität der Quelldaten
- Die Verfügbarkeit (rechtlich, sozial, organisatorisch, technisch)
- Der Preis für den Erwerb der Quelldaten

Das Kriterium Preis spielte für dieses Data Warehouse eine untergeordnete Rolle, da sowieso nur frei verfügbare Datenquellen in Betracht gezogen wurden. Ebenso sind die soziale und organisatorische Verfügbarkeit in diesem Fall irrelevant, sie spielen nur für betriebliche Einsätze eine Rolle. Die folgenden Datenquellen wurden für die Auswertungen im Teil II genutzt:

- DBLP [Ley]
- Google Scholar [Goo]
- ACM Portal [fCM]
- Citeseer [Cit]

Bei `DBLP` handelt es sich um eine handgepflegte Datenbank von wissenschaftlichen Publikationen. Durch diese manuelle Verwaltung der Daten, besitzt `DBLP` ein hohes Maß an Qualität. Die Datenbank umfasst aktuell fast eine Million Publikationen und dazu jeweils verschiedene Attribute wie Autoren, Erscheinungsjahr und Konferenz bzw. Journal in dem sie erschienen ist. Dazu noch gibt es noch weitere Attribute, die an dieser Stelle nicht weiter relevant sind. Die Datenbank ist komplett als Xmldatei verfügbar[1] und wird ständig aktualisiert.

`Google Scholar` und `Citeseer` sind Suchmaschinen für wissenschaftliche Publikationen. Beide beinhalten zusätzlich zu den Publikationsinformationen auch Daten darüber welchen anderen Publikationen sie zitiert wurden. Während `Google Scholar` ausschließlich über die Suchfunktion abgefragt werden kann, gibt es zu `Citeseer` einen Xmldump der Webdaten.

In `ACM Portal` werden alle Publikationen, geordnet nach Konferenzen, Journals oder andere, aufgelistet, die von `ACM` veröffentlicht wurden. Auch hier sind Informationen über zitierende Papiere verfügbar, allerdings müssen auch diese bei `ACM` veröffentlicht worden sein. Daher sind die bei `ACM` auch kleinere Zitierungszahlen, als bei den anderen beiden zu erwarten. Dafür ist die Liste annähernd vollständig und von relativ hoher Qualität. Aber auch für `ACM Portal` gibt es keine Möglichkeit eine komplette Datenbank zu erhalten, es müssen wie bei `Google Scholar` die Webdaten abgegriffen werden.

[1] `http://dblp.uni-trier.de/xml/`

Alle vier Datenquellen erfüllen das erste Kriterium. Der Zweck des Data Warehouses ist es Zitierungsanalysen durchzuführen. Während DBLP zwar keinerlei Zitierungsinformationen enthält, ist es fast vollständig bezüglich wissenschaftlicher Papiere im Bereich der Informatik. Durch die hohe Verfügbarkeit, Qualität und Aktualität eignen sich die DBLP Daten sehr gut als Datenbasis.

Die Qualität der anderen Datenquellen liegt unter der von DBLP. Bei Google Scholar und ACM liegt dies vor allem an Schwierigkeiten bei der Datenextraktion, siehe dazu Kapitel 4. Bei Citeseer fehlen teilweise sogar wichtige Attribute wie Konferenz bzw. Journal.

Die Verfügbarkeit ist bei allen Quellen prinzipiell gewährleistet. Allerdings kann es rechtlich problematisch sein, Webdaten automatisiert herunterzuladen. So bittet ACM in den Nutzungsbedingungen darum, keine intelligenten Agenten einzusetzen, um Artikel herunterzuladen. Das Durchsuchen der Webseite wird allerdings nicht explizit ausgeschlossen. Noch kritischer sieht es Google Scholar. Hier sollen Daten nur unverändert dem Nutzer angezeigt und dürfen nicht zwischen gespeichert werden. Allerdings wird auch hier die Nutzung von Robotern und sofortige Extraktion der Daten nicht explizit ausgeschlossen. Außerdem bezieht sich das Verbot der Nutzung eines Caches für die Webseiten, nur auf die Bereitstellung des Dienstes für andere Nutzer auf der eigenen Webseite. Das Herunterladen der Webseiten und das offline weiterverarbeiten fällt streng genommen nicht unter diese Definition. So lange die Informationen also nicht kommerziell genutzt oder verfälscht und immer nur mit Quellenangabe dargestellt werden, sollte die Verwendung unproblematisch sein.

Weitere Datenquellen die in Betracht gezogen wurden sind unter anderen Live Search Academic [Mic], The Collection of Computer Science Bibliographies [CSB] und Scopus [Sco]. Die ersten beiden enthalten keinerlei Zitierungsinformationen und auch sonst keine zusätzlichen gewinnbringenden Daten für das Data Warehouse. Scopus ist dagegen nicht frei verfügbar und scheidet daher aus.

2.2 Aufbau des ETL-Prozesses

Während der Erstellung und Befüllung des Data Warehouses, werden zwei relationale Datenbanken verwendet. Die Basisdatenbank wird direkt aus den Datenquellen befüllt. Die Basisdatenbank ist weitgehend normalisiert und enthält keine redundanten Daten. Sie ist nicht auf Performanz für Analyseabfragen optimiert. Außerdem hält die Datenbank alle relevanten Daten, die aus den Datenquellen gewonnen wurden, sofern sie nicht durch Duplikate in den verschiedenen Quellen zusammengefasst wurden.

Die zweite Datenbank ist das Data Warehouse. Hier werden nur die zur Analyse notwendigen Daten gespeichert und für die spätere Analyse optimiert. Das Data Warehouse

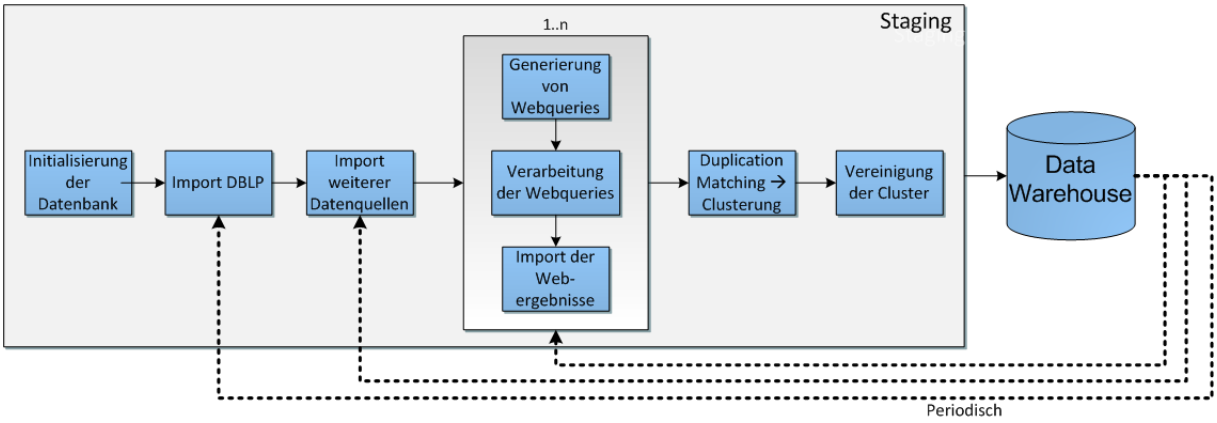


Abbildung 2.1: grafische Übersicht über den gesamten ETL-Prozess

wird bei jedem Abschluss eines ETL-Vorgangs mit der Basisdatenbank synchronisiert, so dass die gleichen Daten zugrunde liegen.

Der ETL-Prozess umfasst nur das Befüllen der Basisdatenbank. Für jede Datenquelle ist das Ausführen von mindestens einer Komponente notwendig. Die verschiedenen Komponenten werden sequentiell ausgeführt und bauen damit aufeinander auf. Abbildung 2.1 stellt den Grobaufbau des gesamten Prozesses schematisch dar.

Die Abbildung zeigt den Stagingbereich und wie das Ergebnis, die Datenbank in das Data Warehouse geladen wird. Im Stagingbereich wird ausschließlich die Basisdatenbank verwendet. Vor dem Befüllen des Data Warehouses wird dessen zugrunde liegende relationale Datenbank komplett geleert. Das vereinfacht den Abgleich mit der Datenbank und ein Erhalt der Daten wäre nur für die Pflege einer Historie notwendig, die aber ausgeschlossen wurde. Die gestrichelten Linien skizzieren, an welchen Stellen der ETL-Prozess nach der Erstausführung periodisch neu gestartet werden kann. Dabei soll es so sein, dass die innere Schleife in regelmäßigen Abständen, zum Beispiel wöchentlich, und automatisch ausgeführt wird. Wie sich später noch zeigen wird, können damit jeweils unterschiedliche Teilmengen der Webdatenquellen aktualisiert werden, wodurch die Last gleichmäßiger verteilt wird. Die äußere Schleife kann dagegen sporadisch manuell laufen, da sich diese Quellen nicht so oft ändern und durch die höhere Datenmenge auch eine höhere Last ausgelöst wird. Außerdem ziehen sie das Neuabfragen einiger Webdatenquellen nach sich.

Die *Initialisierung der Datenbank* wird nur beim erstmaligen Erstellen der Datenbank durchgeführt. Dabei wird die Basisdatenbank komplett geleert. Somit ist es zum Beispiel möglich die Datenbank komplett neu aufzusetzen, wenn sie aufgrund des Imports einer unsauberen Datenquelle verunreinigt wurde und die unsauberen Daten nicht mehr auf

einfache Weise herausgefiltert werden können. Außerdem werden vorhandene Stammdatentabellen, wie die Ländertabelle, mit Werten befüllt.

Im nächsten Schritt *Import DBLP* wird die aktuelle Version der `DBLP` Datenbank eingelesen. Wie im Abschnitt 2.1 angeführt, besitzt `DBLP` eine hohe Qualität und dient daher als sehr gute Datenbasis. Die späteren Analysen werden sich vor allem auf Publikationen bestimmter Konferenzen und Journals beziehen, die in `DBLP` vollständig eingepflegt sind. Die anderen Datenquellen werden hauptsächlich die noch fehlenden Daten ergänzen. Am wichtigsten sind hierbei die Zitierungsinformationen, aber auch Informationen über das Institut oder das Land in dem das Papier erstellt wurde oder von welchem Typ das Papier ist.

Beim *Import weiterer Datenquellen* werden Quellen eingelesen, die in kompletter Form vorliegen oder zumindest einen direkten wahlfreien Zugriff ermöglichen. In diesem Fall ist vor allem `Citeseer` gemeint, dessen Datenbankdumps frei verfügbar sind und so komplett importiert werden können. Auch in diesen Schritt fällt eine Exceldatei, die im Artikel [RT05] als Datengrundlage diente. Sie liefert manuell zusammengetragene Informationen über die Papiere der analyserelevanten Konferenzen und Journalen von 1994 bis 2003, die für einen Vergleich mit der damaligen Analyse notwendig sind.

Anschließend erfolgt die Webdatenintegration. Dieser Schritt wird für jede Webdatenquelle und Abfragestrategie getrennt ausgeführt. Er besteht aus drei Teilschritten. Als erstes erfolgt die *Generierung von Webqueries*. Hier werden je nach Abfragestrategie Daten aus der Datenbank abgerufen und und zur Webabfrage aufbereitet. Je nach Beschaffenheit der Webdatenquelle, kann dieser Schritt auch entfallen, wenn die Abfragen unabhängig von den `DBLP` Inhalten sind. Anschließend wird die Webdatenquelle abgefragt und die Ergebnisse wiederum in aufbereiter Form abgelegt. Diese Ergebnisse werden zum Schluss in die Datenbank importiert. Auf die Webdatenintegration wird in Kapitel 4 auführlicher eingegangen.

Bis zu diesem Punkt wurden noch keinerlei Objectmatchingverfahren ausgeführt. Natürlich enthalten die verschiedenen Datenquellen keine disjunkten Inhalte, schließlich sollen sie `DBLP` ergänzen, daher liegen an dieser Stelle einige Duplikate in der Datenbank vor. Zum einen die qualitativ hochwertigen `DBLP` Publikationen, zum andern die `Google Scholar`, `ACM` und `Citeseer` Publikationen, welche in den Titeln, Autoren und anderen Attributen Fehler aufweisen. Daher werden die Daten nun geclustert und die Duplikate anschließend jeweils vereinigt. Wäre `DBLP` vollständig und das Clusteringverfahren perfekt, würden auf diese Weise nur die um zusätzliche Informationen angereicherten `DBLP` Publikationen übrig bleiben. In der Realität bleiben jedoch einige Publikationen mehr übrig.

Nach Abschluss der Arbeit im Stagingbereich, enthält die Basisdatenbank die aktuellen Daten. In [BG04] wird zwar empfohlen, dass Staging- und Basisdatenbank getrennt gehalten werden sollten und die Daten aus dem Stagingbereich in die Basisdatenbank

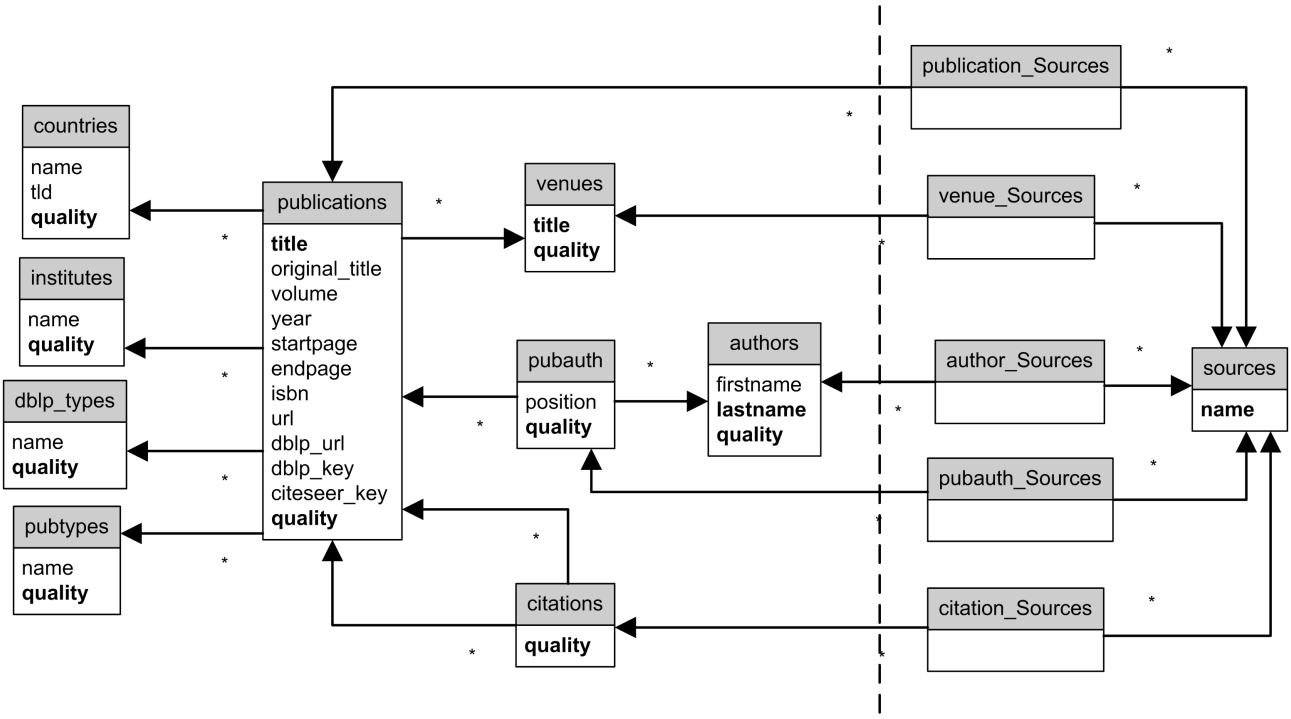


Abbildung 2.2: relationales Schema der Basisdatenbank

übernommen werden sollten, doch hier wäre dies unvorteilhaft. Zum einen werden innerhalb des Stagings alle Daten der Basisdatenbank benötigt, zum andern müssen im Zweifelsfall alle Daten in der Basisdatenbank aktualisiert werden. Die Daten müssten demnach sowohl vor als auch nach dem Arbeiten komplett transferiert werden. Sollen die Daten ín der Basisdatenbank manuell bearbeitet werden, muss demnach darauf geachtet werden, dass gleichzeitig kein ETL-Prozess arbeitet. Oder die manuellen Daten werden extern gepflegt und können dann in den ETL Prozess als weitere Datenquelle aufgenommen werden.

2.3 Datenbankschemata

Im vergangenen Abschnitt wurden die beiden verwendeten Datenbanken angesprochen. Im folgenden sollen ihre Schemata gezeigt und erklärt werden.

2.3.1 Basisdatenbank

Abbildung 2.2 zeigt das relationale Schema der Basisdatenbank in UML-Notation. Primär- und Fremdschlüsselspalten wurden aus Gründen der Übersichtlichkeit ausgeblendet. Die zentralen Tabellen sind hier `[dbo].[publications]`, `[dbo].[venues]`, `[dbo].[authors]` und `[dbo].[pub_types]`, sowie die dazugehörigen M:N-Tabellen und die spezielle M:N-Tabelle `[dbo].[citations]`. Diese Tabellen beinhalten die Nutzdaten der Datenbank.

Demgegenüber steht die Tabelle `[dbo].[sources]` und die entsprechenden M:N-Tabellen zu den Zentraltabellen. Diese Daten sind Auditinformationen. So kann im Nachhinein noch nachvollzogen werden, aus welchen Datenquellen bestimmte Datensätze stammen. Da Duplikate im letzten Schritt des ETL-Prozesses zusammengefasst werden, können einem Datensatz mehrere Datenquellen zugeordnet sein.

Auch bei den Attributen muss zwischen Nutzinformationen und Metainformationen unterschieden werden. Die `quality` Attribute geben Aufschluss über die Qualität des Datensatzes. So besitzen `DBLP` Datensätze eine höhere Qualität als `Google Scholar` Datensätze. Weiter bietet das Attribut `dblp_type` und die entsprechende `dblp_types`-Tabelle Aufschluss über die Typzuordnung einer Publikation in `DBLP`, zum Beispiel „inproceeding“ oder „article“. Da es sich bei den Primärschlüsseln um surrogate keys handelt, helfen die Attribute `dblp_key` und `citeseer_key` später einen einfachen Differenzableich mit den entsprechenden Datenquellen zu ermöglichen. Aus Performanzgründen liegen die beiden Attribute denormalisiert vor. Würden noch weitere Datenquellen mit eigenem Primärschlüssel aufgenommen werden, müssten die Attribute normalisiert in eine 1:N-Tabelle ausgelagert werden. Allerdings besitzen die Webdatenquellen keine eigenen eindeutigen Schlüssel, so dass es nur bei diesen beiden Schlüsseln bleibt.

Die zentrale Tabelle der Datenbank ist `[dbo].[publications]`. Bis auf das Attribut `title` sind alle Attribute optional. Allerdings beinhaltet das Attribut `year` die zweitwichtigste Information. Zusammen mit den Autoren können damit die meisten Publikationen eindeutig identifiziert werden. Die Nennungsreihenfolge der Autoren einer Publikation wird durch das `position` Attribut der M:N-Tabelle `[dbo].[pubauth]` festgelegt.

Die für den Zweck des Data Warehouses ebenso wichtige Tabelle, ist `[dbo].[citations]`. Sie bildet die gerichtete binäre Relation von Zitierungen ab. Dabei wird zwischen zitierten und zitierenden Publikationen unterschieden.

2.3.2 Data Warehouse

Der relationale Entwurf der dem Data Warehouse zugrunde liegenden Datenbank, orientiert sich sehr stark am Verwendungszweck der Analyse. Die Datenbank ist sehr stark de-

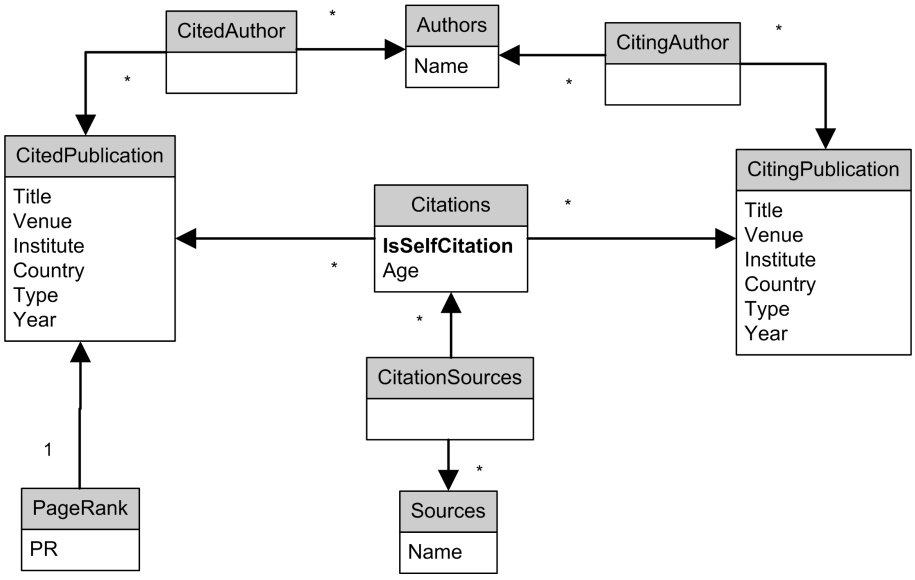


Abbildung 2.3: relationales Schema des Data Warehouses

normalisiert und bildet ein Stern Schema [CD97], wobei die `Citations` und die `PageRank`-Tabelle die Faktentabellen darstellen. Einzig die Autoren- und Quellentabellen erinnern aufgrund ihrer M:N Tabellen an ein Schneeflockenschema. Dies ist aber kein Widerspruch, kennt man die interne Arbeitsweise des verwendeten OLAP Tools *MS SQL Server Analysis Services*. Hier werden die M:N-Tabellen ebenfalls als Faktentabellen angesehen und anschließend als normale Dimensionstabelle an die Zitierungs- bzw. Publikationstabellen gebunden. Nur so ist eine korrekte Berechnung der Faktenzahlen bei Co-Autoren oder mehreren nicht disjunkten Quellen möglich.

In Abbildung 2.3 wird das Schema abgebildet. Die Tabellen `CitedPublications` und `CitingPublications` bilden die Dimensionstabellen für zitierte bzw. zitierende Publikationen. Obwohl die Inhalte der beiden Tabellen nicht disjunkt sind, ist eine Trennung notwendig. Wieder aufgrund der internen Arbeitsweise von SSAS, das verschiedene Dimensionen auf der gleichen Faktentabelle auch verschiedene Dimensionstabellen abbilden muss.

Die Datenquellenangaben wurden aus der Basisdatenbank nur für die Zitierungsinformationen übernommen. Auf diese Weise können die Zitierungszahlen der verschiedenen Quellen miteinander verglichen werden. In den Dimensionen wird die Unterscheidung für die Analyse nicht benötigt, daher entfallen sie dort.

Das Attribut `IsSelfCitation` ist ein berechnetes Feld. Es wird auf 1 gesetzt, wenn sich die Autoren der zitierten und der zitierenden Publikation überschneiden, sonst auf 0. `Age`

ist ebenfalls berechnet und gibt die Differenz zwischen den Erscheinungsjahren der beiden Publikationen an. Im OLAP Würfel werden dann die beiden Attribute und zusätzlich die Summe der Zeilen als Measures definiert.

In die `CitedPublications` Tabelle werden nur die Publikationen übernommen, die laut Basisdatenbank mindestens einmal zitiert wurden. Umgedreht gilt dies auch für die `CitingPublications` Tabelle. Da die Webabfragen nur einen Teil der Publikationen berücksichtigen, beinhaltet das Data Warehouse nur Analyserelevante Daten und ist so noch performanter.

Neben dem Namen in den analog aufgebauten Publikationstabellen, gibt es weitere Attribute. Sie bilden untereinander unhabhängige Hierarchien der jeweiligen Dimension. Damit können zum Beispiel Vergleiche unterschiedlicher Jahrgänge in einer oder verschiedenen Konferenzen angestellt werden.

2.3.3 Multidimensionales Schema

Auf der Basis des relationales Schemas, kann ein multidimensionales Schema entworfen werden. Dabei wird zwischen Measure-Gruppen und Dimensionen unterschieden. Die Measure-Gruppen entsprechen grob gesagt den möglicherweise verschiedenen Faktentabellen. Sie bestehen wiederum aus einzelnen Measures, das sind die Attribute der Faktentabelle zusammen mit einer Definition wie die Attributwerte aggregiert werden. Die Dimensionen entsprechen den verschiedenen Dimensionstabellen. Im Falle des Sternschemaaufbaus, besteht eine Dimension aus genau einer Dimensionstabelle. Die Dimensionen bestehen wieder aus Hierarchien. Ein Element der feingranularsten Ebene einer Dimensionsierarchie muss genau einer Zeile in der Dimensionstabelle entsprechen. Ein Element der Faktentabelle darf auf höchstens ein Element der feingranularsten Ebene je Dimension verweisen. Dies wird durch die Fremdschlüsselbeziehungen im relationalen Schema sichergestellt. Eine Ausnahme bilden die M:N-Relationen zwischen Fakten- und Dimensionstabellen. Hier werden so genannte *Intermediate Measure Groups* [SHW+06] erstellt, womit die Dimensionen entsprechend eindeutig aufgelöst werden können.

Abbildung 2.4 zeigt das multidimensionale Schema entsprechend dem relationalen aus dem letzten Abschnitt. Die Würfel sollen die `Citations`- und die `PageRank`-Measuregruppe darstellen. Die im letzten Abschnitt angesprochenen *Intermediate Measure Groups* sind nicht abgebildet, da sie nur eine interne Verwendung finden, für Abfragen aber irrelevant sind. `Citations` besteht aus verschiedenen Measures. `Citation Count` ist als $Count(*)$ realisiert. Das gleiche Ergebnis könnte man erzielen, würde eine zusätzlichen Attribut eingeführt werden, welches immer den konstanten Wert 1 besitzt und darüber die Summenfunktion als Aggregationsfunktion definiert ist. Das Measure `Self Citation Count` ist als Summe über das gleichlautende Attribut definiert. Bei `AvgAge` wird als Aggregationsfunktion dagegen der Durchschnitt über das Zitierungsalter verwendet. Das Measure

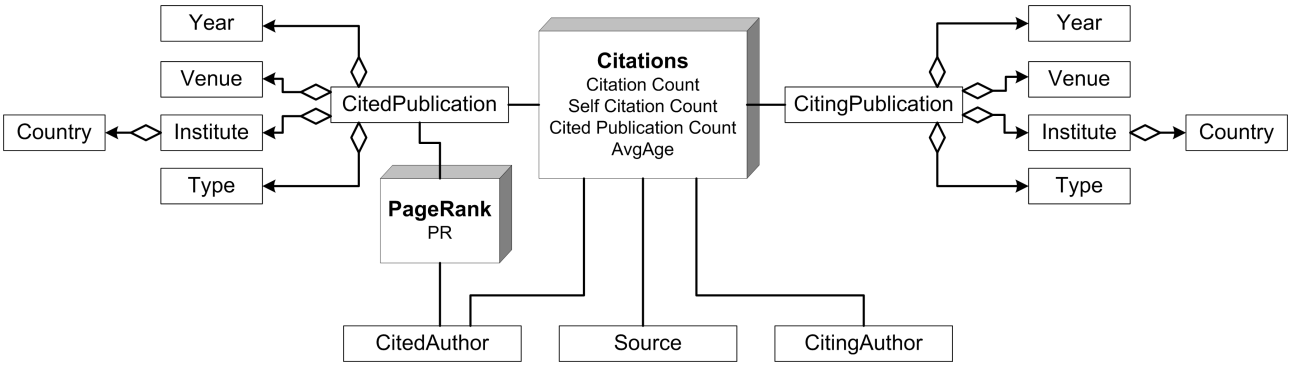


Abbildung 2.4: multidimensionales Schema des Data Warehouses

`PR` in `PageRank` verwendet wieder die Summenfunktion. Dazu kommen noch abgeleitete Measures, so genannte *Calculated Members*, um die häufig genutzten Konstrukte abzukürzen. Dazu gehört zum Beispiel `Non Self Citation Count`, welches als Differenz zwischen `Citation Count` und `Self Citation Count` definiert ist.

Auf der Seite der Dimensionen gibt es fünf Stück. Die `Cited`... bzw. `Citing`... Dimensionen sind symmetrisch aufgebaut. Die `Publications`-Dimensionen sind in verschiedene Hierarchien unterteilt, auf dessen untersten Ebene die entsprechende Publikation steht. Die Dimensionen sind voneinander unabhängig, da zum Beispiel eine Konferenz in verschiedenen Jahrgängen statt fand, andererseits in verschiedenen Jahrgängen auch unterschiedliche Konferenzen statt fanden.

Einzig die Hierarchie `Country-Institute` besitzt mehr als zwei Ebenen. Aus dem relationalen Schema der Basisdatenbank geht dies zwar nicht hervor, dies liegt allerdings nur an fehlenden Informationen. So ist das Land oftmals leichter (eindeutig) herauszufinden als das Institut an dem die Arbeit veröffentlicht wurde. Daher fehlt die Institutangabe öfter. Bei entsprechender relationalen Struktur könnte dies aber nicht abgebildet werden. Daher existieren dort Institut und Land nebeneinander. Dass es hier nicht zu Widersprüchen kommt, müsste eine Konsistenzbedingung erzwingen.

Normalerweise müssten auch die Autorendimensionen entsprechend als Hierarchieebenen definiert werden. Hier hat aber das Wissen über die spätere Implementation einen entscheidenden Einfluss auf das Design des Schemas. M:N-Beziehungen werden in den *Analysis Services* erst ab Version 2005 unterstützt. Aber auch hier können sie nur direkt auf die Faktentabelle angewendet werden. Jedes Element einer Dimensionshierarchiestufe darf höchstens einem Element der grobgranulareren Ebene zugeordnet werden. Da eine Publikationen im Allgemeinen von mehreren Autoren verfasst wurde, wäre diese Bedingung verletzt. Die Auswertung nach allen Publikationen eines bestimmten Autors ist durch spezielle `MDX`-Abfragen aber immer noch möglich, wenn auch etwas komplizierter.

Zusammen bilden die abgebildeten Measuregruppen und die Dimensionen einen OLAP-Würfel. Da schon verschiedene Scheiben des Würfels voraggregiert gespeichert werden, werden viele Anfragetypen schnell und effizient ausgeführt.

3 Integration von Datenquellen mit wahlfreiem Zugriff

In Kapitel 2 Abschnitt „Datenquellen“ wurden zwei Arten von Datenquellen vorgestellt, die für dieses Data Warehouse relevant sind. Auf der einen Seite Datenquellen auf die komplett wahlfrei zugegriffen werden kann, auf der anderen Webdatenquellen die nur durch das Abrufen verschiedener Webseiten einlesbar sind. Dieses Kapitel beschäftigt sich mit der Integration der ersten der beiden Arten.

Zu dieser Kategorie gehören `DBLP` [Ley] und `Citeseer` [Cit]. Auf verschiedene kleinere Quellen die ausschließlich für vergleichende Analysen verwendet wurden, wird an dieser Stelle nicht eingegangen. Beide Quellen sind als Dateien im Xmlformat verfügbar, die so vollständig importiert werden können. Wobei das Format der `DBLP`-Xmldatei durch eine nicht eindeutige DTD definiert wird und das verwendete ETL-Tool `SSIS`[1] damit Probleme hat die Datei korrekt zu importieren. Hier war die Entwicklung einer eigenen SSIS-Komponente notwendig.

Auf `Citeseer` sind zwei verschiedene Xmlformate verfügbar. Das erste entspricht dem Standardformat `OAI`[2] [LdS06]. Dieses Format definiert eine Art Austauschkatalog für digitale Objekte jeglicher Art. Leider sieht das Format keinerlei Beziehungen zwischen diesen Objekten vor, speziell keine Zitierungsinformationen zwischen Publikationen. Daher ist dieses Xmlformat an dieser Stelle unbrauchbar. Das zweite Xmlformat entspricht `OAI` mit zusätzlichen proprietären `Citeseer`-Erweiterungen.

3.1 Qualität

`DBLP`, erstellt und gewartet von der Universität Trier, wurde und wird manuell gepflegt. Schleichen sich doch Fehler ein und werden von Nutzern bemerkt, können diese den Verantwortlichen das mitteilen und die Fehler werden dann beseitigt. D.h. `DBLP` ist kein geschlossenes System, daher ist die Qualität entsprechend sehr gut.

Bei `Citeseer` dagegen handelt es sich eher um eine Blackbox. Es kann für Außenstehende

[1] Microsoft SQL Server Integration Services
[2] Open Archive Initiative

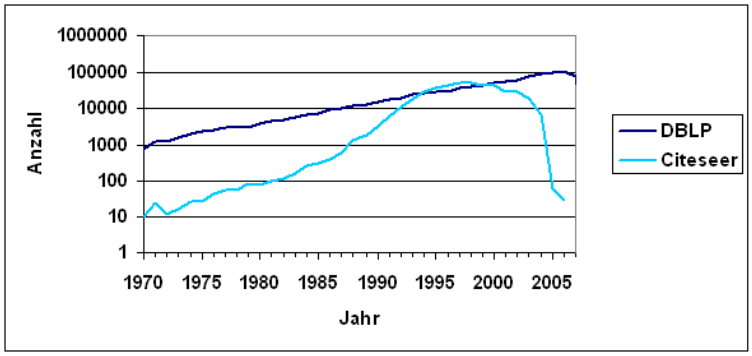


Abbildung 3.1: Vergleich der gelisteten Publikationen in DBLP und Citeseer. Die Y-Achse ist logarithmiert.

nicht so recht nachvollzogen werden, woher die Daten stammen und wie sie zusammengetragen wurden. Trotzdem weist auch `Citeseer` eine vergleichsweise hohe Qualität auf. Allerdings fehlen teilweise wichtige Attribute, wie Konferenz oder Journal der Veröffentlichung eines Papiers. Andererseits bietet es mit den Zitierungsangaben auch zusätzliche Informationen gegenüber `DBLP`, so dass die Nutzung dennoch gewinnbringend ist.

Leider scheint sich proprietäre `Citeseer` Format noch in der Betaphase zu befinden, obwohl jegliche Hinweise darauf fehlen. Aber während das `OAI`-Format xmlkonform ist, verstößt dieses gleich gegen mehrere Xmlstandards (zum Beispiel [W3C06]). So besitzt das Dokument keinen eindeutigen Xmlwurzelknoten, Attributwerte beinhalten teilweise selbst Anführungszeichen und wurden nicht durch „&*quot;*“ ersetzt. Teilweise wurden auch schließende Tags einfach ausgelassen oder es fehlen im Standard vorgeschriebene Attribute. Dazu kommt noch eine falsche Kodierung. In der Xmlspzifikation ist vorgeschrieben, dass keine Zeichen mit einem Code unter $0x20$, außer $0x09$, $0x0A$ oder $0x0D$ in der Datei auftreten dürfen. Sie können entsprechend durch „&*#x<Code>*“ ersetzt werden. Auch dagegen verstößt die `Citeseer`-Xmldatei. Aus diesem Grund muss sie erst repariert werden, wodurch ein Informationsverlust auftritt. Ca. 3% der Xmldateien im Archiv können nicht repariert werden und gehen somit komplett verloren.

3.2 Aktualität

Die `DBLP`-Datenbank wird ständig aktualisiert. Im Juni 2007 waren darin noch ca. 890.000 Publikationen gelistet, im Februar 2008 schon fast eine Million. Dieser große Anstieg in der relativ kurzen Zeit geht einerseits auf die neu veröffentlichten Papiere zurück. Andererseits werden auch ständig neue Listen aufgenommen, welche teilweise auch ältere Publikationen beinhalten.

Abbildung 3.1 zeigt einen Vergleich der Anzahl der gelisteten Publikationen in DBLP und Citeseer. Daran ist klar zu erkennen, dass Citeseer die meisten Publikationen für Mitte der 90er Jahre gelistet hat und dort sogar leicht über DBLP liegt. Bei älteren Publikationen und jüngeren Jahrgängen liegt es aber Größenordnungen darunter. Gerade die jüngeren Jahrgänge, ab 2001, zeigen sehr deutlich, dass Citeseer kaum aktuelle Daten beinhaltet. In 2004 werden nur noch 6942 Einträge angezeigt, gegenüber fast 86.000 in DBLP. In den beiden Jahren darauf sind es nur noch 60 bzw. 27. Bei diesem Vergleich muss allerdings mit bedacht werden, dass in DBLP für ca. 98% der Datensätze eine Jahresangabe existiert, bei Citeseer sind es nur ca. 66%. Die restlichen 33% tauchen daher im Diagramm gar nicht auf und die Kurve müsste demnach etwas höher liegen.

Obwohl Citeseer noch weiterentwickelt wird, was der Blick auf Citeseer^x Beta[3] beweist, scheinen die Daten nicht mehr aktualisiert zu werden. Dies muss bei der Analyse beachtet werden, um keine falschen Rückschlüsse, auf zum Beispiel zurückgehende Zitierungszahlen, zu ziehen.

[3] http://cs1.ist.psu.edu:8080/acksearch/

4 Webdatenintegration

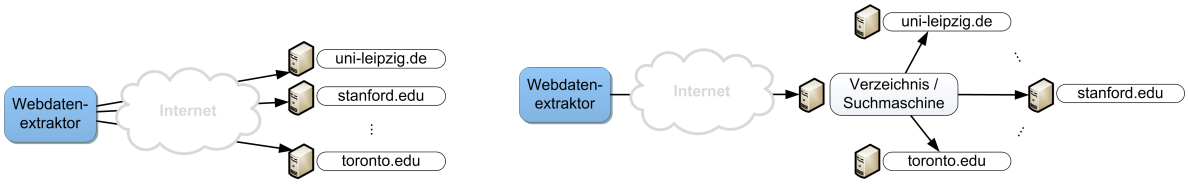
Die größte Schwierigkeit bei der Integration von Webdaten liegt in der Datenextraktion. Die Daten liegen zum Teil verteilt auf viele Webseiten vor und sind dort vermischt mit Stilinformationen innerhalb einer HTML-Struktur. Fehlende semantische Informationen erschweren das Auffinden der relevanten Daten, die Verknüpfung mit anderen Objekten und die Zuordnung von Textteilen zu Attributen zusätzlich.

Im Rahmen der Erstellung des Data Warehouses wurde ein Tool entwickelt, welches genau diese Aufgaben sehr flexibel erledigt und für jede neue Webdatenquelle einfach anzupassen ist. In diesem Kapitel wird das grundsätzliche Vorgehen beim Integrieren von Webdaten mithilfe dieses Tools erläutert und dabei auf Problempunkte genauer eingegangen.

4.1 Crawling vs. Searching

Im Grunde gibt es zwei Herangehensweisen um Daten über verschiedene Objekte aus dem World Wide Web zu sammeln (z.B. [BYRN99]). Die erste Möglichkeit wäre, alle Objekte an ihrem Entstehungspunkt abzurufen, d.h. auf dem Server auf sie veröffentlicht wurden. Um diese Webseiten zu finden wird Crawling eingesetzt. Dabei werden die Verknüpfungen von Webseiten zu anderen verfolgt. An den Endpunkten des Crawlinggraphs befänden sich die wissenschaftlichen Publikationen, welche interpretiert und die Ergebnismenge aufgenommen werden würden. In Abbildung 4.1a wird das Zugriffsmuster dargestellt.

Der Nachteil dieser Vorgehensweise liegt zum einen in der riesigen abzurufenden und zu verarbeitenden Datenmenge. Zwar gibt es Möglichkeiten den Suchraum einzugrenzen und damit die Arbeit zu minimieren (z.B. [AAGY01]), trotzdem kann die Suche nicht zu weit eingeschränkt werden, da sonst ganze Abschnitte des Webs ausgelassen werden könnten. Dazu kommt, dass ein signifikanter Anteil von Webseiten Duplikate sind, dies aber nur durch schwieriges Clustering zu erkennen ist. Weiter ist Aufgrund des exponentiellen Wachstums und schnellen Änderungen des Webs eine ständige Aktualisierung der durchsuchten Seiten notwendig. Ein nicht minder großes Hindernis stellt die Internationalität der Dokumente dar. So müssten diese, um Zitierungen herauszufinden, zum Beispiel nach Wörtern wie „Literatur“, „Bibliography“ oder „参考文献“ (chinesisch) durchsucht werden. Auch das Format der einzelnen Einträge dieser Listen kann sich unterscheiden. Daher



(a) Crawling (b) Searching

Abbildung 4.1: Schematische Darstellung der Unterschiede zwischen einfachem Crawling und zielgerichteten Suchen beim Abrufen von Daten aus dem Web.

wäre vorab ein hoher Analyseaufwand notwendig, um alle Fälle zu erkennen und korrekt zu interpretieren.

Die zweite Möglichkeit, in Abbildung 4.1b zu sehen, ist es nicht auf die Quellen der Informationen direkt zuzugreifen, sondern auf ein Webverzeichnis, welches diese Daten schon katalogisiert hat. Diese Methode wird Searching genannt. Die Idee hierbei ist, dass das Webverzeichnis wahrscheinlich eine Obermenge der gewünschten Daten enthält und so diese herausgesucht werden müssen. In den meisten Fällen besitzt das Webverzeichnis eine Suchfunktion und ist damit eine Suchmaschine, es ist aber auch möglich, dass bestimmte Inhalte über eine statische Adresse zu erreichen sind, die nur einmal manuell herausgesucht werden muss. Das Webverzeichnis kann sich, wie in Abbildung 4.1b angedeutet, selbst durch Crawling erstellen und aktualisieren oder wird über externe Datenquellen oder manuelle Pflege befüllt. Dies muss einen Nutzer des Verzeichnisses aber nicht interessieren.

Der Vorteil der zweiten Variante ist klar. Die Komplexität des Sammelns und Aussiebens von Daten, wird durch das Verzeichnis übernommen. `Google Scholar` [Goo] ist zum Beispiel ein solches Webverzeichnis für wissenschaftliche Publikationen. Dort ist das Fachwissen über das Abrufen und Organisieren dieser riesigen Datenmengen durch die Suchmaschine `Google` schon vorhanden. Zusätzlich besitzen die großen Verzeichnisse meist große Rechenzentren, die darauf ausgerichtet sind große Datenmengen zu verarbeiten. Auch der Umstand, dass nur kleine Teilmengen der abgerufenen Daten benötigt wird, ist für solche Verzeichnisse kein Nachteil, schließlich wollen sie gerade so vollständig wie möglich sein. Die großen Suchmaschinen haben noch einen weiteren Vorteil, der der Wichtigkeit von guten PageRanks geschuldet ist: sie dürfen teilweise auch geschützte Inhalte indexieren. Zum Beispiel sind bei `Google Scholar` einige Publikationen von `www.springerlink.de` gelistet, die für normale Nutzer nur gegen Bezahlung, und zwar je Artikel, abrufbar sind. Deutlich wird dies, da neben den allgemeinen Daten wie Titel, Autoren oder Jahr der Veröffentlichung, auch Zitierungsinformationen existieren, die nur im `PDF` des Artikels zu finden sind.

Die Nutzung eines Verzeichnisses ist damit wesentlich einfacher und effizienter. Es muss nur noch ein Server kontaktiert und die abgerufenen Daten können so weit wie möglich

eingeschränkt werden. Einziger Nachteil, welcher aber auf keinen Fall missachtet werden darf, ist die fehlende Transparenz. Es kann nicht oder nur schwer nachvollzogen werden woher die katalogisierten Daten stammen. Bei `Google Scholar` werden die Zitierungsinformationen beispielsweise aus dem „Literatur"-Teil des Dokumentes herausgesucht. Wie dabei mit Schreibfehlern, andere Schreibweisen oder ähnlich klingende Titel umgegangen wird, ist nur schwer zu erkennen. Daher muss bevor eine neue Webdatenquelle eingesetzt genau geprüft werden, ob die Qualität den gewünschten Ansprüchen genügt.

Obwohl Searching-Verfahren schon recht gut ist, reicht es in der Praxis nicht aus. Über die Suche wird nur eine Übersichtsseite erreicht, die noch nicht alle gewünschten Informationen beinhaltet. Die relevanten Daten sind noch weiter verteilt. Daher wird nach dem Suchen noch eine Technik mit dem Namen *Browsing* eingesetzt. Diese ist dem Crawling recht ähnlich, mit dem Unterschied, dass nur ein vergleichsweise kleiner Teil der Links auf einer Webseite verfolgt wird und zwar zielgerichtet nach einer bestimmten Strategie. So wie ein Nutzer beim Browsen im Web auch nicht alle auf einer Seite enthaltenen Links verfolgt, sondern nur diejenigen, die ihn interessieren. Wichtiges Merkmal ist zudem, dass keinen externen Verknüpfungen nachgegangen wird und so immer noch nur ein Server abgefragt wird.

Mit dieser Technik: „Searching + Browsing", können so gut wie alle durchsuchbaren Webquellen abgefragt werden. Im nächsten Abschnitt wird dazu ein allgemeines Verfahren vorgestellt, welches in einem Tool verwirklicht wurde.

4.2 Vorgehensweise zur Datenextraktion

In Abbildung 2.1 wurden drei Schritte abgebildet, die nötig sind, um die Webdaten gefiltert zu extrahieren.

1. Generierung von Webqueries
2. Verarbeitung der Webqueries
3. Import der Webergebnisse

Die Generierung der Webqueries ist einfach gehalten. Es wird eine oder mehrere Xmldateien mit passenden Webquery-Knoten geschrieben. Deren Aufbau wird im nächsten Abschnitt näher erläutert. Abhängig von der zu verarbeitenden Webseite erfolgt die Generierung einmalig und zwar wenn statische Einstiegsseiten den Ausgangspunkt bilden. Soll eine Suchfunktion genutzt werden, ist eine dynamische Generierung erforderlich. Hierbei werden abhängig von der Strategie die benötigten Daten aus der Datenbank im Xmlformat abgerufen und durch die anschließende Ausführung einer XSL-Transformation in

```
<webquery url="http://scholar.google.com/scholar?q=$searchstring$&amp;hl=en&amp;lr="
          convert2xhtml="true"
          xslt="google.xsl">
  <param name="pub_id" value="509757"></param>
  <param name="searchstring"
         value="allintitle:&quot;Automated Selection of Materialized Views
                and Indexes in SQL Databases.&quot;"/>
  <param name="mode" value="google_bytitle" />
</webquery>
```

Abbildung 4.2: Beispiel eines Webqueries für eine Publikation mit Suchstrategie „allintitle“.

die korrekte Form gebracht. Die Strategie entscheidet, ob ein Webquery pro Publikation, Autor oder Publikationsort erzeugt wird und welche Suchparameter jeweils übergeben werden. Bei der dynamischen Generierung wird sich nur auf `DBLP` Publikationen in der Datenbank beschränkt, da sie zum einen eine hohe Datenqualität aufweisen, zum anderen fast vollständig bezüglich den relevanten Konferenzen bzw. Journale sind.

Der letzte Punkt ist an dieser Stelle weniger interessant, da es sich hierbei nur um einen einfachen Xmlimport in die Stagingdatenbank handelt. An dieser Stelle wird noch keine Objectmatching oder sonstige Vereinigungen durchgeführt, dies passiert erst in einem späteren Schritt.

4.2.1 Aufbau der Webqueries

Bei Webqueries handelt es sich um die kleinsten Bestandteile im Webextraktionsprozess. Ein Webquery steht für eine Webanfrage und legt die anschließende Verarbeitung des Ergebnisses fest. Die Bedeutung eines Webqueries wird durch die Suchstrategie definiert. Im Allgemeinen können durch die Verarbeitung eines Webqueries beliebig viele Datenknoten entstehen.

Da es sich bei HTML, wie auch XML, um Unterklassen von SGML handelt, lässt sich HTML relativ leicht in XML umformen. Und da zudem für XML zahlreiche Verarbeitungsmöglichkeiten bestehen, bietet sich die Beschränkung des gesamten Extraktionsprozesses auf XML an. Daher werden auch die Webqueries, in XML formuliert. Abbildung 4.2 zeigt eine solche Anfrage für `Google Scholar` mit der „allintitle“-Strategie. Mehr zu Suchstrategien in Abschnitt 4.3.

Ein Knoten besteht aus den folgenden Bestandteilen:

- `url`

  Gibt die URL der aufzurufenden Webseiten an. Der Wert darf Platzhalter enthalten,

welche mit $ umschlossen werden. Auf diese Weise wird der Inhalt automatisch konvertiert, um eine valide URL zu bilden.

- `convert2xhtml`

  Da die meisten Webseiten im älteren HTML-Format geschrieben sind, müssen sie entsprechend in X/HTML umgewandelt werden, um sie ordentlich verarbeiten zu können. In der Implementierung wird dafür die Komponente SgmlReader [Lov06] genutzt.

- `xslt`

  Hier wird die Xml-Transformation angegeben, die die Webseite anschließend verarbeitet. Auf diese Weise werden die wesentlichen Informationen extrahiert. Die Transformation bestimmt zudem die verwendete Strategie.

- `param`

  Es können beliebig viele Parameter angegeben werden. Zum einen werden in der URL enthaltene Platzhalter durch diese ersetzt. Zum anderen werden alle Parameter an die Nachfolgetransformation übergeben, sofern sie dort deklariert wurden.

Im Beispiel zeigt das URL-Feld auf die `Google Scholar` Webseite, welche auch aufgerufen wird, wenn man einen Suchbegriff eingibt und bestätigt. Die Konvertierung zu X/HTML ist eingeschalten und als Nachverarbeitung ist die entsprechende Transformation für `Google Scholar` eingestellt. Als Parameter gibt es den Suchstring, der die gleiche Form hat, als hätte ihn der Nutzer von Hand eingegeben. Daneben, die interne ID der Publikation in der Datenbank. Diese wird ausschließlich für Nachverfolgungszwecke verwendet. Das Matching kann damit nicht durchgeführt werden, da möglicherweise völlig andere Publikationen als Suchergebnis erscheinen. Der letzte Parameter gibt den Modus an, in dem sich der Extraktionprozess gerade befindet. Dieser ist notwendig, da die gleiche Transformation in mehreren Ebenen ausgeführt wird und nur so die richtige Abarbeitungsreihenfolge sichergestellt wird.

4.2.2 Verarbeitung der Webqueries

Aus dem Aufbau des Webqueries wird deutlich, dass die Verarbeitung der Webseiten mit XSL-Transformationen realisiert wird. Daher handelt es sich um einen rein XSL-getriebenen Prozess, denn auch die Filterung der relevanten Daten und die weitere Suchstrategie wird von diesen Transformationen gesteuert.

Im Abschnitt 4.1 wurde verdeutlicht, dass ein einfaches Suchen nicht ausreicht. Das

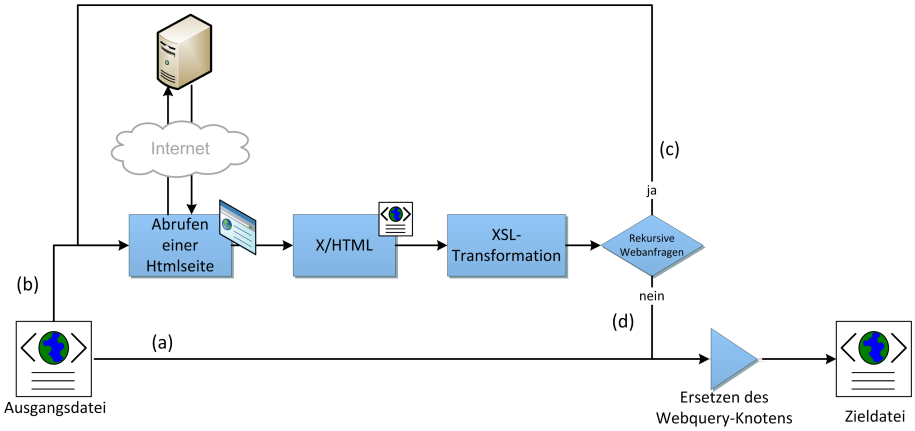


Abbildung 4.3: Schematische Darstellung des allgemeinen Vorgehens zur Webdatenextraktion.

heißt das Abrufen einer Webseite pro initialen Webquery kann noch nicht das gewünschte Ergebnis erzielen. Es sind noch weitere Schritte notwendig. Abbildung 4.3 zeigt das allgemeine Vorgehen schematisch, ausgehend von einer XML-Datei, welche die datenbankgenerierten bzw. statischen Webqueries enthält.

Im einfachsten Fall enthält die Ausgangsdatei überhaupt keine Webquery-Knoten. Dann wird nur der unterste Datenflusspfad verfolgt und damit die Datei eingelesen und sofort wieder ausgegeben werden. Das hieße Ausgangsdatei und Zieldatei wären identisch. Dies ist sehr wichtig, denn es sollen keine schon vorhanden Informationen zerstört werden.

Im zweiten Fall reicht eine einmalige Webanfrage pro Webquery aus. Auch hier wird die Ausgangsdatei inkrementell gelesen. So lange der Parser nicht auf einen Webquery-Knoten trifft, werden die Daten wieder den untersten Datenflusspfad (a) entlang geleitet und sofort ausgegeben. Wird ein Webquery-Knoten gefunden, wird der darüber liegende Pfad (b) eingeschlagen. Als erstes wird nun die URL komplett zusammengebaut und die entsprechende Webseite abgerufen. Optional wird das HTML-Dokument nach X/HTML konvertiert. Anschließend wird die XSL-Transformation ausgeführt. Das entstehende XML-Fragment beinhaltet ausschließlich die Nutzdaten der Webseite in Form von strukturierten Datensätzen. Das heißt, jegliche Style-Informationen wurden entfernt, die Daten aber, ihren semantischen Bedeutungen entsprechend, in eine einheitliche Struktur gefasst. Da eine Webanfrage ausgereicht hat, wird bei „Rekursive Webanfragen“ der „Nein“-Pfad (d) verfolgt. In der Struktur der Ausgangsdatei, welche den untersten Pfad (a) verfolgt hat, wird nun noch das Ergebnis der Transformation an die Stelle des fehlenden Webquery-Knotens übertragen und in die Zieldatei geschrieben. Dadurch wird die Grobstruktur der Ausgangsdatei nie verändert, ausschließlich an den Stellen, an denen sich Webquery-Knoten befinden, können zusätzliche Daten entstehen. Die Webquery-Knoten werden demnach einfach durch die neu gewonnenen Daten ersetzt.

Wenn Browsing notwendig wird, ist der Ablauf noch etwas komplexer. Nun extrahiert die XSL-Transformation neben den Nutzinformationen, zusätzlich noch URLs zu ergänzenden Informationen. Aus diesen URLs werden neue Webquery-Knoten generiert. In diesem Fall, wird bei „Rekursive Webanfragen“ der „Ja“-Pfad (c) verfolgt. Die Ausgabe der Transformation fließt nun nicht sofort in die Zieldatei, sondern wird als Ausgangsdatei für eine neue rekursive Instanz des Prozesses definiert. In diesem neuen Prozess erfolgt die Verarbeitung wieder ganz normal. Die Ausgangsdatei wird nach Webqueryknoten durchsucht und entsprechend die Webseiten abgerufen und verarbeitet. Sonstige Informationen in der Datei werden sofort ins Ziel geschrieben. Das Ergebnis des rekursiven Prozesses ist somit die Vermengung seiner Eingangsinformationen mit den neuen rekursiv abgerufenen Daten. Dieses Ergebnis wird nun wieder ganz normal an die Stelle des initialen Webquery-Knotens geschrieben. Da bei einem Verarbeitungsprozess keine Informationen gelöscht werden, enthält die Zieldatei am Ende Daten aus den Webseiten erster und zweiter Instanz.

Durch die Rekursion ist eine beliebige Abfragetiefe möglich. Das beste Beispiel hierfür ist die Verfolgung von mehreren Ergebnisseiten. Auf Seite eins steht der Link zu Seite zwei, auf Seite zwei der zu Seite drei, usw. Wichtig ist hierbei, dass auch die XSL-Transformationen eine Abbruchbedingung besitzen, so wie es bei rekursiven Algorithmen der Fall ist. Ansonsten würden eventuell, vorausgesetzt man begrenzt sich auf einen Server, alle Seiten des Servers abgerufen werden, sofern sie sich entsprechend verlinken. Im Fall des Verfolgens der nächsten Seite, ist die Abbruchbedingung bei der letzten Seite angekommen zu sein. Verfolgt man aber die Zitierungen, so erhielte man in der zweiten Instanz die zitierenden Papiere. Da diese aber selbst wieder zitiert wurden sein können, dürfen diese Links nicht weiter verfolgt werden.

4.2.3 Verarbeitung von Webqueries am Beispiel Google Scholar

Die im letzten Abschnitt vorgestellte Vorgehensweise ist sehr allgemein gehalten und funktioniert daher mit (fast) allen existierenden Webseiten, so lange sie möglichst einheitlich strukturiert sind. Nun soll davon eine Variante für eine bestimmte Webquelle abgeleitet werden. Dazu muss als erstes das Verhalten des Nutzers analysiert werden, wenn er die gewünschten Informationen abrufen möchte. Auf der Startseite von `Google Scholar` befindet sich nur ein Eingabefeld. Wird dort ein Suchbegriff eingetragen und bestätigt, gelangt der Nutzer zur Ergebnisseite. In Abbildung 4.4 ist dazu das Ergebnis der Suchanfrage aus Abschnitt 4.2.1 dargestellt.

Es erscheinen im Allgemeinen mehrere Ergebnisse, hier nur eins. Im optimalen Fall entsprechen alle Ergebnisse dem Objekt, aus dessen Instanz in der Datenbank die Webquery abgeleitet wurde. Mehrere Ergebnisse können durch Schreibfehler im Titel, fehlende oder unterschiedliche Jahresangaben oder Autoren entstehen. Google versucht diese Duplikate auch selbst schon zu finden und zu gruppieren. Dies wird durch die Angabe *„all 5*

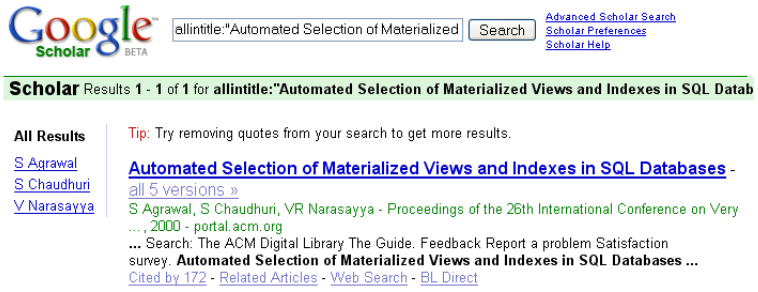


Abbildung 4.4: Ergebnis der Beispielanfrage aus Abbildung 4.2

Abbildung 4.5: Ergebnis der Verfolgung des „Cited by“-Links aus Abbildung 4.4

versions“ angedeutet. Auch wirklich unterschiedliche Papiere mit ähnlichen Titeln führen dazu, dass mehrere Ergebnisse auftreten können. Die durchschnittliche Anzahl der Ergebnisse hängt maßgeblich von der verwendeten Suchstrategie ab.

Jedes Ergebnis besteht aus verschiedenen Informationen. In der ersten Zeile steht immer der Titel des Papiers. In der zweiten stehen Autoren, Konferenz und Erscheinungsjahr, wovon aber keine Angabe obligatorisch ist. Hier zeigt sich schon die erste Schwierigkeit beim Extrahieren der Daten, schließlich müssen die Angaben korrekt zugeordnet werden. Fehlen aber Angaben, könnten theoretisch andere als diese interpretiert werden. Außerdem unterscheiden sich die Anzahl der Autoren stetig, auch hier muss aufgepasst werden, dass sie korrekt aufgeteilt werden. Glücklicherweise wird nach dem letzten Autor immer ein Bindestrich und zwischen Konferenz bzw. Journal und Jahr ein Komma eingefügt, so dass sich diese Zuordnung leicht mithilfe von regulären Ausdrücken lösen lässt.

In der letzten Zeile stehen mehrere Links, an dieser Stelle ist nur „*Cited by 172*“ in-

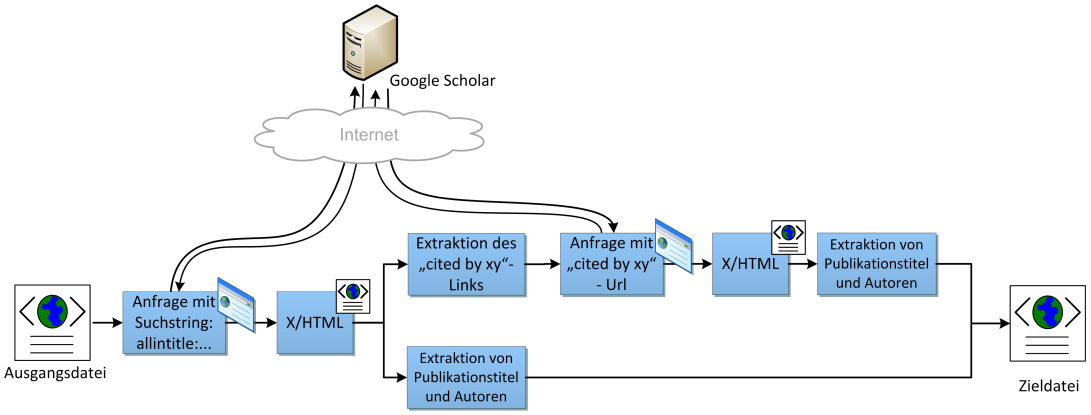


Abbildung 4.6: Schematische Darstellung des regulären Vorgehens zur Webdatenextraktion von Google Scholar.

terressant. Dieser Link führt auf eine weitere Seite, welche die Publikationen enthält, von denen die gesuchte zitiert wurde. Alle Ergebnisse die keinen solchen Link enthalten, werden ignoriert, da sie keinen Mehrwert darstellen.

Verfolgt man diesen Link erhält man eine Seite, die ganz analog zur ersten aufgebaut ist, siehe Abbildung 4.5. Der einzige Unterschied äußert sich darin, dass an der Stelle, an welcher die Anzahl der Ergebnisse steht, statt *„Result x - y of z for <Suchstring>“* nun *„Result x - y of* **about** *z* **citing** *<Titel des Papiers>“* zu lesen ist. Zu bemerken ist hierbei dass Google seinen Ergebniszähler mit dem Wort „about“ schon vorab relativiert. Die folgende Zahl entspricht auch genau der, die im Zitierungslink zu lesen war. Und tatsächlich stellt sich heraus, dass beim Durchblättern der zitierenden Papiere meist weniger auftauchen als angegeben. Dies zeigt besonders deutlich, dass sich bei einer Zitierungsanalyse nicht auf die Zahlen verlassen werde sollte. Sie dienen nur als Richtwert.

Ausgehend von diesen Überlegungen, kann das allgemeine Webdatenextraktionsverfahren an `Google Scholar` angepasst werden. Im regulären Fall, gibt es statt der beliebig tiefen Rekursion nur zwei Ebenen. Die erste Ebene führt den Aufruf mit Suchstring aus und generiert für jeden *„Cited by“* Link eine neue Webquery. Diese werden separat in der zweiten Instanz verarbeitet. Damit liegt für n zu ergänzenden `DBLP`-Publikationen die Anzahl der abzurufenden Webseiten insgesamt bei $O(kn)$, wobei k durch die Suchstrategie bestimmt wird. Mit der hier verwendeten Strategie liegt k in der Realität zwischen 3 und 4. Allerdings kann k auch kleiner als 1 sein, wenn die Strategie mehrere Papiere in einer Abfrage bündeln kann.

Abbildung 4.6 zeigt den Prozess im beschriebenen regulären Fall. Nach dem Webaufruf der ersten Instanz teilt sich der Datenfluss in zwei Teile auf. Im unteren werden die wichtigen Daten der zitierten Papiere extrahiert, im oberen werden die zitierenden Papiere

aufgelöst und deren Daten entsprechend extrahiert. Am Ende werden beide Datenströme so zusammengefasst, dass eine geschachtelte Xml-Struktur entsteht, in der die zitierten Papiere die zitierenden als Unterknoten besitzen. Diese Spaltung und Zusammenfügung entspricht der Redefinition der Erstinstanzausgabe als Eingabe der zweiten Instanz und Ersetzung der Webqueryknoten.

Wie erwähnt erfolgt auch die Datenextraktion innerhalb der XSL-Transformation. Dass hierfür eine genau Analyse des Verhaltens von in diesem Fall `Google Scholar` erforderlich war, um bei verschiedenen Darstellungen stets das gewünschte Ergebnis zu erhalten, soll der folgende Ausschnitt aus der Transformation verdeutlichen.

```
<xsl:attribute name="title">
  <xsl:choose>
    <xsl:when test =./span[@class='w']">
      <xsl:variable name="title" select="normalize-space(script:replaceHtmlEntities(./span[@class='w']))"/>
      <xsl:choose>
        <xsl:when test="starts-with($title, '[') and contains($title, ']')">
          <xsl:value-of disable-output-escaping="yes" select ="normalize-space(substring-after($title, ']'))"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:value-of disable-output-escaping="yes" select ="$title"/>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="write_firstline"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:attribute>
```

In diesem für den Zweck recht langen Ausschnitt wird nur der Titel extrahiert und keine sonstigen Informationen. Diese komplizierte Variante ist allerdings notwendig. `Google Scholar` unterscheidet bei der Darstellung danach, ob es einen Link zu dem Papier gibt oder nicht. Der letzte Fall kann eintreten, wenn die Informationen allein aus den „Literatur“-Abschnitten anderer Publikationen entnommen wurden. Im ersten Fall kann einfach der komplette Inhalt eines *span*-Knotens ausgelesen werden. Im letzten macht sich bemerkbar, dass in HTML schließende Tags optional sind, wovon die Webseite von `Google Scholar` auch exzessiven Gebrauch macht. Dadurch ist der Titel nicht so einfach von den nachfolgenden Textabschnitten unterscheidbar. Hier muss die komplette erste Zeile ausgelesen werden. Dies passiert hier mit dem Aufruf vom Template *write_firstline*, welches hier aus Übersichtsgründen nicht abgedruckt ist. Da der gefundene Suchstring fett gedruckt wird, Stoppwörter aber ausgelassen werden, muss sich das Template durch die verschiedenen Styletags durcharbeiten, bis es entweder auf ein Zeilenumbruch (*br*) oder Absatz (*p*) stößt. Des Weiteren muss der Titel noch von diversen HTML Entitäten befreit werden, beispielsweise Auslassungspunkte, die durch *…* kodiert sind. Außerdem werden dem Titel manchmal Zusatzhinweise über den Datensatz in eckigen Klammern vorangestellt, etwa *[Book]* oder *[Citation]* (siehe Abb. 4.7). Diese müssen ebenfalls entfernt werden.

Abbildung 4.7: Beispiel für Titel mit Zusatzhinweisen, welche entfernt werden müssen.

Die Extraktion der weiteren Informationen gestaltet sich ähnlich. Die Trennung der Autoren, Konferenz und Jahr erfolgt in mehreren `JScript`-Funktionen unter Nutzung von regulären Ausdrücken. Die Trennung der verschiedenen Autoren untereinander ist nur mit einem rekursiven Template zu realisieren, da es in XSL keine Schleifen in der benötigten Form gibt.

Bisher wurde betont vom regulären Vorgehen bei der Webextraktion gesprochen. Wurde ein Papier sehr oft zitiert, müssen mehrere Ergebnisseiten verarbeitet werden, da maximal 100 Ergebnisse pro Seite anzeigbar sind. Da wie schon erwähnt, der Ergebniszähler auch nicht genau die wirkliche Anzahl der Ergebnisse widerspiegelt, gibt es nur die Möglichkeit die *Next*-Links rekursiv zu verfolgen.

Eine weitere Besonderheit von `Google Scholar` ist es, dass die maximale Anzahl der zurückgelieferten Ergebnisse auf 1000 begrenzt ist. Bei 100 Ergebnissen pro Seite gibt es daher nur 10 Seiten. Hier hilft nur ein kleiner Trick weiter. Gibt der Ergebniszähler eine höhere Zahl an, werden die Ergebnisse auf ihr Erscheinungsjahre eingegrenzt. Das heißt beispielsweise wird erst versucht die Datensätze ab 2000 und älter und die ab 2001 abzurufen. Auch dies geschieht wieder rekursiv, befinden sich in dem Zeitraum immer noch zu viele Ergebnisse wird weiter eingegrenzt. Diese Methode hat einen entscheidenden Nachteil. Nicht alle Publikationen besitzen eine Jahresangabe, wodurch diese in keinen Zeitrahmen fallen. Daher bietet nur die Kombination der beiden Varianten eine maximale Ausbeute, obwohl selbst hier noch Publikationen ohne Jahr verloren gehen, nämlich die auf Seite 11 und aufwärts.

Der folgende Xmlcode zeigt einen beispielhaften Ausschnitt aus dem Ergebnis der Webqueryverarbeitung.

```
<citations>
  <publication title="Fast Algorithms for Mining Association Rules in Large Databases"
      url="http://portal.acm.org/citation.cfm?id=672836&amp;amp;dl=GUIDE," year="1994"
      venue="... of the 20th International Conference on Very Large Data ..."
      quality="0.5" pub_id="561794" resultIdentity="12062243690989504063" citationCount=" 1747" >
    <authors quality="0.4">
      <author quality="0.4" firstname="R" lastname="Agrawal" />
      <author quality="0.4" firstname="R" lastname="Srikant" />
    </authors>
    <citation
        title="Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach"
```

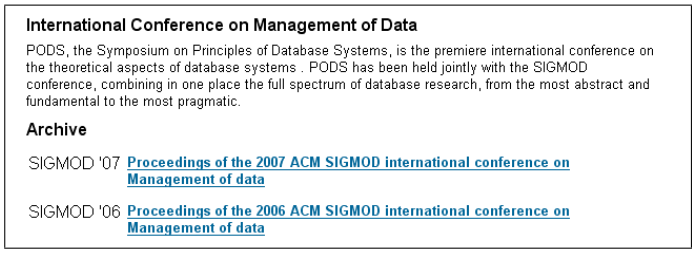
```
        url="http://www.springerlink.com/index/P102L655K52U6327.pdf" year="2004"
        venue="Data Mining and Knowledge Discovery"
        publication_quality="0.5" quality="0.8">
      <authors quality="0.4">
        <author quality="0.4" firstname="J" lastname="Han" />
        <author quality="0.4" firstname="J" lastname="Pei" />
        <author quality="0.4" firstname="Y" lastname="Yin" />
        <author quality="0.4" firstname="R" lastname="Mao" />
      </authors>
    </citation>
    ...
  </publication>
</citations>
```

Auf der obersten Ebene stehen die zitierten Publikationen, welche aus der Webabfrage mit Hilfe der Suchfunktion gewonnen werden. Sie verfügen über die Angabe des Titels, des Publikationsortes, des Erscheinungsjahres und der URL unter der sie `Google Scholar` gefunden hat. Die Attribute `resultIdentity` und `citationCount` werden ausschließlich zum Zweck des Differenzabgleichs benötigt (siehe Abschnitt 4.4). Der Publikation untergeordnet sind zum einen eine Liste ihrer Autoren. Zum anderen eine Reihe von `citation`-Knoten, von denen jeder für eine Zitierung dieser Publikation steht, also aus der Verfolgung des *cited by*-Links hervorgehen. Die `citation`-Knoten sind analog zu den `publication`-Knoten aufgebaut, nur die Hilfsattribute zum Differenzabgleich fehlen. Jedem Objekt wird ein manuell festgelegter Qualitätswert zugeordnet. Dies erleichtert anschließend die Wahl der qualitativ besten Information. Beispielsweise ist der Name der Konferenz der dargestellten Publikation nur unvollständig dargestellt. Die Autorenvornamen sind sämtlich nur mit Initialen angegeben. Auch ist die Liste der Autoren nicht immer vollständig.

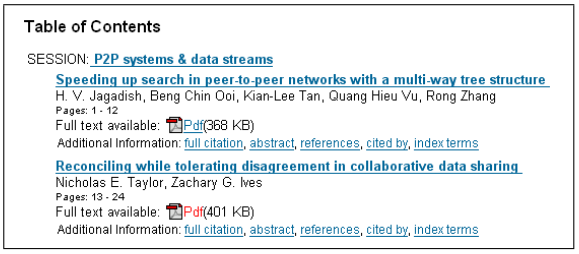
4.2.4 Einbinden weiterer Webdatenquellen

Um eine weitere Webdatenquelle einzubinden, reicht es eine passende XSL-Transformation zu erstellen. Auch hier ist der Webseitenaufbau gründlich zu analysieren, um erst einmal die grobe Vorangehensweise zu ermitteln. Beispielsweise ist es bei `ACM` [fCM] zwar auch möglich nach einzelnen Papieren zu suchen, allerdings werden hier auch ganze Listen von Konferenzen und Journals gelistet. Damit ist es sogar möglich statische Webqueries einmalig zu erstellen, welche die Hauptseite der jeweiligen Konferenz ermitteln und danach ausschließlich mittels Browsing durch die einzelnen Papiere zu navigieren. Unterhalb der Konferenz bzw. Journal, werden die Papiere noch in Jahrgänge unterteilt. Damit sind bei `ACM` im Gegensatz zu den zwei bei `Google Scholar` sogar drei Abfrageebenen notwendig: Konferenz → Jahrgang → zitierende Papiere, wie in Abbildung 4.8 dargestellt.

Weiter ist es notwendig zu wissen, wie die Suchfunktion Webseite unter bestimmten Umständen reagiert. Zum Beispiel reagiert Suchfunktion von `Citeseer` [Cit] auf kleinste Schreibfehler oder zu lange Suchstrings mit der Meldung: *„No documents match Boolean query. Trying non-Boolean relevance query.“*. Die dann erscheinenden Ergebnisse scheinen



(a) Jahrgangsliste für SIGMOD Conference



(b) SIGMOD Conference 2006

(c) zitierende Publikationen

Abbildung 4.8: Drei Schritte sind notwendig, um auf ACM Portal die zitierenden Publikationen zu ermitteln. Da alle Publikationen schon nach Konferenzen und Journalen sortiert sind, ist eine Suche nach einzelnen Publikationen nicht notwendig.

willkürlich zu sein, denn die Titel haben nichts mit dem Suchtext gemeinsam.

Zu Beobachten ist zudem, wie sich die Darstellung verändert, wenn Informationen über eine Publikation nicht vorhanden sind. Bei `ACM` werden bei den zitierenden Publikationen die Angaben wie Autor, Konferenz bzw. Journal, Erscheinungsjahr und weitere kommasepariert angegeben. Fehlt eine Information, rutschen die anderen einfach auf. Jahreszahlen können einzeln oder als komplettes Datum mit Tag und Monat, auch dies in verschiedenen Varianten, und beide Formen gleichzeitig vorkommen. Zudem können natürlich auch Titel selbst Kommas beinhalten. Abbildung 4.9 zeigt ein paar solcher Varianten. Der komplette Text muss daher erst einmal in einzelne Teile zerlegt werden. Hierbei hilft die Beobachtung, dass immer mindestens ein Autor und auch immer der Titel und die Konferenz bzw. Journal angegeben wird. Teilweise werden Seitenzahlen mit angegeben. Diese fallen dadurch auf, dass sie mehr Zahlen als Buchstaben beinhalten. Autorennamen bestehen meist aus zwei oder drei Teilen (erster und zweiter Vorname und Nachname). Auf diese Weise kann ziemlich zuverlässig bestimmt werden, wieviele Autoren es gibt und wo die anderen Attribute beginnen und aufhören. Das Jahr kann dadurch bestimmt werden, dass nach vier ziffrigen Zahlen welche mit 19 oder 20 beginnen gesucht wird.

Auf diese Weise kann so gut wie jede Webseite als Webdatenquelle genutzt werden. Die Informationen müssen nur genügend strukturiert sein, da es sonst unmöglich ist, die relevanten Daten zu extrahieren und auf die korrekten Attribute zu mappen.

Abbildung 4.9: Bei der Darstellung der zitierenden Publikationen, treten bei ACM Unregelmäßigkeiten auf. Alle Attribute werden schlicht kommasepariert aneinander gereiht.

4.3 Suchstrategien

Sofern für das Auslesen der Datenquelle die Suchfunktion der Seite genutzt wird, spielt die Suchstrategie eine maßgebliche Rolle für die erhaltenen Ergebnisse. Verschiedene Strategien unterscheiden sich in Vollständigkeit der Ergebnisse und Performanz. Doch die Wahl der Strategie hängt auch von den Suchoptionen der Seite ab, welche meist unter „Erweiterte Suche“ (bzw. „Advanced Search“) zu finden sind. Bei `Google Scholar` sind neben der normalen Volltextsuche auch Einschränkungen auf Titel, Autoren, Jahr, Publikationsort oder verschiedene Wissengebiete möglich. Die verschiedenen Optionen eignen sich mehr oder weniger gut zur Anwendung in einer Suchstrategie. Während beispielsweise das Jahr eine sehr gut Einschränkung bieten würde, um die meisten ähnlich klingenden Publikationen auszuschließen, würden gleichzeitig auch die korrekten, bei denen lediglich die Jahresangabe in `Google Scholar` fehlt, ausgefiltert werden. Auch eine Einschränkung des Publikationsortes ist kritisch. Einerseits können auch diese Informationen fehlen, andererseits ist unterscheiden sich die Schreibweisen zum Teil signifikant, beispielsweise „VLDB“ und „International Conference of Very large Databases“.

Im Beispiel im Abschnitt 4.2.1 verwendete Strategie erstellt für jede Publikation, für welche Informationen in der Datenbank ergänzt werden sollen, je einen Webquery-Eintrag. Außerdem wird hierbei die *„allintitle“* Funktion verwendet und zusätzlich der DBLP-Titel in Anführungszeichen gesetzt. Der Vorteil dieser Strategie ist, dass in den Ergebnissen fast nur Einträge auftauchen, nach denen auch gesucht wurde. Oftmals wird genau ein Ergebnis gefunden. Auf der anderen Seite werden aber auch relevante Publikationen aufgrund von Schreibfehlern nicht gefunden. Zum Beispiel *„Fast Algorithms for Mining Association Rules in Large Databases“* existiert in dieser Schreibweise sowohl in `DBLP` als auch in `Google Scholar`. Allerdings gibt in letzterem dieselbe Publikation noch einmal ohne den Zusatz *„in large Databases“*, welche durch die strenge Strategie nicht gefunden wird.

Eine wesentlich liberalere Variante ist die Nutzung der *„intitle“* Funktion und das Ersetzen der Anführungszeichen durch runde Klammern. Auf diese Weise wird die Suche wesentlich

unschärfer, es müssen nicht mehr alle gesuchten Wörter im Titel auftauchen und auch die Stellung der Wörter ist egal. Stoppwörter werden komplett ignoriert, wodurch der oft auftretende Fehler des Vertauschens von Präpositionen wie „for“ durch „on“ keine Rolle mehr spielen. Obwohl die Ergebnismenge nun vollständig, ist diese Strategie schon wieder zu liberal. Es werden viel mehr irrelevante Einträge gefunden, wodurch, zusammen mit der Sortierung nach Zitierungen, relevante Einträge zu weit nach hinten geschoben werden. Da unmöglich für alle gefundenen Einträge die zitierenden Papiere aufgelöst werden können, gehen zwangsläufig einige verloren.

Als Mittelweg zwischen der strengen und der liberalen Lösung bietet sich die Hinzunahme eines neuen Filterkriteriums an. Meistens sind die ersten drei Autoren einer Publikation auch bei `Google Scholar` gelistet. Daher kann der erste Autor mit als Suchkriterium verwendet werden. Selbst wenn bei `Google Scholar` die Reihenfolge der Autoren vertauscht ist, wird die Publikation noch gefunden. Dann sollte die Strategie noch berücksichtigen, dass auch in `DBLP` Schreibfehler auftreten können, was sich dadurch äußert dass gar keine Ergebnisse gefunden werden, da die Existenz einer ähnlichen Publikation mit genau dem Schreibfehler im Autor äußerst unwahrscheinlich ist. Auf diese Weise werden (fast) keine relevanten Ergebnisse weggefiltert, andererseits verschwinden die meisten irrelevanten Einträge.

Diese Strategie, welche final die Form *„intitle:(<Titel>) author:<Nachname des ersten Autors>“* hat, wurde auch für das Extrahieren der Daten für die Analyse im Teil II verwendet. Im Gegensatz zur ersten, welche bei der Komplexität $O(kn)$ ein k zwischen 3 und 4 hatte, bewegt sich k hier um die 5. Dies ist ein guter Kompromiss zwischen Performanz und Vollständigkeit.

Weitere Untersuchungen von Strategien sollen an dieser Stelle nicht ausgeführt werden. Die beschriebenen basieren auf den realen Anforderungen und empirischen Untersuchungen. Es sind noch viele weitere Verbesserungen hinsichtlich Performanz, aber sicher auch Vollständigkeit denkbar. Beispielsweise könnten mittels Signifikanzanalyse die signifikanten Wörter eines Titels herausgesucht werden. Dadurch wäre die Ergebnismenge nicht deutlich größer, aber Schreibfehler im Rest des Titels oder Abkürzungen desselben wären irrelevant. Eine andere Möglichkeit ist alle Publikationen mit einem gleichen Autor zu einer Webanfrage zusammenzufassen und damit die Performanz wesentlich zu steigern. Näheres zu diesen Ansätzen ist zum Beispiel in [TAR07] zu finden.

4.4 Differenzabgleich

Webinhalte sind stetigen Änderungen unterworfen, davon sind auch die genutzten Webdatenquellen nicht ausgeschlossen. Werden beispielsweise neue Publikationen in `Google Scholar` importiert, welche einige der abgefragten Publikationen zitieren, ändert sich bei

diesen die Liste aller sie zitierenden Publikationen. Das heißt aber auch, ein einmaliges Abrufen aller Webdaten reicht nicht aus, die Daten müssen in regelmäßigen Abständen aktualisiert werden. Die einfachste Möglichkeit wäre, den gesamten Abfrageprozess erneut komplett durchlaufen zu lassen. Da sich im Zweifelsfall nur wenige Daten geändert haben, ist dieses Vorgehen äußerst unperformant. Der optimalste Weg wäre nur die Differenzen abzugleichen.

Leider verfügt `Google Scholar` über keine Möglichkeit geänderte Daten seit dem letzten Crawling aufzulisten, schließlich ist die Suchmaschine auch eigentlich nicht für eine maschinelle Verarbeitung konzipiert. Da sich neu hinzukommende Publikationen auch nicht auf das aktuelle bzw. das vergangene Jahr beschränken, sondern prinzipiell auch älter sein können oder die Jahreszahl überhaupt unbekannt sein kann, hilft an dieser Stelle auch die Jahresbeschränkung nicht weiter.

Mit einem Trick lässt sich die Arbeit dennoch minimieren. Der „Cited By“ Link enthält eine GS-eigene und GS-weit eindeutige Cluster-ID. Anhand dieser, lässt sich das Cluster später wieder identifizieren. Ein Cluster entspricht hierbei mehreren Publikationen, die GS als die gleiche Publikation erkannt hat. Die Bezeichnung des Links enthält zudem die Anzahl der bekannten zitierenden Publikationen. Mit diesen beiden Informationen, kann recht genau vorhergesagt werden, ob sich etwas an der Liste geändert hat. Nur bei einer Änderung, muss die neu Liste verarbeitet werden. Die minimale Anzahl von notwendigen Webanfragen liegt damit bei gleicher Strategie bei n, also im optimalsten Fall bei einem Fünftel verglichen zum Komplettabruf. Die entstehende Datenmenge liegt dagegen deutlich darunter, da bei keiner Änderung auch keinerlei Daten in die Ergebnisliste übernommen werden.

Dieses Vorgehen hat allerdings zwei Schwachpunkte. Im Worst-Case kommt zur Liste der zitierenden Publikationen gerade einmal eine hinzu. Da diese eine nicht identifiziert werden kann, muss die gesamte Liste neu abgerufen werden. Enthält die Liste mehr als 100 Elemente, sind dann mehr als eine zusätzliche Abfrage notwendig. Bei mehr als 1000 Elementen muss sogar die beschriebene 1000er-Grenze-Umgehungsstrategie zum Einsatz kommen, welche die notwendigen Abfragen noch einmal deutlich erhöht. Der zweite Schwachpunkt liegt in der GS-Cluster-ID. Sporadisch werden von GS alle Cluster neu berechnet, wodurch sich alle Cluster-IDs ändern. Danach sind die schon abgerufenen und ungeänderten Listen von zitierenden Publikationen nicht mehr als ungeändert zu erkennen, wodurch alle Daten komplett neu abgerufen werden müssen.

Prinzipiell wäre das beschriebene Verfahren auch für andere Webdatenquellen anwendbar, so lang eine quellenweit eindeutige Cluster-ID und die Anzahl der zitierenden Publikationen jeweils bekannt sind. Leider ist die letztere Kennzahl bei `ACM` schoneinmal nicht verfügbar. `ACM` muss daher immer komplett abgerufen werden.

4.5 Probleme bei der Webdatenextraktion

- Organisatorische Schwierigkeiten

  Ein organisatorisches Problem ist, dass die Nutzung der Webdatenquellen in der beschriebenen Art und Weise rechtlich zumindest in einer Grauzone liegt. Automatisiertes Abfragen ist zwar in den AGBs nicht direkt ausgeschlossen, aber auch nicht ausdrücklich erlaubt.

  `Google Scholar` versucht auch dagegen vorzugehen indem IP-Adressen bei zu exzessiver Nutzung blockiert werden. Je nach Abfragegeschwindigkeit kann dies nach 400 bis 1000 Anfragen innerhalb eines kürzeren Zeitraums zu einer bis zu 24 stündigen Sperre führen. Die nächste Sperre erfolgt dann umso schneller. Empirische Beobachtungen haben gezeigt, dass maximal drei Anfragen in der Minute zu keiner Sperrung führen. Bei der verwendeten Strategie können so nur knapp über 860 Publikationen am Tag aufgelöst werden. Damit ist die Webextraktion trotz geringer Ressourcennutzung die längste Operation im gesamten ETL-Prozess. Auch `Citeseer` äußert sich bei einer Webnutzung, die aufgrund der verfügbaren Dumps nicht notwendig war, damit, dass bei mehr als 15 Anfragen in der Minute eine *„Server temporär nicht verfügbar“*-Meldung erscheint. Bei einer Verteilung auf verschiedene Rechner tritt dies nicht auf, so dass auch hier eine IP-Sperre vorliegen muss.

- Technische Schwierigkeiten

  - hohe Datenmenge

    Die Betrachtung der Datenmenge hat zwei Betrachtungswinkel, die Größe der Ausgangsdatenbank und die Größe des Ergebnisses der Webdatenextraktion. `DBLP` umfasst circa eine Million Publikationen. Mit der verwendeten Strategie müssten circa. fünf Millionen Webseiten abgerufen werden.

    Im Schnitt wird ein Papier um die 50 mal zitiert. Aufgrund der Vorgehensweise und des fehlenden Object Matchings zwischendurch, wird jedes extrahierte Papier, insbesondere auch die zitierenden als eigenständige Publikation in die Datenbank eingefügt. Dazu kommen irrelevante Publikationen, welche bei der Suche mit gefunden werden. Dadurch liegt die hinzukommende Datenmenge bis zu zwei Größenordnungen über dem Teil, aus welchem die Webqueries anfänglich erstellt wurden.

    Aus diesen beiden Gründen ist es unmöglich alle Publikationen der `DBLP` Datenbank zu berücksichtigen. Eine Beschränkung auf einzelne Konferenzen und Journale ist notwendig.

- Inhaltliche Schwierigkeiten
  - Uneinheitliche Formatierung

    HTML kennt keinerlei semantische Strukturierungsfunktionen, stattdessen werden Inhalt und Layout miteinander vermischt. Einzelne Phrasen werden mal normal, fett oder kursiv gedruckt, Inhalte abhängig von anderen unterschiedlich angeordnet. Zu dem werden oft verschiedene Attribute schwer trennbar zusammen dargestellt. Das Attributmatching gestaltet sich dadurch problematisch und setzt eine eingehende Analyse voraus.

  - Unvollständige Daten

    Die Webverzeichnisse besitzen nicht immer alle Daten. So fehlen teilweise Jahresangaben oder Publikationsort komplett. Autorenvornamen werden meist nur als Initialen angegeben. Zudem werden Titel und Publikationsorte oft abgekürzt und mit Auslassungspunkten gekennzeichnet. Gibt es mehr als drei Autoren, sind davon meist auch nicht alle angegeben.

  - Duplikate

    Besonders `Google Scholar` ist auch von Duplikaten betroffen, da die Informationen sämtlich von gecrawlten Webseiten stammen. Zwar besitzt es, vor allem durch die langjährige Erfahrung von `Google`, einen sehr guten Clusterungsalgorithmus. Allerdings können auch bei Schreibfehlern und unterschiedlichen Angaben nicht immer alle Duplikate erkannt werden.

4.6 Zusammenfassung

Obwohl die inhaltliche Qualität der Webdaten aufgrund einiger Probleme bei der Extraktion nicht sehr gut ist, ist es notwendig diese Quellen zu nutzen, da andere Quellen nicht die benötigten Informationen liefern. Die Qualität der Referenzen zwischen den Datensätzen ist dagegen sehr gut. Dies macht eine abschließende Duplikaterkennung umso wichtiger, damit die fehlerhaften Inhalte durch die qualitativ hochwertigen `DBLP` Inhalte ersetzt werden.

In anschließenden Abfragen an die Datenbank muss außerdem beachtet werden, dass nicht alle Publikationen in `DBLP` für die Webdatenextraktion berücksichtigt werden konnten. Anfragen wie *„Wieviele andere Publikationen werden im Durchnitt zitiert?“* liefern damit unsinnige Ergebnisse, da die Webdaten nur verlässliche Anfragen aus der Sicht der zitierten Papiere ermöglichen, nicht aber umgekehrt.

5 Data Cleaning

Die Datenqualität spielt für spätere Auswertungen eine entscheidende Rolle. Hier gilt der Leitsatz *„Garbage in, garbage out“*, welcher besagt, dass Fehler in den Ausgangsdaten auch Fehler in den Ergebnissen verursachen. Die Ergebnisse können durch fehlende Datenqualität vollkommen ihre Gültigkeit verlieren und daraus gezogene Rückschlüsse einfach falsch sein. Vor allem im Umgang mit externen Datenquellen ist darauf zu achten, dass Fehler in einzelnen Quellen nicht die Qualität des gesamten Systems kompromittieren.

Auf qualitätsmindernde Eigenschaften der Webdatenquellen wurde in Kapitel 4 schon eingegangen. Möglichkeiten zur nachträglichen Qualitätssteigerungen sind Verfahren der Kategorien *Data Scrubbing* und *Data Auditing* [BG04]. Auf diese soll an dieser Stelle nicht genauer eingegangen werden.

Im nächsten Abschnitt wird auf `DBLP` eingegangen, da Fehler dort vor allem einen negativen Einfluss auf die Webdatenextraktion besitzen können. Besonders wichtig wird durch die Verwendung der verschiedenen nicht disjunkten Datenquellen eine Beseitigung der Duplikate, siehe Abschnitt 5.2.

5.1 Titelbereinigung

5.1.1 Motivation

Obwohl `DBLP` handgepflegt und damit ziemlich sauber ist, enthält es auch kleinere Fehler. Um über Online-Suchmaschinen Informationen über bestimmte Publikationen zu ermitteln, spielt, je nach verwendeter Suchstrategie (Abschnitt 4.3), die Qualität der Titelangaben eine entscheidende Rolle. Daher lohnt es sich, diese einmal genauer zu betrachten. Dabei fällt es auf, dass bei einigen Titel Metainformationen in Klammern enthalten sind.

In Abbildung 5.1 sind einige solcher geklammerter Ausdrücke aufgelistet. Während „`abstract`“ bzw. „`extended abstract`“ Aufschluss über das Paper selbst gibt, zeigen die anderen Varianten den Typ an. So kann es sich zum Beispiel um ein Demonstrationspapier oder um ein Tutorial handeln. Obwohl diese Ausdrücke sicherlich informativ sind, gehören sie

1. A Territorial Database Management System (Abstract).
2. Data Functions, Datalog and Negation (Extended Abstract).
3. High-Dimensional Index Structures, Database Support for Next Decade's Applications (Tutorial).
4. InfoSleuth: Semantic Integration of Information in Open and Dynamic Environments (Experience Paper).
5. Toto, We're Not in Kansas Anymore: On Transitioning from Research to the Real (Invited Industrial Talk).
6. Controlled natural language interfaces (extended abstract): the best of three worlds.

Abbildung 5.1: geklammerte Metainformationen

1. Architectures and Algorithms for Internet-Scale (P2P) Data Management.
2. The Ins and Outs (and Everthing in Between) of Data Warehousing.
3. 2d-Bubblesorting in Average time O(N lg N).
4. An Online Video Placement Policy based on Bandwith to Space Ratio (BSR).
5. What is the Data Warehousing Problem? (Are Materialized Views the Answer?)

Abbildung 5.2: erwünschte geklammerte Ausdrücke

eigentlich nicht zum Titel.

Werden diese geklammerten Ausdrücke nicht entfernt, so kann dies später zu zwei Problemen führen. Wird beim Benutzen einer Suchmaschine zum Auflösen der Zitierungsinformationen eine zu strenge Strategie verwendet, ist es unmöglich Ergebnisse zu erzielen. Auch beim Matchen verschiedener Datenquellen, kann die Qualität beeinflusst werden. Schließlich werden die anderen Quellen, wie `Google Scholar` oder `ACM`, nicht diese geklammerten Metainformationen beinhalten. Damit wird die Ähnlichkeit abgestuft und ein Match möglicherweise verhindert.

Also stellt sich die Frage, wie diese Ausdrücke automatisch erkannt werden können, um sie anschließend zu entfernen. Dabei sollten auch neue Varianten und Mischformen erkannt werden. So findet sich zum Beispiel auch „`extended abstract - invited talk`“ und ähnliche Variationen bzw. Kombinationen in der `DBLP`-Datenbank. Die triviale Lösung wäre, alle geklammerten Ausdrücke zu entfernen. Dagegen sprechen allerdings die Beispiele in Abbildung 5.2. Diese gehören alle samt eindeutig zum Titel des jeweiligen Papers und die Entfernung hätte genau den gegenteiligen Effekt. Auch an der Position des geklammerten Ausdrucks ist nicht absehbar, ob er zum Titel gehört oder nicht. Obwohl Metainformationen meist am Ende auftreten, zeigt der sechste Titel in Abbildung 5.1, dass sie auch in der Mitte stehen können. Andererseits können auch korrekte Titel Klammern am Ende beinhalten. Siehe dafür die letzten beiden in Abbildung 5.2.

5.1.2 Bewertung

Um ein wirksames Verfahren zu entwickeln, muss dessen Qualität bestimmt werden können. Dadurch können verschiedene Varianten miteinander verglichen werden. Hierzu wird die Precision P und Recall R bestimmt [BYRN99]. Sei dazu *gelöscht* die Menge der vom Algorithmus gelöschten Ausdrücke und *meta* die Menge der Ausdrücke die Metainformationen darstellen und daher zu löschen wären.

$$P = \frac{\text{gelöscht} \cap \text{meta}}{\text{gelöscht}}$$

und

$$R = \frac{\text{gelöscht} \cap \text{meta}}{\text{meta}}$$

, wobei P in diesem Fall angibt, wieviele gelöschte Klammerausdrücke auch tatsächlich gelöscht werden sollten. R dagegen gibt den Anteil der tatsächlich gefundenen zu löschenden Ausdrücke an. Da sich die beiden Werte invers zu einander verhalten, bietet sich das F-Measure als Kombination der beiden Werte an:

$$F = 2 * \frac{P * R}{P + R}$$

. Zur Auswertung der Algorithmen wurden 1000 zufällig ausgewählte Klammerausdrücke manuell bewertet, so dass die Ergebnisse dazu verglichen werden können.

5.1.3 Triviale Vorgehensweise

Wie schon erwähnt, wäre die einfachste Lösung einfach alle Klammern in Publikationstiteln zu entfernen. Bei genauer Betrachtung lässt sich aber feststellen, dass es sich bei nicht einmal die Hälfte der Klammern, um überflüssige Ausdrücke handelt. Mit diesem Verfahren würde man zwar für Recall 1 erhalten, schließlich wurden alle zu löschenden Ausdrücke wirklich gelöscht, allerdings wäre $P = 0,43$ und damit $F = 0,60$ für Testmenge. Ließe man dagegen alle Titel unberührt, erhielte man immerhin $P = 1$, da allerdings kein einziger überflüssiger Audruck entdeckt wurde gilt ebenfalls $R = F = 0$. Dies zeigt deutlich, dass das F-Measure weniger ein Indikator dafür ist, wie viel falsch oder richtig gelöscht wurde, als für die Effektivität des Algorithmus.

Es ist also auf triviale Weise möglich ein F-Measure von 0.6 zu erreichen. Diesen Wert gilt es daher zu schlagen.

5.1.4 Stoppwortanalyse

Einen besseren Ansatz erhält man, wenn man sich die Ausdrücke in Abbildung 5.1 einmal genauer betrachtet. Dies sind zwar manuell ausgewählte Beispiele, aber trotzdem reprä-

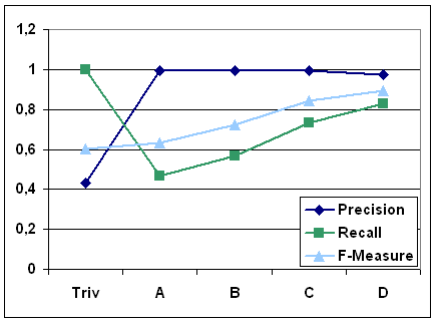


Abbildung 5.3: Bewertung des Stoppwortalgorithmus

sentativ. Gleich in drei Ausdrücken kommt das Wort „abstract“ vor, zwei davon mit dem Zusatz „extended“. Diese Beobachtung legt die Idee nahe, zu erst einmal alle Klammern zu löschen, in denen das Wort „abstract“ vorkommt. Abbildung 5.3 zeigt die Ergebnisse unter „A“, im Vergleich zum trivialen Verfahren und den noch kommenden Varianten. Mit $F = 0,63$ liegt diese Variante nur leicht über dem trivialen Wert. Aber in Hinsicht auf Precision und Recall unterscheidet sich Variante A deutlich von Triv. Während der Recall natürlich nicht sonderlich hoch sein kann, wenn man sich nur auf ein einziges Wort konzentiert, so ist die Präzision bei annähernd 100%. Einziger falscher Fund in der Testmenge ist die Publikation

- HYLAS: program for generating H curves (abstract three-dimensional representations of long DNA sequences)

. Dies zeigt einerseits, dass das Wort „abstract“ in Klammern deutlich auf eine Metainformation hindeutet. Andererseits zeigt das Beispiel auch die Gefährlichkeit, wichtige Informationen zu entfernen, die einen gänzlich anderen Sinn haben. Also nicht einmal als grenzwertig zu betrachten sind.

Um den Recall-Wert noch zu erhöhen müssen weitere Terme ausgefiltert werden. Dafür kann man sich die Liste von vorkommenden Termen zusammen mit ihren Häufigkeit anschauen. Dieses Verfahren wird für allgemeine Dokumente als Stoppwortanalyse bezeichnet [HQW06]. Wobei in diesem Fall, aufgrund der speziellen Auswahl von in Klammern auftretenden Termen, nicht von Stoppwörtern gesprochen werden kann. Das Prinzip ist aber das gleiche, denn genau wie Stoppwörter haben auch Terme wie „abstract“, welche in sehr vielen Klammern auftreten, einen niedrigen Informationsgehalt für den Titel und sind damit überflüssig. Abbildung 5.4 zeigt die zehn am meist auftretenden Terme.

Wie erwartet nehmen „abstract“ und „extended“ die ersten beiden Positionen ein, wobei letzteres fast immer gemeinsam mit „abstract“ auftritt. Auffällig in dieser Liste sind die Einträge vier bis sieben. Hierbei stehen „n“, „k“ und „M“ meist für ganzzahlige Variablen

Anzahl	Term
3869	abstract
2098	Extended
1169	n
919	Panel
514	k
498	and
469	of
442	a
432	M
421	Preliminary

Abbildung 5.4: Top 10 der in Klammern auftretenden Terme

wie in „n über k“ oder „k Merge“. Bei „and“, „of“ und „a“ handelt es sich dagegen um Stoppwörter im klassischen Sinn, daher an dieser Stelle irrelevant.

Mit diesem Wissen kann Variante A verbessert werden. In Variante B wurden zusätzlich zu Klammern mit „abstract“, auch Klammern mit „extended“ oder „Panel“ entfernt. In Variant C kommen zusätzlich noch „Preliminary“, „session“, „Paper“, „Invited“, „Report“, „Tutorial“ und „Note“ dazu. Mit Variante B steigt der F-Measure-Wert immerhin schon auf $0,72$ an, mit Variante C sogar $0,84$. Dabei sinkt die Präzision nicht nennenswert ab, allerdings liegt der Recall bei Variante C immer noch nur bei $0,73$, d.h. bei vier zu entfernenden Ausdrücken, wird einer nicht gefunden. Betrachtet man die nicht gefundenen Ausdrücke, findet man einige der Art:

- A Distributed Computational Science Simulation Environment (May 2003).
- Book review: Mathematical methods for neural network analysis and design (The MIT Press, 1996).

Einige Ausdrücke enthalten Jahreszahlen. Diese tauchen aber nicht in der Topliste der Terme auf, da sie sich auf verschiedene Jahre verteilen. Ersetzt man die Jahreszahlen durch den generischen Term „[year]“, ordnet dieser sich mit 3546 Vorkommnissen sogar direkt hinter „abstract“ ein. Variante D berücksichtigt zusätzlich zu C auch noch den Term „[year]“. Damit wird ein F-Measure von $0,89$ erreicht.

5.1.5 Automatisierte Stoppwortanalyse und Fachwortanalyse

Es sei vorweggenommen, dass sich das F-Measure von Variante D nicht mehr deutlich verbessern lässt. Trotzdem sei auf zwei entscheidende Nachteile dieses Verfahrens hingewiesen. Zum einen wird der hohe Wert vor allem durch die hohe Präzision erreicht. Es

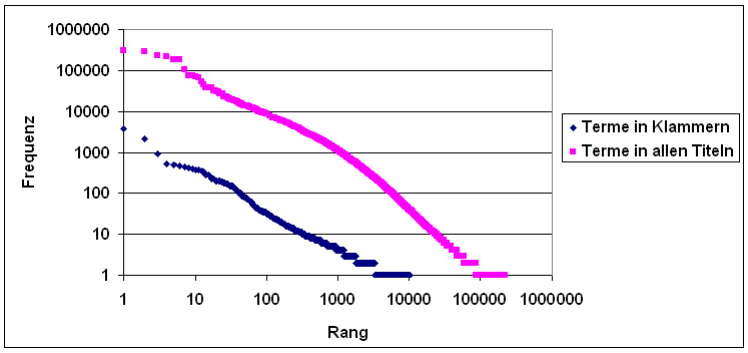


Abbildung 5.5: Korrelation von Rang und Frequenz der Terme. Die untere Kurve stellt alle Terme die in Klammern vorkommen dar, die obere alle Terme in allen Titeln ohne Einschränkung.

werden immer noch nur vier von fünf zu löschenden Ausdrücken erkannt.

Aber ein größerer Nachteil ist, dass die rauszufilternden Terme manuell rausgesucht werden müssen. So steht der Term „Note“ beispielsweise an 32. Position. Für jeden in den Filter aufgenommenen Term muss sichergestellt werden, dass er auch wirklich einen postiven Effekt hat, also er keine oder nur wenige falsche Funde erzeugt, wie das Wort „abstract“ im Beispiel „HYLAS: ...“. Außerdem macht es das Verfahren unflexibel. Zum Beispiel gibt es in `DBLP` auch ein paar wenige deutsche Publikationen mit dem Zusatz „(Zusammenfassung)“. Würden nun vermehrt deutsche Publikationen in `DBLP` eingelesen werden, müsste das Verfahren manuell angepasst werden.

Zum besseren Verständnis, wie ein automatisches Verfahren die Relevanz eines Termes ermitteln kann, soll Abbildung 5.5 dienen. Hier werden die Häufigkeiten der Terme dargestellt, in absteigender Reihenfolge. Beide Achsen sind logarithmiert. Offensichtlich sind die entstehenden Kurven, in dieser Achsendarstellung, annähernd linear. Wie in [BYRN99] dargestellt, gilt für Terme in normalen Texten das Zipfsche Gesetz. Es besagt, dass i.häufigste Term $\frac{1}{i^\theta}$ so oft vorkommt, wie der häufigste Term, wobei θ den die Verteilung beschreibenden Exponent darstellt. Die Häufigkeit eines Terms mit Rang i kann damit geschätzt werden: $f(n, V, i) = \frac{n}{i^\theta H_V(\theta)}$. Hierbei steht n für die Anzahl der Terme insgesamt und V für die Größe des Vokabulars, also die Menge verschiedener Terme. $H_V(\theta)$ ist die harmonische Zahl von V mit der Ordnung θ, definiert als $H_V(\theta) = \sum_{j=1}^{V} \frac{1}{j^\theta}$.

Die beiden in Abbildung 5.5 dargestellten Kurven stehen in dem Sinne für verschiedene Textmengen. Die untere berücksichtigt nur Terme in Klammern, die obere alle Terme in allen Titeln. Auch die Häufigkeiten sind entsprechend berechnet. Nicht überraschend ist, dass sowohl die maximale Häufigkeit, als auch das Vokabular bei der zweiten Kurve um zwei Größenordnungen höher liegt. Schließlich beinhalten auch nur ein Bruchteil der Titel Klammern und auch der Inhalt der Klammern ist meist kurz gehalten. Inter-

essant ist dagegen, dass beide Kurven annähernd parallel verlaufen. Dies deutet auf die Gleichartigkeit des Auftretens von Termen in Titeln und Klammern hin. Das wird auch bestätigt, betrachtet man die Werte für θ. In [BYRN99] führen die Autoren an, dass θ für reale Texte Werte zwischen $1,5$ und $2,0$ annimmt. Für die geklammerten Terme gilt dagegen $\theta = 0,95$ und für die Terme aller Titel $\theta = 1,05$. Obwohl sich die beiden Werte deutlich von einander unterscheiden, liegen sie im Vergleich zu realen Texten viel enger zusammen. Das bedeutet, dass in Titeln eine höhere Gleichverteilung herrscht. Das kann man damit erklären, dass Titel wesentlich kürzer sind als Texte und daher nur wenige Füllworte enthalten können, um den Kern der Publikation auszudrücken. Da jedoch eine riesige Themenvielfalt herrscht, müssen sich zwangsläufig auch die Titel stärker von einander unterscheiden.

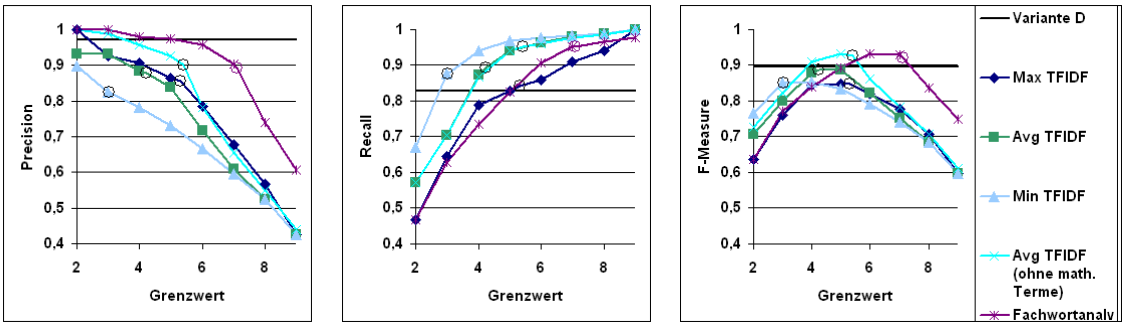
In Abbildung 5.4 wurde die zehn häufigsten Terme in Klammern aufgeführt. Für alle Titel insgesamt sind dies die Terme: `of, for, a, and, the, in, on, with, to, an`. Aufgrund der höheren Länge von Titel gegenüber Klammerausdrücken kommen nun also auch viel mehr klassische Stoppwörter zum Einsatz.

Die Idee zum Entfernen von irrelevanten Klammern ist nun wieder die gleiche, wie im letzten Abschnitt. Nur das nun die irrelevanten Terme erst noch identifiziert werden müssen. Intuitiv sinkt die Relevanz eines Terms, mit der Menge der Ausdrücke, in denen er vorkommt. Aufgrund der Kürze der Ausdrücke, entspricht dies annähernd dem absoluten Vorkommen des Terms. Aufgrund dieser Überlegungen kann die Relevanz eines Terms mit Hilfe des TF-IDF Maßes berechnet werden. V stehe wieder für das vorkommende Vokabular.

$$
\begin{array}{rcl}
\text{Sei} & tf_{t,a} & = \text{Anzahl der Vorkommen von Term t in Ausdruck a} \\
 & freq_t & = |\{a : tf_{t,a} > 0\}| \\
 & max\_freq & = max\{freq_t : t \in V\} \\
 & idf_t & = log\left(\frac{max\_freq}{freq_t}\right) \\
\text{dann ist} & tfidf_t & = tf_t * idf_t
\end{array}
$$

Daraus ergibt sich, dass der Wert von $tfidf$ von einem Term t im gleichen Maße ansteigt, wie seine Relevanz. Mit Hilfe eines Grenzwertes können damit alle irrelevanten Terme identifiziert und ausgefiltert werden. Da ein Ausdruck aus mehreren Termen besteht, gibt es die Möglichkeiten der Minimum-, Maximum- oder Durchschnittsbildung. Abbildung 5.6 zeigt die Ergebnisse dieses Verfahrens in Abhängigkeit vom Grenzwert. Wie im letzten Abschnitt, wurden Jahreszahlen unter einem generischen Term zusammengefasst. Zudem wurden alle anderen Zahlen ignoriert. Zum Vergleich ist ebenfalls die beste Variante mit statischem Wortfilter, Variante D, abgebildet.

Wie man sieht unterscheiden sich die Maximas des F-Measures stark hinsichtlich des genutzten Grenzwerts an dieser Stelle. Die Kreise auf den Kurven zeigen das Ergebnis an der Stelle, an der der Grenzwert den Median aller $min-, max-$, bzw. $avg - tfidf$



(a) Precision (b) Recall (c) F-Measure

Abbildung 5.6: Auswertung der automatischen Titelsäuberungsverfahren. Die schwarzen Kreise markieren den Schnittpunkt von Median TFIDF und Grenzwert. Die schwarze Gerade stellt das Ergebnis von Variante D aus dem letzten Abschnitt.

Werte annimmt. Dieser Wert liegt immer recht nah am Maximum, da auch unter den Klammerausdrücken ungefähr die Hälfte als irrelevant gilt.

Von den drei neuen Varianten, schneidet $avg - tfidf$ offenbar am besten ab. Obwohl $min - tfidf$ einen hohen Recallwert hat, ist die Precision recht schlecht. Dies liegt daran, dass die klassischen Stoppwörter, wie „`and`“ einen niedrigen Wert für $tfidf$ erhalten und damit den Ausdruck dominieren. Bei $max - tfidf$ verhält es sich gerade anders herum. Die Precision ist zwar hoch, aber der Recall ist schlecht. Hier dominieren seltene Terme (z.B. auch Schreibfehler) den ganzen Ausdruck. Mit $avg - tfidf$ werden die Vorteile beider Varianten kombiniert und somit das beste Ergebnis erzielt.

Allerdings liegen die F-Measures aller drei Varianten unterhalb dem von Variante D, wobei $avg - tfidf$ dem schon sehr nah kommt. Vor allem die Precision ist viel niedriger, da nicht mehr nur ausgewählte Terme gefiltert werden. Dafür ist der Recall bei allen neuen Varianten höher, da nun auch Ausdrücke gefunden werden, deren Terme nicht manuell ermittelt wurden.

Betrachtet man die noch falsch erkannten Ausdrücke, fällt auf, dass einige mathematische Terme immer in Klammern auftreten. Bestes Beispiel ist „O(log n)“ als Komplexitätsangabe. Diese Ausdrücke werden von diesem Algorithmus natürlich fälschlicherweise als irrelevant erkannt. Sie müssen daher für die Ermittlung der Termhäufigkeiten ignoriert werden. Die angepasste Variante von $avg - tfidf$ stellt die fünfte Kurve in Abbildung 5.6 dar. Es ist nochmal eine signifikante Steigerung des F-Measures auf 0.93 zu verzeichnen.

Eine weitere Variante entsteht, wenn man die in [HQW06] beschriebene Fachwortanalyse

etwas abwandelt. In einer Fachwortanalyse wird das Auftreten von Termen eines Textes oder eine Textmenge gegen den Gesamtkorpus verglichen. Dabei werden die Terme ermittelt, die relativ signifikat öfter vorkommen als im Korpus. Diese gelten dann als Fachwörter des Textes bzw. der Textmenge. Zum Beispiel werden in Publikationen aus dem Bereich der Informatik Wörter wie „Algorithmus“ oder „Datenbank“ relativ wesentlich öfter vorkommen, als in allen Publikationen zusammen.

Hier in diesem Kontext könnte man die Menge aller Klammerausdrücke als die zu analysierende Textmenge betrachten. Die Menge aller Titel ist der Gesamtkorpus. Die Überlegung ist nun, dass Wörter wie „`Panel`“ oder „`Paper`“ wesentlich öfter in Klammern vorkommen und damit als „Fachwörter“ in diesem Bereich gelten. Nur kann man diese Worte nun nicht einfach filtern, da es auch Gegenbeispiele gibt. So kommt das Wort „`abstract`“ zwar relativ gesehen öfter in Klammern vor, allerdings nicht signifikat öfter. Es würde also nicht erkannt werden. Dagegen werden oftmals Abkürzungen in Klammern geschrieben. So kommen diese insgesamt nur selten vor und werden daher bei $avg-tfidf$ daher gar nicht in Betracht gezogen. Aber sie kommen relativ öfter vor als im Korpus. Also muss der ursprüngliche Wert von $tfidf$ immer noch Einfluss haben.

$$
\begin{array}{lll}
\text{Sei} & freq_t^{(glob)} & = \text{Vorkommen von } t \text{ im Gesamtkorpus} \\
 & idf_t^{(glob)}, max\_freq^{(glob)}, V^{(glob)} & \text{analog definiert} \\
 & {idf'}_t^{(glob)} & = idf_t^{(glob)} * \dfrac{log \sum_{t \in V} freq_t}{log \sum_{t \in V^{(glob)}} freq_t^{(glob)}} \\
\text{dann sei} & idf_t^{FA} & = idf_t^{\min\left\{1, \frac{idf_t}{{idf'}_t^{(glob)}}\right\}}
\end{array}
$$

Der Faktor bei ${idf'}_t^{(glob)}$ ist notwendig, um die unterschiedliche Textmenge auszugleichen. Die Variante $avg - tfidf$ kann nun so abgewandelt werden, dass sie idf_t^{FA} benutzt. Das Ergebnis ist in der letzten Kurve in Abbildung 5.6 zu sehen. Es hat sich nur auf der Grenzwertachse verschoben, es sind aber keine Verbesserungen zu erkennen. Dies deutet schon darauf hin, dass das Verfahren allgemein gesättigt ist und die verbleibenden Fehler mit dieser Methode nicht beseitigt werden können.

Der Recall ist nun mit Werten um 0.95 recht gut, die Precision mit 0.90 immer noch akzeptabel. Betrachtet man die falsch eingeschätzten Ausdrücke, so ist auch ersichtlich, warum das automatisierte Auffinden so schwierig ist. Die Klammern im folgenden Beispiel werden beispielsweise nicht gefunden. Im ersten Fall aufgrund eines Schreibfehlers, im zweiten aufgrund der sehr seltenen Abkürzung von „`abstract`“.

- Towards Synthesis of Nearly Pure Prolog Programs (Extende Abstract).

- Causal encoding of Markov sources (Ph.D. Abstr.).

Die Klammern in den nächsten beiden Titel werden dagegen fälschlicherweise entfernt. Im ersten Fall werden `m` und `n` ignoriert, dadurch bestimmt das Stoppwort `and` den Wert. Im zweiten Fall handelt es sich um das Trademarkzeichen, welches immer in Klammern geschrieben wird. Auch dieses kommt relativ häufig vor und wird damit entfernt.

- Partitions and (m and n) Sums of Products.

- Series Approximation Methods for Divide and Square Root in the Power3(TM) Processor.

Ein besseres Ergebnis wäre wohl nur mit einem intelligenteren Verfahren zu erreichen. Etwa indem getaggte Wörterbücher verwendet würden oder der Algorithmus mit einer Trainingsmenge selbst erlernen könnte, welche Klammern relevant sind und bei welchen es sich wirklich nur um Metainformationen handelt.

5.2 Duplikaterkennung

Die Vereinigung von verschiedenen Datenquellen führt in allen Einsatzbereichen oft zu Duplikaten. So sind zum Beispiel zwei Kundendatenbanken nicht notwendigerweise disjunkt, besonders wenn sich beide in einem ähnlichen geographischen oder wirtschaftlichen Umfeld bewegen. Bei einer Zusammenführung müssten diese Überschneidungen erkannt werden.

Beim Aufbau dieses Data Warehouse wurden Duplikate aber gerade vorsätzlich erzeugt. Dies ergibt sich aus der Arbeitsweise des ETL-Prozesses. Die qualitativ hochwertige Datenbank `DBLP` soll schon möglichst vollständig sein. Die aus den weiteren abgefragten Datenquellen gewonnenen zitierten Papiere sind damit zu großen Teilen Duplikate der `DBLP`-Publikationen. Dies gilt auch für die zitierenden Papiere, denn auch wenn nach ihnen nicht direkt aus den `DBLP` Daten heraus gesucht wurde, so bewegen sie sich in der gleichen Domäne. Aufgrund der Arbeitsweise bei der Webdatenextraktion (Abschnitt 4.2), entstehen auch schon in den Daten der Webdatenquellen per Definition Duplikate. So kann ein zitiertes Papier auch als zitierendes auftreten oder ein Papier mehrere andere Papiere im Suchbereich zitieren.

Umso wichtiger ist es nun eine Duplikaterkennung durchzuführen, um bei den Auswertungen verschiedene Objekte nicht mehrmals zu zählen. Daher muss als erstes ein Object Matching durchgeführt werden, um Paare des gleichen Papiers herauszusuchen. Dieses Object Matching kann allerdings nicht nur von den unsauberen Datenquellen auf `DBLP` erfolgen, obwohl `DBLP` dublettenfrei ist. Mit dieser Methode würden aber keine Dubletten

zwischen den Datenquellen erkannt, welche kein korrespondierenden Datensatz in DBLP besitzen.

Das Problem wird noch dadurch verschärft, dass auch die Datenquellen selbst, aufgrund von Fehler in der dortigen Datenextraktion (zum Beispiel aus den PDF Dateien oder OCR aus eingescannten Publikationen), zu unterschiedlichen Schreibweisen in Publikationstiteln oder Autoren und damit zu zusätzlichen Duplikaten kommt. Außerdem sind die Publikationen nicht durch global eindeutige IDs, wie *ISBN* oder *DOI*[1], gekennzeichnet, wodurch eine einfaches ID-Matching unmöglich wird. Es sind unscharfe Verfahren nötig, die auch über Schreibfehler hinwegsehen und die Duplikate trotzdem aufdecken.

Da mit dem Matching nur Paare entstehen, ist anschließend eine Clusterung notwendig. Damit werden alle Dubletten einer realen Publikation genau einem Cluster zugeordnet. Nur so sind später korrekte Auswertungen möglich.

5.2.1 Object Matching

Object Matching bedeutet für jede Objektreferenz in Menge A eine ähnliche Objektreferenz in Menge B zu finden. Die Zuordnung muss nicht notwendigerweise eindeutig sein. Zudem müssen A und B nicht disjunkt in Bezeug auf die Objektreferenzen sein. Zum Beispiel macht auch $A \subset B$ einen Sinn, wenn Duplikate aufgelöst werden sollen, für die keine Objektreferenz in $B \backslash A$ existiert, aber $B \backslash A$ duplikatfrei ist. Dieser Fall tritt hier ein und $B \backslash A$ entspricht den importierten DBLP-Datensätzen.

Im Bereich der Zitierungsanalyse wird dieser Vorgang auch oft als *Citation Matching* bezeichnet. Da entsprechende Verfahren in der Literatur schon oft und ausführlich untersucht wurden (bspw. in [EIV07]), sollen an dieser Stelle nur grundlegende Betrachtungen durchgeführt werden, die für die weitere Arbeit notwendig sind.

Die einfachsten Objekt Matcher sind attributbasierte Verfahren. Hier werden die Ähnlichkeiten der Attributwerte zwischen den Referenzen, zum Beispiel mit einer Editierdistanz, gemessen und anschließend gemittelt. Mittels eines Grenzwertes wird bestimmt ob zwei Referenzen möglicherweise auf das gleiche Objekt verweisen oder nicht.

Im Falle der Zitierungsdatenbank, können beispielsweise der Publikationstitel, die Autoren und das Erscheinungsjahr als Attribute genutzt werden. Allerdings gibt es im Allgemeinen mehrere Autoren zu einer Publikation, die aber meist in der gleichen Reihenfolge angegeben werden. Außerdem werden in Google Scholar und ACM oft maximal drei Autoren angegeben und deren Vornamen abgekürzt. Mit diesem Wissen, können die Nachnamen der ersten drei Autoren in der korrekten Reihenfolge kommasepariert als

[1] Digital Object Identifier

Titel	Autoren	Jahr	Quelle
Automated Selection of Materialized Views and Indexes in SQL Databases	Sanjay Agrawal, Surajit Chaudhuri, Vivek R. Narasayya	2000	DBLP
Automated Selection of Materialized Views and Indexes in SQL Databases	S Agrawal, S Chaudhuri, VR Narasayya	2000	Google Scholar
Automated selection of materialized views and indexes in microsoft sql server	A Sanjay, C Surajit, VR Narasayya	2000	
Automated Selection of Materialized Views and Indexes in SQL	S Agrawal, A El Abbadi	2000	

Abbildung 5.7: Beispiele für Duplikate in der Stagingdatenbank

Autorattribut genutzt werden. Dazu kommt, dass das Jahr nicht für alle Datensätze vorhanden ist und damit keine einfache Mittelung der Ähnlichkeiten möglich wird. Hier ist es am besten die Ähnlichkeiten erst nur für Titel und Autoren zu ermitteln und später für Paare mit bekannten Jahr auf beiden Seiten die Ähnlichkeiten entsprechend anzupassen.

In Abbildung 5.7 sind für ein Beispiel die verschiedenen Duplikate, begrenzt auf `DBLP` und die ersten drei Treffer bei `Google Scholar`, dargestellt. Allein bei `Google Scholar` gibt es aber noch sechs weitere verschiedene Schreibweisen und damit Duplikate. Der `DBLP` Eintrag beinhaltet den korrekten Titel, die Autorennamen und Publikationsjahr. Mit dem beschrieben Verfahren würde der erste `Google Scholar`-Treffer mit einer Ähnlichkeit von $1,0$ dem `DBLP`-Eintrag zugeordnet werden. Im zweiten GS Beispiel ist der Titel leicht abgewandelt, was auf die verwendete normalisierte Editierdistanz aufgrund des langen Titels nur einen sehr kleinen Einfluss hat. Dagegen sind bei den ersten beiden Autoren Vor- und Nachname vertauscht und die Nachnamen dadurch noch abgekürzt. Auch im dritten Beispiel sind vor allem die Autoren vom Fehler betroffen. Der zweite Autor hat nichts mit dem Papier zu tun, sondern gilt als Editor der 2000er VLDB Konferenz.

Dieses Beispiel demonstriert das Problem beim Finden der Duplikate. Besonders rein attributbasierte Verfahren reagieren auf Schreibfehler äußerst anfällig. Allerdings stellt das Weglassen der Autoren bei der Ähnlichkeitsanalyse keine Alternative dar, da sich viele Papiere in der gleichen Domäne bewegen und sich damit Titel oft ähneln. Hier im Beispiel werden die Duplikate letztlich doch gefunden, da die Ähnlichkeit der Titel bei nahezu 1 liegt, das Jahr gleich ist und sich die Autoren nicht völlig unterscheiden.

Alternative Ansätze zur Duplikatbestimmung sind möglich, wobei die Attributähnlichkeiten immer die Grundlage bilden. Zum Beispiel ist es möglich Kontextinformationen wie Koautorschaften zu nutzen, um auch falsche Schreibweisen zu finden (zum Beispiel [OEL+06]). Mittels eines datenflussorientierten, probabilistischen Ansatzes [DHM05] wären auch die Vor- und Nachnamendreher eine Informationsquelle, da sie immerhin eine kleine Ähnlichkeiten der Autoren beisteuern und damit die Wahrscheinlichkeit, dass sich die beiden Referenzen auf das gleiche Objekt beziehen steigt. Zugleich werden hier durch den Datenfluss auch Informationen schon gefundener Duplikate genutzt, um die Ähn-

lichkeit zu anderen Referenzen zu erhöhen. Auch ist es möglich Kontextinformationen in weiter entfernten Ebenen zu nutzen [DHM05], [KM05]. Der Spezialfall der Zitierungsanalyse [BACÖAH05] nutzt dabei das Wissen über Zitierungen. Zwei Publikationen sind damit ähnlich, wenn sie die gleichen Publikationen zitieren. In einer Erweiterung könnten statt der Vorraussetzung, dass es sich bei der zitierten um die gleiche Referenz handeln muss, auch die Ähnlichkeiten setzen. Das würde heißen, zwei Publikationen wären ähnlich, wenn die von beiden zitierten Publikationen ähnlich sind. Hier wird auch deutlich, dass die attributbasierten Verfahren weiterhin eine wesentliche Rolle spielen, schließlich können auch komplett verschiedene Papiere zufällig die gleichen oder ein paar gleiche Papiere zitieren. Aus diesem Grund werden in [KM05] auch Verbindungen zu sogenannten Hubs, also oft referenzierte Knoten, untergewichtet, da diese Verbindungen eine kleinere Aussagekraft besitzen.

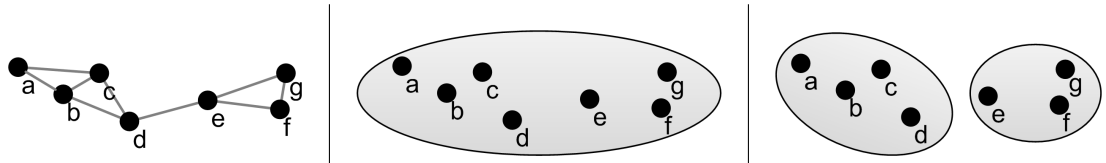
Um zu verhindern, dass falsche Duplikate ermittelt werden, kann Domänenwissen in Form von Bedingungen eingesetzt werden ([DLLH03] oder [SLD05]). Beispielsweise wäre die Gleichheit der Jahresangabe eine gute Bedingung, besser als die Nutzung der Attributähnlichkeit, denn was heißt 2007 und 2008 liegen genau ein Jahr auseinander? Eine weitere mögliche Bedingung wäre, dass eine nicht-`DBLP`-Publikation auf maximal eine `DBLP`-Publiktion gemappt werden darf, da `DBLP` dublettenfrei ist.

Eine weitere wichtige, aber problematische Aufgabe ist es, dass das Einfügen von neuen Datensätzen möglichst effizient erfolgt [LKM+07], schließlich sollen vor allem die Webdatenquellen regelmäßig aktualisiert werden. In dem, im ETL-Prozess relisierten, Verfahren ist dies nur teilweise gelöst. Wie schon erwähnt bildet bei den genannten Mengen A und B die DBLP-Datenbank bei der ersten Ausführung die Menge $B \backslash A$. Dies ist auch bei späteren Ausführungen der Fall, obwohl hier ein `DBLP`-Datensatz durch die Dedublizierung der früheren Iterationen, schon um neue Informationen angereichert worden sein kann. Alle Datensätze die nicht auf eine `DBLP`-Publikation gemappt werden konnten, gehen in der neuen Ausführung im vollen Umfang in die Menge A ein. Der Grund ist der, dass bei der Clusterung (siehe Abschnitt 5.2.2), Cluster in einzelne Cluster zerfallen können. Kommen neue Datensätze hinzu, könnten diese als Bindeelement dienen und so die Cluster verschmelzen.

5.2.2 Clusterung

Nach dem beschriebenen Object Matching liegen nur die Ähnlichkeiten zwischen den Objektreferenzen vor. Die Liste dieser Ähnlichkeiten bildet einen ungerichteten[2] gewichteten Graphen. Um die Daten für Auswertungen verwenden zu können, muss jede Referenz eindeutig einem Objekt zuordenbar sein. Dies ist im Allgemeinen aber nicht der Fall, da jede Objektreferenz im Graph mehrere Verbindungen zu anderen Knoten besitzen kann. Die

[2] an dieser Stelle wird von symmetrischen Ähnlichkeitsmaßen ausgegangen



(a) Ähnlichkeitsgraph (b) Transitive Hülle (c) HAC

Abbildung 5.8: Zweidimensionale Darstellung eines Ähnlichkeitsgraphes und dessen Clusterung. Die Kanten in (a) stehen für Ähnlichkeiten zwischen den Knoten, die Länge der Kanten für deren Gewichtung.

in Abbildung 5.7 vorkommenden Elemente sind im besten Fall alle untereinander verbunden, in diesem Fall wäre die Zuordnung eindeutig. Allerdings gibt es auch Fälle in denen die Teilgraphen nicht vollständig verbunden sind. Zwei `DBLP`-Einträge sind per Definition nicht verbunden. Sind sie dennoch ähnlich, gibt es vermutlich Knoten aus anderen Datenquellen, die mit beiden `DBLP`-Knoten verbunden sind. Es ist also eine Clusterung notwendig, um die Zuordnungen eindeutig zu bestimmen. Die Clusterung sollte streng sein, da sonst in Auswertungen nicht klar wäre, wann eine Information zu einem Objekt gezählt werden darf.

In Abbildung 5.8a ist ein solcher Ähnlichkeitsgraph dargestellt. Die grafische Anordnung entspricht auch den Ähnlichkeiten der Knoten untereinander. Im Allgemeinen würden die Knoten aber einen m-dimensionalen Raum einnehmen, wobei m kleiner als die Anzahl der Knoten ist. Wie man leicht sieht, sind jeweils die Knoten a,b,c und b,c,d, aber auch f und g sehr ähnlich zueinander. Der Knoten e liegt in der Mitte zwischen diesen Knotenmengen, aber mehr zu f und g hingeneigt.

Nun gibt es verschiedene Clusteringverfahren. Da der Gesamtgraph aus vielen unabhängigen Teilgraphen besteht, ist die transitive Hülle die einfachste Version. Dabei werden alle Knoten, die untereinander mit mindestens einem Pfad verbunden sind, dem gleichen Cluster zugeordnet. Das hat den Nachteil, dass schon weiter entfernte Brückenelemente, also Elemente zwischen zwei Teilgraphen, die zu beiden Graphen relativ aber nicht vollständig unähnlich sind, dazu führen, dass beide Teilgraphen in ein Cluster gelangen. In Abbildung 5.8b ist dies der Fall. Obwohl a und g sehr weit auseinander liegen, befinden sie sich im gleichen Cluster, da sie durch e verbunden sind.

Andere sehr effiziente Algorithmen wie der `k-Means`-Algorithmus [Mac66] und seine Derivate, können hier nicht angewendet werden, da dafür die Anzahl der Cluster a priori feststehen muss. Es ist nur bekannt, dass die Clusteranzahl größer oder gleich der Datensätze in `DBLP` sein muss, da aber `DBLP` nicht vollständig ist, kann die Gleichheit ausgeschlossen werden.

Eine bessere Alternative bieten *Hierarchical Agglomerative Clustering*-Verfahren (HAC),

z.B. [DGC06]. Hierbei werden die Knoten anfänglich selbst als Cluster betrachtet. Anschließend werden jeweils die beiden ähnlichsten Cluster miteinander verschmolzen. Von da an werden die beiden einzelnen Cluster nicht mehr einzeln betrachtet, sondern nur noch das zusammengeschmolzene Cluster. Dieses Verfahren verfährt iterativ, bis es keine Cluster mehr gibt, die verschmolzen werden können. Da das Endergebnis wieder der transitiven Hülle entsprechen würde, muss die Verschmelzung an einem festgelegten Grenzwert abgebrochen werden.

Als Vorraussetzung zur Ausführung des HAC-Verfahrens, ist ein Ähnlichkeitsmaß zwischen den Knoten, hier schon vorhanden, und eine Funktion, die aus diesen die Ähnlichkeiten zwischen zwei Clustern berechnet. Dafür gibt es drei wesentliche Funktionen, die in diesem Fall anwendbar sind. `Single-Link` bestimmt den Wert der beiden ähnlichsten Knoten der beiden Cluster. Das Ergebnis würde der transitiven Hülle entsprechen, wobei alle Kanten mit einem Gewicht unterhalb des HAC-Grenzwertes entfernt wurden. `Complete-Link` bestimmt den Wert der beiden unähnlichsten Knoten. Dies würde dafür sorgen, dass die Cluster nur einen bestimmten Durchmesser besitzen dürfen. Dadurch wird das Clusteringverfahren aber relativ unstabil gegenüber dem Granzwert. Kleine Änderungen könnten dafür sorgen, dass viele Cluster entstehen bzw. zusammenfallen. Im Beispiel wären a und d in getrennten Clustern und nur einer von beiden wäre mit b und c zusammen, obwohl auch der andere eine relativ hohe Ähnlichkeit zu diesen besitzt. Einen Mittelweg bietet `Average-Link`. Hier wird der Durchschnitt der Ähnlichkeiten zwischen allen Knoten der beiden Cluster ermittelt. Das Ergebnis für das Beispiel ist in 5.8c zu sehen.

Die hierarchische Natur des Verfahrens ist an dieser Stelle irrelevant. Nur das Endergebnis beim Erreichen des Grenzwertes ist interessant. Die Hierarchien werden sozusagen flachgedrückt. Da die Teilgraphen relativ unabhängig voneinander sind, kann der Algorithmus relativ performant angewandt werden, indem er gleichzeitig auf allen Teilgraphen läuft. Auf diese Weise sind maximal so viele Schritte nötig, wie die maximale Anzahl an Knoten in einem Teilgraph.

5.3 Relationales Merging

Im letzten Abschnitt wurde dargestellt, wie Duplikate innerhalb der Daten gefunden und gruppiert werden können. Trotzdem beinhalten die verschiedenen Elemente der Cluster teilweise ergänzende, teilweise aber auch widersprüchliche Informationen. In Abbildung 5.9 sind zwei Cluster mit jeweils zwei Beispielelementen abgebildet. Zum einen unterscheiden sich jeweils die Titel der Elemente, zum anderen fehlen Angaben wie Jahr oder Venue in einigen Einträgen. Es ist also eine Vereinigung der Clusterinhalte auf einen Datensatz der relationalen Datenbank notwendig, um bei Auswertungen zum Beispiel über das Jahr, eindeutige Aussagen über das Objekt treffen zu können.

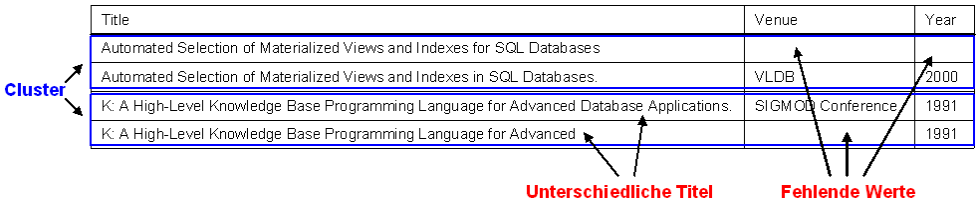


Abbildung 5.9: Beispiel zweier Cluster und die Unterschiede in den Clusterelementen.

1. Stufe: eine Tabelle

Im ersten Schritt kann eine einzelne geclusterte Tabelle betrachtet werden. Der *naive* Ansatz ist, dass einfach alle Duplikate bis auf einen Datensatz gelöscht werden. Der verbleibende Datensatz könnte zum Beispiel der `DBLP`-Datensatz sein, falls einer im Cluster existiert. Dieser Ansatz hat allerdings den Nachteil, dass alle zusätzlichen Informationen aus den anderen Quellen verloren gehen würden. Ist beispielsweise kein `DBLP`-Datensatz im Cluster, da diese meist vollständig und korrekt befüllt sind, könnte ein Datensatz das Jahr beinhalten, ein anderer den Publikationsort. Eine Information geht verloren. Dieser Ansatz ist also nur sinnvoll, wenn sich immer mindestens ein qualitativ hochwertiger Datensatz im Cluster befindet. Dann werden auch keine Daten unterschiedlicher Datensätze vermischt. Diese Vorraussetzung ist aber im Allgemeinen nicht gegeben.

Aus diesem Grund ist ein *erweiterter* Ansatz notwendig, der auch die Informationen der anderen Datensätze berücksichtigt. Dafür wird erst ein Clusterzentrum gewählt und dann werden für alle Attribute die jeweils besten Werte im Cluster zusammengesucht. Die Frage ist nun, welcher Attributwert jeweils gewählt wird. Eine Möglichkeit ist ein Majority Vote durchzuführen, d.h. es wird der Wert übernommen, der in den meisten Datensätzen des Clusters vorkommt. Das geht aber schief, wenn man die Arbeitsweise bei der Integration der Webdaten (Abschnitt 4.2) berücksichtigt. Ein Webdatensatz kann zum Teil mehr als einmal in identischer Form in der Datenbank vorkommen, nämlich dann, wenn die entsprechende Publikation mehrere andere zitiert, nach denen gesucht wurde. Dann wäre der eigentlich korrekte `DBLP` Wert stets in der Unterzahl und es würde das Attribut des Webdatensatzes übernommen werden.

Eine Alternative stellt das Vergeben von Qualitätswerten dar. Es wird dann der jeweils qualitativ höchstwertigste Attributwert im Cluster ausgewählt und in den verbleibenden Datensatz übernommen. Im in Abbildung 2.2 dargestellten Schema sind schon entsprechende Attribute, zusätzlich zu den Nutzdaten, zu sehen. Beim Import der Daten müssen entsprechend die Qualitätsattribute befüllt werden. `DBLP` erhält dabei den höchsten Wert. Damit werden, falls verfügbar, immer die `DBLP` Daten bevorzugt. Nichtausgefüllte Attribute in `DBLP` können aber dennoch ergänzt werden.

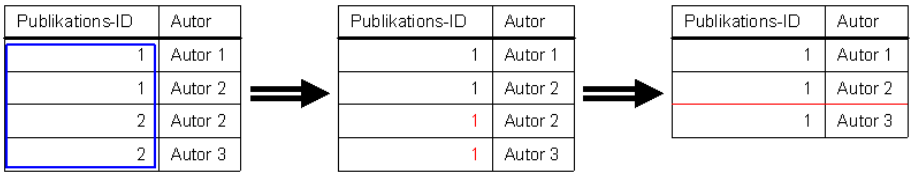


Abbildung 5.10: Beispiel zum Merging von Fremdschlüsseln

2. Stufe: Beachtung von Fremdschlüsselbeziehungen

Die Datenbank besteht nicht nur aus einzelnen Tabellen, sondern aus mehreren, die über Fremdschlüssel miteinander verbunden sind. Die abhängigen Tabellen sind beim Mischen der Clusterdatensätze mit zu beachten. Zum einen ist es notwendig die Fremdschlüssel auf die jeweiligen Clusterzentren umzulenken, da alle anderen Datensätze der Cluster am Ende gelöscht werden. Dabei können allerdings neue Duplikate in der abhängigen Tabelle entstehen, da es sich bei diesen Tabellen meist um M:N-Tabellen handelt. Besitzen beispielsweise zwei Publikationen eines Clusters eine Zuordnung zu dem gleichen Autor, ist dieser nach der Verschmelzung doppelt. Siehe dazu Abbildung 5.10. Um zu Erkennen, wann ein Datensatz der abhängigen Tabelle ein Duplikat ist, muss vorher festgelegt werden, welche Attribute sich unterscheiden müssen. Das ist meist für die Fremdschlüssel auf die andere Tabelle der M:N-Relation der Fall. Weitere Attribute, wie zum Beispiel die Autorposition, werden auf die gleiche Weise wie in Stufe 1 neu bestimmt.

Natürlich ist es aufgrund der unterschiedlichen Schreibweisen von Autorennamen in den Webdatenquellen möglich, dass unter den Autoren nicht alle Duplikate gefunden werden. Da in `DBLP` die Autoren immer vollständig ausgefüllt sind, ist es aber unnötig die falsch geschriebenen Autoren der Webdaten beizubehalten. Daher werden auch hier die Qualitätsattribute mit einbezogen und nur die M:N-Zuordnung der höchsten verfügbaren Qualität übernommen.

3. Stufe: Kompletter Algorithmus

Zusammengenommen ergibt sich folgender Algorithmus:

1. Gegeben: geclusterte Tabelle, Clustertabelle für diese Tabelle
   Bsp: `[dbo].[publications]` als geclusterte Tabelle. Clustertabelle ist eine temporäre Tabelle, welche jeder Publikation genau eine Cluster-ID zuordnet.

2. Bestimmung der jeweiligen Clusterzentren

3. Für jede Fremdschlüsselbeziehung auf diese Tabelle:
   a) Bestimmung einer abgeleiteten Clustertabelle für die abhängige Tabelle
   b) Vereinigung der neuen Cluster (rekursiver Aufruf)
   c) Umleiten der Fremdschlüssel auf die Clusterzentren
   d) Optional: Löschen von M:N-Zuordnungen geringerer Qualität
4. Für jede Spalte der geclusterten Tabelle:
   a) Suchen des nichtleeren Wertes mit der höchsten Qualität für jedes Cluster
   b) Übernahme dieses Wertes ins Clusterzentrum

   Löschen aller Zeilen, die nicht Clusterzentrum sind

Mit dieser Methode lassen sich beliebige relationale Cluster so in ein Element vereinigen, dass der Informationsverlust minimal wird und damit die Gesamtqualität des System sogar ansteigt, da qualitativ schlechtere Attributwerte oder gar Zuordnungen von besseren überschrieben werden.

In Punkt 3b wird eine Rekursion eingeleitet. Damit wird die abhängige Tabelle selbst wieder als geclusterte Tabelle betrachtet und der Algorithmus auf dieser ausgeführt. Damit sind auch höhere Rekursionsstufen möglich. Im Schema in Abbildung 2.2 betrifft das beispielsweise die Tabellen `[dbo].[citations]` und `[dbo].[citation_Sources]`. Die erstere ist eine M:N-Tabelle zwischen Publikationen, die letztere zwischen Zitierungen und Datenquellen. Natürlich müssen auch die letzteren Zuordnungen aktualisiert werden. Sind beispielsweise zwei Publikationen aus `Google Scholar` und aus `ACM` gekommen, werden deren Zitierungen gemischt. Haben sie Zitierungen auf die gleichen Publikationen, werden auch die Quellen vereinigt. So sind am Ende Auswertungen über die Gesamtzitierungszahlen oder der Vergleich von `GS`- und `ACM`-Zitierungen leicht möglich.

Am Beispiel der `[dbo].[citations]`-Tabelle wird auch deutlich, dass die Iteration über alle Fremdschlüsselbeziehungen und nicht nur über die abhängigen Tabellen notwendig ist. So werden zum Beispiel erst die Zitierungen aus der Sicht der zitierenden Publikation, später aus der Sicht der zitierten entsprechend geclustert und vereinigt.

5.4 Zusammenfassung

Die Säuberung der Daten hat eine essentielle Bedeutung, um eine interpretierbare Auswertung, aus der keine falschen Schlüsse gezogen werden, zu gewährleisten. Die hier besprochenen Methoden wurden genutzt, um das in der Analyse verwendete Data Warehouse aufzubereiten. Andere teilweise effizientere, teilweise bessere Qualität versprechende Verfahren sind möglich und in einigen wissenschaftlichen Arbeiten untersucht worden. Aufgrund des modularen Aufbaus des beschriebenen ETL-Prozesses zur Erzeugung und Wartung des Data Warehouses, ist es aber jederzeit möglich neue Verfahren auszuprobieren und die Ergebnisse zu vergleichen.

Teil II

Zitierungsanalysen

6 Verwendete Daten

Aufgrund der in Teil I ausgeführten Umstände, ist es unmöglich eine vollständige Datenmenge zu schaffen. Im Gegensatz zu Data Warehouses im geschäftlichen Umfeld, bei denen meist eine begrenzte Anzahl von Datenmengen existiert und jede einzelne nahezu vollständig für einen gewissen Teilbereich ist, gibt es für Zitierungsanalysen eine Unmenge an Bibliographiedatenbanken, welche allesamt unvollständig sind. Im geschäftlichen Umfeld ist es daher „nur“ schwierig, das heißt mit finanziellen und zeitlichen Kosten verbunden, ein vollständiges Data Warehouse zu erzeugen. Für das Zitierungs-Warehouse ist dies sogar theoretisch unmöglich.

Da man auch im geschäftlichen Umfeld nicht immer bereit ist die vollen Kosten zu tragen, wählt man dort oft den *Think big, Start small*-Ansatz (z.B. [BG04]). Dabei werden einige Datenquellen ganz und Teile von anderen Datenquellen erst einmal außen vor gelassen, wodurch die Komplexität der Aufgabe wesentlich verringert wird. Der Nachteil dieser Methode ist allerdings, dass die für die Analyse benötigten Daten vorher identifiziert werden müssen. Wurden vertikal getrennte Teilmengen, wie zum Beispiel Bestellinformationen, vergessen beziehungsweise übersehen, äußert sich das darin, dass die gewünschten Abfragen über diese nicht mehr durchführbar sind, da die Objekte oder Objektattribute fehlen. Wurden dagegen horizontal getrennte Teile ausgelassen, beispielsweise eine ganze Niederlassungen, so sind die Abfragen trotzdem möglich, führen aber vielleicht zu falschen Ergebnissen, wenn zum Beispiel die Gesamtumsatzzahlen der Firma abgefragt werden sollen. Statistische Verzerrungen und Fehler sind die Folge.

Diese Gegebenheiten müssen für die Gestaltung der Analyse immer berücksichtigt werden. Die zu integrierten Datenquellen müssen so beschnitten werden, dass eine Analyse der Zitierungen immer noch möglich ist. Für den Einfluss von Publikationen spielt im wesentlichen nur eine Rolle, wie oft sie selbst zitiert wurden, nicht aber wieviele andere Papiere sie selbst zitieren. Daher bietet sich die Begrenzung der zitierten Papiere auf gewisse gewünschte Konferenzen und Journale an, in denen sie erschienen sind. In diesem Fall sind dies speziell die Konferenzen *VLDB*[1], *SIGMOD*[2] und die Journale *VLDB Journal*, *SIGMOD Record* und *ACM TODS*[3]. Natürlich werden die zitierenden Papiere nicht auf diese Publikationsorte eingeschränkt, da nur ein Teil der Zitierungen aus ihnen stammt, die absoluten Zitierungszahlen für die erfassten Papiere aber möglichst vollstän-

[1] `http://www.vldb.org`
[2] `http://www.sigmod.org`
[3] `http://tods.acm.org`

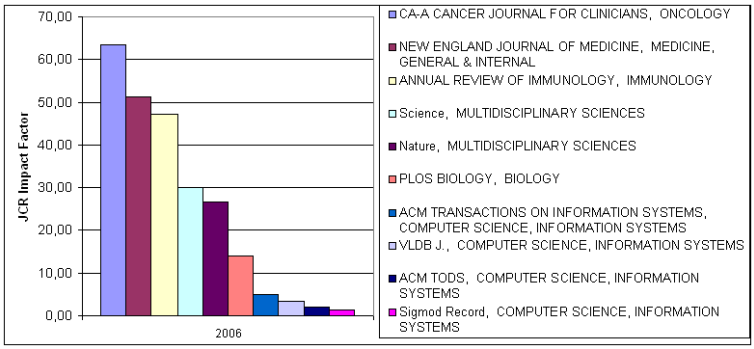


Abbildung 6.1: Vergleich der JCR Impaktfaktoren diverser Journale für 2006. Quelle: [oK06]

dig sein sollen. Die verwendeten Datenquellen wurden in Abschnitt 2.1 aufgeführt und erläutert.

Die Beschränkung der zitierten Papiere auf einzelne Themengebiete hat aber noch einen anderen wichtigen Hintergrund. In verschiedenen wissenschaftlichen Gebieten, herrschen andere Zitierungspraktiken. Während in einer Disziplin Themen nach wenigen Jahren schon veraltet sind, und damit entsprechende Publikationen nicht mehr zitiert werden, sind in anderen Disziplinen auch ältere Papiere noch aktuell. Die Disziplinen unterscheiden sich auch hinsichtlich der Dauer, bis die Ergebnisse einer Arbeit von anderen aufgegriffen werden. Sind in einer Disziplin die Zitierungshalbwertszeiten sehr lang, ist ein geringerer Impaktfaktor zu erwarten, da hierzu nur eine begrenzte Anzahl an Jahren berücksichtigt wird.

Abbildung 6.1 zeigt die JCR Impaktfaktoren für 2006. Ausgewählt wurden für mehrere Bereiche die Journale mit dem höchsten Einfluss, wobei das *CA-A CANCER JOURNAL FOR CLINICIANS* das in den JCR am höchsten bewertete Journal repräsentiert. *Science* und *Nature* sind mit Abstand die beiden besten im Bereich allgemeinen naturwissenschaftlichen Journale.

Es fällt auf, dass vor allem medizinische und biologische Journale sehr hohe Impaktfaktoren erhalten. Dies ist auch mit der höheren Zielgruppe der Journale zu erklären. Mit der Anzahl der Leser steigt steigt auch die Wahrscheinlichkeit der Zitierung. Um aussagefähige Vergleiche zwischen den Ziterungszahlen verschiedener Bereiche durchzuführen, muss die Wahrscheinlichkeit zitiert zu werden aber für alle Papiere der Journale etwa gleich sein ([MM96]). Am besten lässt sich dies sicherstellen, indem nur Journale der gleichen Domäne betrachtet werden. So liegen die hier zu betrachtenden Journale *VLDB Journal*, *SIGMOD Record* und *ACM TODS* vergleichsweise nah beieinander. Auch das Topjournal der gleichen Kategorie ist im Vergleich zu den anderen genannten, in der gleichen Größenordnung.

Mehr zu Impaktfaktoren, ihrer Berechnung und Interpretation wird im folgenden Kapitel behandelt.

Zusätzlich zu den beschriebenen Datenquellen, wurden noch Daten über Publikationstypen und Länderzuordnungen, die auch dem Papier [RT05] zu Grunde lagen, integriert. Diese Daten wurden manuell gesammelt und beziehen sich auf zwischen 1994 und 2003 erschienene Publikationen in den berücksichtigten Publikationsorten. Daher können Analysen über diese Attribute, nur über die angesprochenen Jahre gehen. Weiter werden im Folgenden auch Vergleiche zu den alten Daten angestellt. Diese stammen dann nicht aus dem Data Warehouse, sondern direkt aus dem genannten Papier und werden im Folgenden als *[2005]* und die neuen Daten als *[2008]* bezeichnet werden.

7 Zitierungsanalyse

7.1 Vergleich der Quellen

Nach den gemachten Vorüberlegungen, kann das Data Warehouse nun für seine eigentliche Aufgabe genutzt werden, der Zitierungsanalyse. In Abschnitt 2.1 wurden die genutzten Datenquellen vorgestellt. Sie unterscheiden sich alle sehr stark hinsichtlich Abdeckung. In Abbildung 7.1 sind die Zitierungszahlen für die betrachteten Publikationsorte für die jeweiligen Datenquellen dargestellt. *All* repräsentiert die Vereinigung der Zitierungsinformationen aller Quellen des Data Warehouses. Überschneidungen werden dabei nur einmal gezählt, das bedeutet der jeweilige Wert entspricht nicht der Summe der Werte der einzelnen Quellen, sondern vielmehr der Aggregation über die einzelnen Zitierungen. Die Zahlen für *JCR* stammen nicht aus dem Data Warehouse, sondern wurden direkt aus [oK06] extrahiert und nur zum Vergleich aufgeführt. In *JCR* fehlen per Definition die Konferenzen *VLDB* und *SIGMOD*, außerdem sind hier Zahlen für *VLDB Journal* erst ab 1998 und *SIGMOD Record* ab 2000 aufgeführt. Trotzdem wird deutlich, dass in JCR nur ein Bruchteil der tatsächlichen Zitierungen gefunden werden, wodurch der Einfluss der betrachteten Journale natürlich unterbewertet wird.

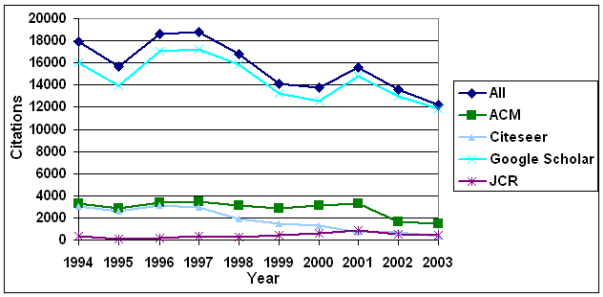
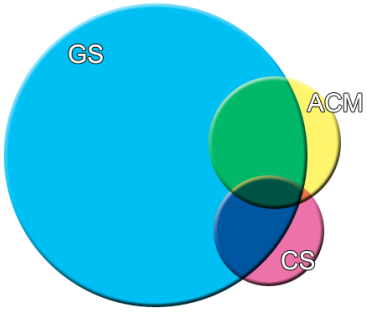


Abbildung 7.1: Zeitliche Entwicklung der Zitierungszahlen der betrachteten Publikationsorte unterteilt nach der jeweiligen Datenquelle, gruppiert nach dem Erscheinungsjahr der zitierten Publikation.

Weiter ist zu bemerken, dass `Google Scholar` mit Abstand die größte Abdeckung unter den Datenquellen hat und damit die GS-Zitierungen eine Größenordnung größer sind, als die der anderen Webquellen. Bei `ACM` liegt dies daran, dass nur Zitierungen von bei



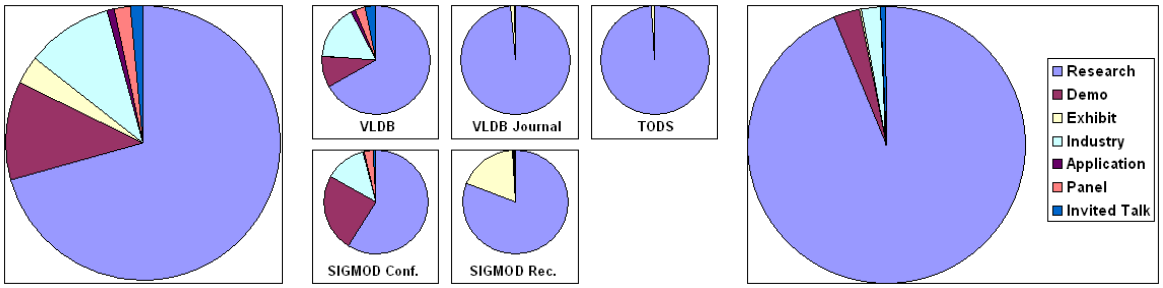
Menge	#Zit.	Menge	#Zit.
GS	145569	$GS \setminus (ACM \cup CS)$	113864
ACM	28525	$ACM \setminus (GS \cup CS)$	5369
CS	17988	$CS \setminus (GS \cup ACM)$	5801
$GS \cap ACM$	22865	$(GS \cap ACM) \setminus CS$	19809
$GS \cap CS$	11896	$(GS \cap CS) \setminus ACM$	8840
$ACM \cap CS$	3347	$(ACM \cap CS) \setminus GS$	291
$GS \cup ACM \cup CS$	157030	$GS \cap ACM \cap CS$	3056

Abbildung 7.2: Überschneidungen der gefundenen Zitierungen in den Datenquellen. GS = Google Scholar, CS = Citeseer.

`ACM` erschienen Publikationen berücksichtigt werden. Die gefundenen Zitierungen sind zum größten Teil eine Teilmenge der GS-Zitierungen, da dort auch die meisten ACM-Publikationen zu finden sind. `Citeseer` enthält ähnlich viele Zitierungen wie `ACM`, doch darunter sind auch einige die bei `Google Scholar`, aufgrund des breiteren wissenschaftlichen Spektrums, nicht zu finden sind. Allerdings sind bei `Citeseer` wenige jüngere Publikationen aufgelistet (vgl. Abbildung 3.1, Seite 22), wodurch die Zitierungen ab 2000 stark gegen null gehen. Auch die übrigen Zitierungszahlen gehen in der jüngeren Zeit zurück, da die Halbwertszeit ca. vier bis fünf Jahre beträgt und damit erst mehr als die Hälfte der potentiell zitierenden Papiere zum Analysezeitpunkt veröffentlicht waren. Generell ist aber ein mindestens gleich bleibender Trend zu erkennen, wenn man berücksichtigt, dass jüngere Papiere in Zukunft wahrscheinlich öfter zitiert werden als ältere.

Abbildung 7.2 visualisiert die Überschneidungen der Zitierungen der einzelnen Quellen. Wie zu erwarten nehmen die bei *Google Scholar* gefundenen Zitierungen den größten Teil der Fläche ein. Die Fläche der *Citeseer*-Zitierungen fällt aufgrund der Abnahme in jüngerer Zeit etwas kleiner aus als *ACM*. Der überwiegende Anteil, ca. 80%, von *ACM* Zitierungen lässt sich auch in *Google Scholar* wiederfinden. Der ausstehende Anteil ist größtenteils auf Fehler bei der Duplikatbeseitigung, aber auch der Problematik der möglichst vollständigen Webdatenextraktion zurückzuführen. Dagegen sind nur ca. 66% der *Citeseer*-Zitierungen auch bei *Google Scholar* auffindbar. Für etwa die Hälfte der restlichen Zitierungen gilt das gleiche Argument wie für *ACM*. Die andere Hälfte, also etwa 16% aller *Citeseer*-Zitierungen ist auf das etwas weitere Spektrum von *Citeseer* zurückzuführen. Die Schnittmenge von *ACM* und *Citeseer* ist dagegen vergleichsweise klein und auch fast vollständig in *Google Scholar* enthalten.

Offensichtlich hat *Google Scholar* mit über 90% aller Zitierungen auch den meisten Einfluss auf die Zitierungsanalyse. Die zusätzliche Nutzung *ACM* und *Citeseer* stellt dagegen nur einen geringen Mehrwert dar. Trotzdem ist die Nutzung sinnvoll, da damit fehlende oder fehlerhafte Daten ausgeglichen werden können.



(a) Anteil der verschiedenen Publikationstypen über alle betrachteten Veröffentlichungen

(b) Anteil von wissenschaftlichen Papieren an den Gesamtveröffentlichungen

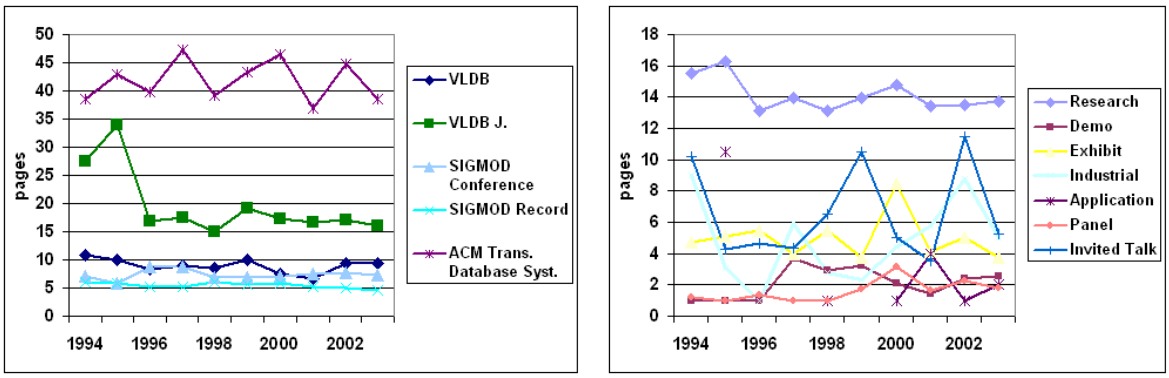
(c) Anteil von Zitierungen von wissenschaftlichen Papieren

Abbildung 7.3: Vergleich des Anteils verschiedener Publikationstypen.

7.2 Publikationstypen

In Abbildung 7.3a sind die Anteile der verschiedenen Publikationstypen für die betrachteten Konferenzen und Journale dargestellt. Es wird deutlich, dass nur Arbeiten über wissenschaftliche Erkenntnisse, Demonstrationen und industrielle Arbeiten und Anwendungen eine bedeutendere Rolle spielen. Eingeladene Vorträge und Diskussionen sind aufgrund ihrer Natur und da sie nur auf Konferenzen vorkommen sehr gering vertreten. Offensichtlich stellen, mit annähernd drei viertel aller Publikationen, die Arbeiten wissenschaftlicher Natur den allergrößten Anteil an allen Veröffentlichungen. Abbildung 7.3b veranschaulicht die Verhältnisse der verschiedenen Publikationen getrennt nach Veröffentlichungsort. Hier wird besonders deutlich, dass nichtwissenschaftliche Papiere vor allem auf Konferenzen vorgestellt werden. Der Anteil rein wissenschaftlicher Veröffentlichungen liegt dort knapp über 60%. In den Journalen spielen dagegen, bis auf SIGMOD Record wo auch einige demonstrative Papiere erscheinen, andere Typen so gut wie keine Rolle. Dieses Verhalten ist problematisch, wenn man Abbildung 7.3c betrachtet. Wissenschaftliche Papiere erhalten über 93% aller Zitierungen, obwohl sie gerade einmal 70% der Publikationen stellen. Das bedeutet dass wissenschaftliche Papiere im Schnitt 84 mal zitiert werden, die restlichen jediglich 13 mal.

Einen Erklärungsansatz bietet möglicherweise der Vergleich der Textlängen der Veröffentlichungen. Schließlich bieten längere Texte in der Regel einen höheren Informationsgehalt und eignen sich daher besser zum Zitieren. Abbildung 7.4 stellt die durchschnittlichen Textlängen einmal getrennt nach Publikationsorten in 7.4a und einmal nach Publikationstypen in 7.4b gegenüber. Letzteres Diagramm besitzt teilweise Lücken, für Jahre in denen keine einzige Publikation (für die die Textlänge bekannt ist) passenden Typs erschienen ist und ist generell für die schwach vertretenen Typen nur exemplarisch zu sehen, da



(a) nach Publikationsorten

(b) nach Publikationstypen

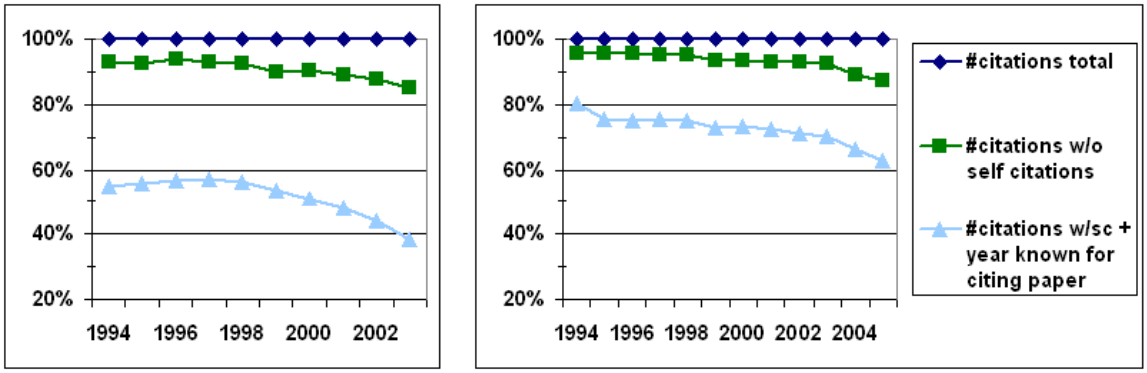
Abbildung 7.4: Durchschnittliche Länge von Publikationen.

die Zahlen dort teilweise über einzelne oder Paare von Papieren gemittelt wurden. Abbildung 7.4a veranschaulicht deutlich, dass auf Konferenzen eher kürzere Publikationen veröffentlicht werden, wogegen bei den Journalen *VLDB* und *ACM TODS* deutlich höhere Textlängen vorherrschen. In Abbildung 7.4b wurden wieder alle Publikationsorte vereinigt. Da die Journale nur vergleichsweise wenige Erscheinungen pro Jahr besitzen, sinkt auch die durchschnittliche Textlänge entsprechend ab. Aus dieser Abbildung wird deutlich, dass wissenschaftliche Papiere im Schnitt um die 15 Seiten lang sind und damit deutlich länger als alle anderen Publikationstypen. Besonders kurz sind Demonstrations- und Diskussionspapiere. Damit ist die Theorie teilweise bestätigt. Nur *SIGMOD Record* bildet hier eine Ausnahme. Trotz kurzer Veröffentlichungen, werden diese vergleichsweise oft zitiert. Laut [RT05] liegt das an einigen oft zitierten Übersichtspapieren, welche in die Statistik entsprechend eingehen.

Diese Erkenntnisse haben entscheidende Auswirkungen auf die Zitierungsanalyse. Schließlich kommen in Journalen beinahe ausschließlich wissenschaftliche Papiere vor, in Konferenzen dagegen nur zu zwei Dritteln. Will man den wissenschaftlichen Einfluss von Journalen und Konferenzen miteinander vergleichen, müssen die anderen Papiere herausgerechnet werden. Aus diesem Grund, werden für den Rest des Kapitels nur noch wissenschaftliche Veröffentlichungen berücksichtigt.

7.3 Selbstzitierungen

Ein weiterer Punkt, der zur Verzerrung von Zitierungsstatistiken führen kann, sind Selbstzitierungen [MM96]. Eine Zitierung wird dann als Selbstzitierung gezählt, wenn mindestens ein Autor des zitierenden Papiers auch an der zitierten Arbeit beteiligt war. Aller-



(a) *[2005]* Quelle: [RT05]

(b) *[2008]*

Abbildung 7.5: GS-Selbstzitierungen, normalisiert auf die Komplettanzahl von GS-Zitierungen.

dings gibt es verschiedene Gründe, warum Autoren ihre eigenen Arbeiten zitieren. Zum einen können Selbstzitierungen dazu dienen den scheinbaren Einfluss des zitierten Papiers zu erhöhen. In [MM96] führen die Autoren dagegen aus, dass meist die gleichen Gründe für die Zitierung der eigenen Arbeit, wie für die anderer Autoren sprechen, welche aber auch wieder politisch motiviert sein können. Eine weitere Ursache für Selbstzitierungen, sind Weiterentwicklungen und Erneuerungen vergangener Erkenntnisse. In allen Fällen verzerren diese Zitierungen das Analyseresultat. Besonders deutlich würden diese Verzerrungen bei Auswertungen über den Einfluss von Autoren oder Instituten.

Abbildung 7.5 zeigt, welchen Anteil die Selbstzitierungen an allen Zitierungen insgesamt besitzen. Sie machen in *[2005]* im Schnitt circa 9% und in *[2008]* circa 6% aus. Diese Veränderungen bedeuten allerdings nicht, dass weniger selbstzitiert wird. Vielmehr sind die absoluten Zahlen in der Zeit gewachsen, wodurch die Selbstzitierungen nicht proportional mit wachsen. So ist auch die Zunahme der Selbstzitierungen zum Ende des Analysezeitraums in beiden Datenmengen zu erklären. Selbstzitierungen finden in der Regel schneller statt, da die Autoren ihre Forschung in der Zeit bis zur Veröffentlichung schon fortsetzen und an einer neuen Publikation arbeiten können. Nach mehreren Jahren gleicht sich das Verhältnis aus. Dieses Verhalten verdeutlicht zusätzlich die Problematik bei der Nutzung von Selbstzitierungen. Denn ein weiteres in [MM96] genanntes Problem der Zitierungsanalyse liegt in unterschiedlichen Zitierungsraten und Zitierungsgeschwindigkeiten. Schnellere Selbstzitierungen verzerren auch hier das Ergebnis. Das gleiche gilt für die Berechnung der Impaktfaktoren, für die nur ein begrenzter Zeitraum nach Veröffentlichung berücksichtigt wird. Daher werden künftige Diagramme von Selbstzitierungen bereinigt sein.

Weiterhin ist in Abbildung 7.5 das Verhältnis von Zitierungen, für deren zitierendes Pa-

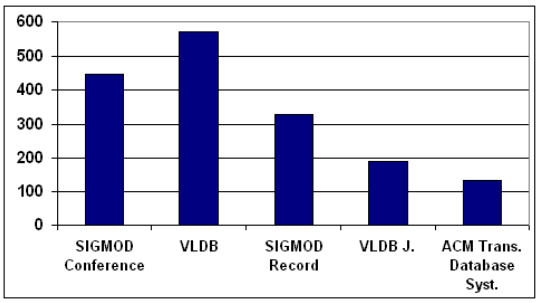
pier das Erscheinungsjahr bekannt ist, dargestellt. Dieses wird schließlich für die Impaktfaktoren benötigt. Auch hier ist ein starker Anstieg in den Daten von *[2008]* im Vergleich zu *[2005]* zu verzeichnen. Während der Anteil *[2005]* noch gänzlich unter 60% lag, liegt er in den *[2008]*er Daten bei im Schnitt 70%. Dieser Anstieg ist durch die Arbeitsweise von `Google Scholar` zu erklären. Zum einen sind, wie in Kapitel 4 ausgeführt, für oft zitierte Publikationen nicht alle Zitierungen vollständig extrahierbar, sondern nur für diese bei denen das Erscheinungsjahr bekannt ist. Zum andern bezieht `Google Scholar` die Jahresangaben scheinbar aus den Zitierungsinformationen anderer Papiere. Das bedeutet zum einen, dass wenn *[2005]* viele der zitierenden Papiere selbst noch nicht zitiert wurden, erhielten sie kein Jahr. Das erklärt auch den Rückgang des Anteils für jüngere Jahrgänge. Außerdem scheint `Google Scholar` seine Extraktionstechnik überarbeitet zu haben. So waren seit dem anfänglichen Testergebnissen zu dieser Arbeit Mitte 2007 bis zu den finalen Ergebnissen ein deutlicher Anstieg der Einträge mit Jahreszahlen wahrnehmbar. Anders ist auch nicht zu erklären, dass selbst der *[2008]* er Wert für 2005 noch über den besten Werten von *[2005]* liegt.

An dieser Stelle soll aber auch noch darauf hingewiesen werden, dass das Ignorieren von Selbstzitierungen problematisch sein kann. Wird eine Arbeit öfters von den Autoren zitiert, weil es wirklich die Grundlage für ein Spektrum an weiteren Arbeiten geliefert hat, zum Beispiel bei einem Joint Venture von Autoren verschiedener Institute, so könnte dem Papier doch intuitiv eine gewisse Bedeutung beigemessen werden. Andere Autoren zitieren dagegen vielleicht eher die darauf aufbauenden Arbeiten. Das Papier würde damit als relativ bedeutungslos eingeschätzt, obwohl es einen relativ großen Einfluss hatte.

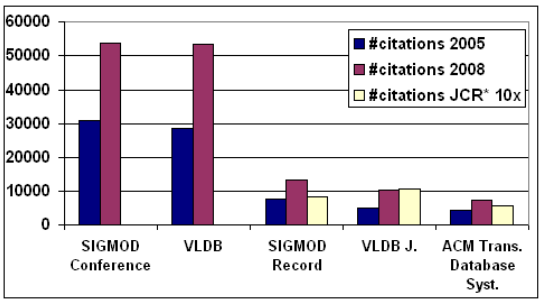
7.4 Zitierungszahlen

Beim Vergleich der absoluten Zahlen von erschienen wissenschaftlichen Publikationen in Abbildung 7.6a fällt auf, dass die meisten Erscheinungen auf Konferenzen auftreten und das obwohl circa ein Drittel deren Publikationen ignoriert werden, da sie nicht in die Kategorie der wissenschaftlichen Arbeiten eingeordnet worden. Unter den Journalen erreicht nur *SIGMOD Record* ähnlich hohe Erscheinungszahlen, was zum großen Teil an den kurzen Beiträgen liegt. Obwohl *VLDB* in der Zeit mehr Veröffentlichungen besitzt, wurden diese insgesamt knapp weniger oft zitiert, als die der *SIGMOD Conference*. Die Veröffentlichungen in Journalen erhielten deutlich weniger Zitierungen. Ein deutlicherer Vergleich macht eine Normalisierung der Zitierungen notwendig, zu sehen in Abbildung 7.6c. Daran ist eindeutig zu erkennen, dass *SIGMOD Conference* mit Abstand am meisten zitiert wird. *ACM TODS* ist von den Journalen am besten bewertet, da auf die wenigen Veröffentlichungen relativ viele Zitierungen fallen.

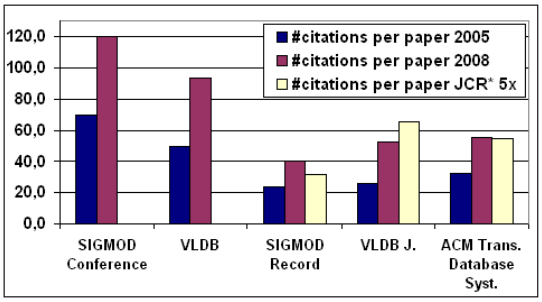
Im Vergleich zu den *[2005]*er Daten, sind die Zahlen *[2008]* zwar deutlich angestiegen, allerdings proportional. Es sind keinerlei Umkehrungen der Größenverhältnisse zu er-



(a) Anzahl erschienener wissenschaftlicher Publikationen



(b) Anzahl Zitierungen

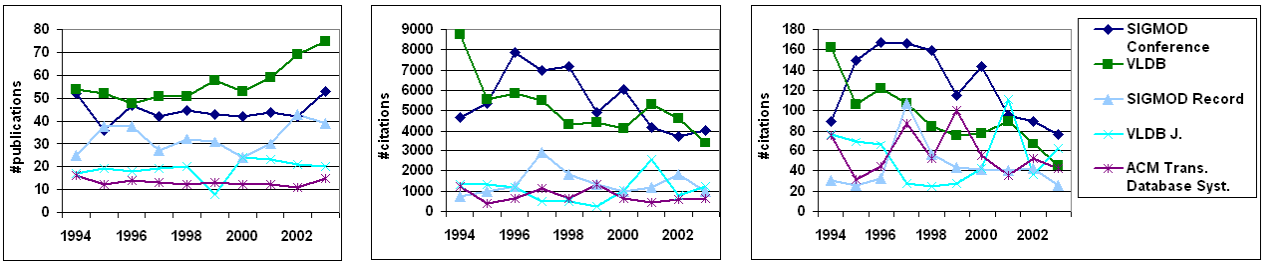


(c) Anzahl Zitierungen pro Publikation

Abbildung 7.6: Vergleich der absoluten Anzahl von erschienen Publikationen zwischen 1994 und 2003 und die erhaltenen Zitierungen.
JCR Daten von 2000 bis 2003, Quelle [oK06], zur Übersichtlichkeit entsprechend vergrößert.

kennen. Dies verdeutlicht, dass absolute Zahlen allein keinen großen Informationsgehalt besitzen. Demnach ist die reine Feststellung, dass *SIGMOD Conference* im Schnitt 60 bzw. 120 Zitierungen pro Publikation erhält relativ wertlos. Erst im Vergleich zu anderen Publikationsorten der Zahlen auf der gleichen Datengrundlage, bekommt diese Größe eine Aussagekraft.

Die gleiche Überlegung gilt für die Daten der JCR. In [oK06] werden nur Journale berücksichtigt und auch nicht alle Jahrgänge. Entsprechend niedrig fallen auch Zitierungszahlen aus. In Abbildung 7.6b sind die Zahlen zur besseren Darstellung verzehnfacht, in 7.6c verfünffacht. Und auch hier sind die Ergebnisse, relativ gesehen, ziemlich ähnlich zu *[2005]* und *[2008]*. Bei genauerer Betrachtung fällt allerdings auf, dass *ACM TODS* und *VLDB Journal* die Plätze getauscht haben und *SIGMOD Record* noch schlechter bewertet wird. Das liegt aber ausschließlich daran, dass in den JCR für *SIGMOD Record* erst ab 2000 Daten vorhanden sind und daher alle JCR Daten auf 2003 bis 2005 begrenzt wurden. Interessanterweise erhält man bei Begrenzung der *[2008]*er Daten auf den gleichen Zeit-



(a) Anzahl erschienener wissenschaftlicher Publikationen (b) Anzahl Zitierungen insgesamt (c) Durchschnittliche Anzahl Zitierungen je Publikation

Abbildung 7.7: Zeitliche Entwicklung der Anzahl von Veröffentlichungen und erhaltene Zitierungen

raum, relativ gesehen, fast exakt die gleichen Ergebnisse.

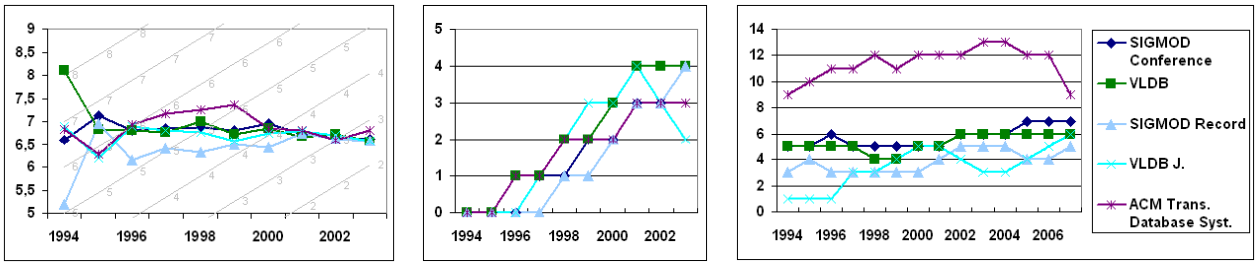
Die Begrenzung der JCR auf Journale und der damit verbundene Ausschluss von Konferenzen, hat wesentliche Nachteile für die Einflussanalyse wissenschaftlicher Publikationsorte. Offensichtlich erzielen Konferenzen eine wesentlich höhere Anzahl von Zitierungen als Journale und besitzen damit vermutlich auch einen höheren Einfluss. Für andere wissenschaftliche Bereiche mag diese Einschränkung berechtigt sein, für die Informatik und speziell den hier betrachteten Ausschnitt, ergibt sich dadurch aber ein falsches Bild.

Zeitliche Entwicklung

In Abbildung 7.7 sind die zeitliche Entwicklung der erschienen Publikationen und die Anzahl der Zitierungen gegenübergestellt. *VLDB* hat im dargestellten Zeitraum einen deutlichen Zuwachs von akzeptierten Publikationen zu verzeichnen, während sich die anderen Journale und Konferenzen in einem mehr oder weniger engen Rahmen bewegen. Trotzdem erhielten *VLDB*-Publikationen meist weniger Zitierungen, als die der *SIGMOD Conference*. Das spiegelt sich auch in der durchschnittlichen Zitierungszahl in Abbildung 7.7c wider. *VLDB* liegt hier meist deutlich unter der *SIGMOD Conference* und teilweise auch unter *ACM TODS* und *VLDB Journal*.

Die Zitierungszahlen nehmen zum Ende hin deutlich ab, was allerdings damit zusammenhängt, dass die Zitierungswahrscheinlichkeit für die neueren Papiere noch deutlich höher liegt, das heißt, es werden zukünftig noch weit mehr Publikationen veröffentlicht, die die neueren Papiere zitieren, als ältere. In [RT05] fiel der Abstieg zum Ende hin noch viel deutlicher aus.

Die Journale besitzen eine deutlich instabilere Zitierungskurve, was speziell in Abbildung



(a) Durchschnittliches Alter der Zitierungen, aus Sicht der zitierten Papiere

(b) Halbwertszeit (Beschränkung auf Zitierungen von wissenschaftlichen Papieren)

(c) Halbwertszeit (ohne Beschränkung)

Abbildung 7.8: Vergleich von Zitierungsalter und Halbwertszeiten.

7.7c auffällt. Die Kurven der Konferenzen verhalten sich wesentlich ruhiger und haben nur wenige Sprünge. Dies deutet darauf hin, dass die Zitierungen einzelner Journaljahrgänge mehr gestaucht sind, als andere. Fehlt in einem Jahrgang ein besonders oft zitiertes Papier, fällt die Gesamtzitierungszahl entsprechend ab. Mehr zu Stauchung in Abschnitt 7.6. Ein anderer Grund liegt in der recht geringen Zahl der Veröffentlichungen, dadurch machen sich kleinere Abweichungen natürlich deutlicher bemerkbar.

7.5 Zitierungsalter

Das Alter von Zitierungen kann mehrere Aussagen besitzen. Zum einen deuten lange Abstände zwischen Veröffentlichung und Zitierung auf grundlegende Erkenntnisse in den Papieren hin, die auch nach längerer Zeit noch gültig und aktuell sind. Zum anderen sind schnelle Zitierungen ein Indiz für einen starken Einfluss auf die aktuelle Forschung.

Abbildung 7.8a stellt das durchschnittliche Alter von Zitierungen von in den entsprechenden Jahrgängen und Publikationsorten erschienen Arbeiten dar. Da für jüngere Papiere ein kürzerer Zeitraum für mögliche Zitierungen vor liegt, nehmen die Werte ab. Würde der betrachtete Zeitraum bis 2008 ausgedehnt werden, würden sie in Null konvergieren. Zur besseren Darstellung wurde das Diagramm gedreht abgebildet. Es ist zu erkennen, dass sich die unterschiedlichen Publikationsorte, hinsichtlich des Zitierungsalters, gar nicht so sehr von einander unterscheiden. Bis auf 1994 liegen alle innerhalb eines Jahres. Trotzdem wird deutlich, dass *SIGMOD Record* eher von kurzfristigen Zitierungen und *ACM TODS* eher von langfristigeren Zitierungen geprägt ist. Zudem fällt auch hier wieder auf, dass sich die Kurven der Konferenzen wesentlich stabiler verhalten, ihr Abstieg verläuft beinahe linear.

Einen besseren Vergleich bietet die *zitierte Halbwertszeit* (engl. cited halflife). Die zitierte Halbwertszeit für ein Jahr x und einen Publikationsort y ist definiert als der Zeitraum, zurückgehend von Jahr x, so dass Zitierungen von Arbeiten welche in diesem Zeitraum in y veröffentlicht wurden, in x mindestens die Hälfte aller Zitierungen des Publikationsortes ausmachen. Zum Beispiel fielen im Jahr 2003 7645 Zitierungen auf *SIGMOD Conference*, davon zitieren 4076 Arbeiten aus dem Zeitraum 1997 bis 2003, zwischen 1998 bis 2003 aber nur 3336, daher ist die zitierte Halbwertszeit in diesem Fall 6.

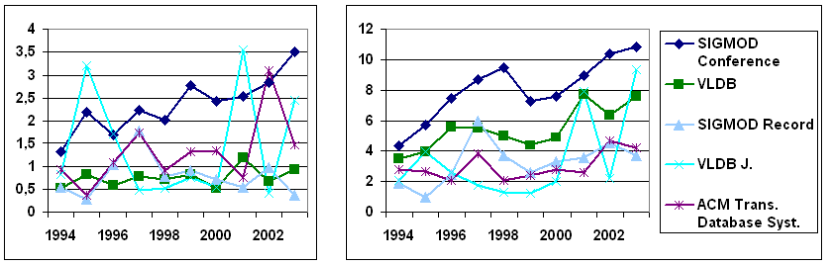
In Abbildung 7.8b und 7.8c sind die zeitlichen Verläufe der zitierten Halbwertszeit abgebildet. Abbildung 7.8b ist bei den zitierten Papieren nur auf wissenschaftliche Arbeiten begrenzt. Da die Kategorisierung des Publikationstyps, wie weiter oben beschrieben, nur für den begrenzten Zeitraum 1994 bis 2003 in den Daten enthalten ist, stellen diese Daten für langfristige Analyse der Halbwertszeit keine gute Grundlage dar. Daher wurde für Abbildung 7.8c diese Einschränkung entfernt.

In dieser Darstellung sticht *ACM TODS* besonders durch seine langfristigen Zitierungen hervor. Hob sich das durchschnittliche Zitierungsalter nur marginal von dem der anderen Publikationsorte ab, beträgt die Halbwertszeit im Schnitt das Doppelte der anderen. Betrachtet man die zitierten *ACM TODS* Arbeiten genauer, fällt auf, dass für einige Jahrgänge einige wenige Publikationen auch nach längerer Zeit, als über fünf oder sechs Jahre, besonders viele Zitierungen erhalten. Erst im jeweils jüngeren Zeitraum tauchen viele verschiedene Arbeiten pro Jahrgang auf. Bei den langfristig zitierten Arbeiten handelt es sich daher eher um Klassiker, die hohe zitierte Halbwertszeit von *ACM TODS* deutet darauf hin, dass dort überproportional viele spätere Klassiker publiziert wurden.

Die beiden weiter betrachteten Journale fallen dagegen sehr viel niedriger aus und liegen damit sogar unter den Konferenzen. Der anfängliche niedrige Wert für *VLDB Journal* ist damit zu erklären, dass die erste Ausgabe aus dem Jahr 1992 stammt und damit die Halbwertszeit gar nicht höher liegen kann. Der steilere Anstieg dagegen, könnte auf eine Tendenz hindeuten, dass auch das *VLDB Journal* zukünftig über den Konferenzen liegen wird. Die beiden betrachteten Konferenzen ähneln sich auffallend, was auf ähnliche Aktualität der Arbeiten und gleiche Zitierungsgewohnheiten schließen lässt.

Als Maß für die Zitierungsgeschwindigkeit dient der *Immediacy Index*. Dieser wird berechnet durch die Division von der Anzahl von Zitierungen die die Arbeiten eines Publikationsortes eines Jahrgangs im gleichen Jahr erhielten, durch die Anzahl der betroffenen Arbeiten. Z.B. wurden auf der *SIGMOD Conference* im Jahr 2003 53 wissenschaftliche Arbeiten veröffentlicht. Diese wurden noch im gleichen Jahr 186 mal zitiert. Damit ergibt sich für den *Immediacy Index* ein Wert von $3,5$.

Das problematische an dieser Berechnung ist, dass in den zugrunde liegenden Daten der Erscheinungszeitpunkt der Publikationen nur auf das Jahr genau vor liegt. Das bedeutet, für Zitierungen im gleichen Jahr verbleibt unterschiedlich viel Zeit. Die Journale erscheinen meist in mehreren Ausgaben pro Jahr, wobei die Anzahl und der Erscheinungs-



(a) klassisch (b) modifiziert (Zitierungen um ein Jahr versetzt)

Abbildung 7.9: Immediacy Index

zeitpunkt nicht immer konstant ist. Bei den Konferenzen wird der Unterschied noch deutlicher. Konferenzen finden nur einmal jährlich statt, dabei *VLDB* meist im September, die *SIGMOD Conference* dagegen meist schon im Juni. Dieses Verhalten spiegelt sich auch im Immediacy Index wieder, siehe Abbildung 7.9a. Während die *SIGMOD Conference* schon im gleichen Jahr einige Zitierungen erhält, befindet sich die Konferenz *VLDB* aufgrund des kurzen Zeitraums von drei verbleibenden Monaten am unteren Ende.

Um die unterschiedlichen Veröffentlichungszeitpunkte etwas auszugleichen, ist es notwendig die klassische Definition des Immediacy Index etwas zu modifizieren und statt der Zitierungen im jeweils gleichen Jahr, die des Folgejahres zu nutzen. Beispielsweise wurden die 53 *SIGMOD Conference* Publikationen von 2003 im Jahr 2004 576 mal zitiert. Damit erhält der modifizierte Immediacy Index einen Wert von $10,9$. Die weiteren Ergebnisse sind in Abbildung 7.9b veranschaulicht. Die Spitzen der Graphen der Journale, speziell *VLDB Journal* sind trotzdem noch durch abweichende Veröffentlichungszeitpunkte oder Anzahl von Ausgaben im jeweiligen Jahrgang begründet. Trotzdem ist klar zu erkennen, dass die Konferenzen im Schnitt deutlich mehr sofortige Zitierungen erhalten, als Journale. Verglichen mit der Durchschnittszahl der Zitierungen (Abb. 7.7c), verhalten sich die Graphen proportional zu einander. Lediglich *ACM TODS* liegt mit seiner, zu den anderen Journalen vergleichsweise hohen, Durchschnittszitierungszahl beim Immediacy Index unter den Erwartungen, was auf die deutlich längere zitierte Halbwertszeit zurückzuführen ist. Die anderen beiden Journale verhalten sich demnach hinsichtlich der zeitlichen Zitierungsgewohnheiten nicht wesentlich von den Konferenzen, da der niedrige Index mit der niedrigen Gesamtzitierungszahl zusammenhängt.

7.6 Skew

Unter Skewness versteht man in der Statistik die Abweichung von der Gleichverteilung. Bei der Zitierungsanalyse spricht man von Citationskew, also der Ungleichverteilung von

Zitierungen auf Publikationen. Ein niedriger Citationskew bedeutet, dass die meisten Publikationen ähnlich oft zitiert wurden. Ein hoher Skew heißt dagegen, dass einige wenige Publikationen sehr oft, die anderen sehr wenig zitiert wurden.

Als Möglichkeit den Citationskew zu messen dient der Ginikoeffizient. Zur Berechnung für diskrete Verteilungen kann die Brownformel zur Annäherung der Lorenzkurve herangezogen werden:

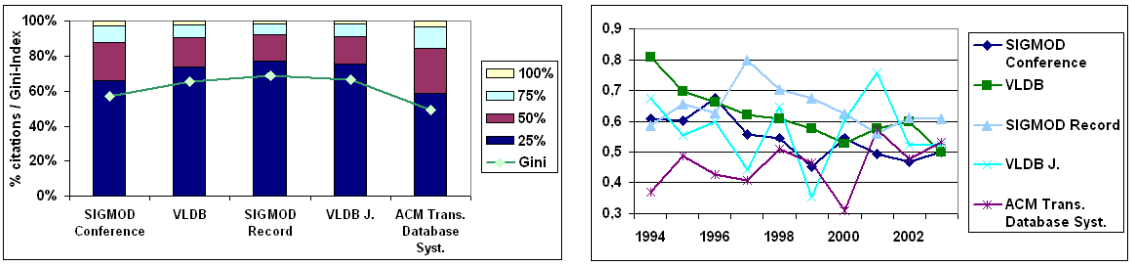
$$
\begin{aligned}
\text{Sei} \quad & n && \text{die Anzahl der betrachteten Objekte} \\
& x_k && \text{die Größe des } k. \text{ gemessenen Objektes, für } k \in [1,n] \\
& y_k && \text{der gemessene Wert für das } k. \text{ Objekt} \\
\text{derart, dass} \quad & && \frac{y_k}{x_k} \geq \frac{y_{k-1}}{x_{k-1}}. \\
\text{Seien definiert:} \quad & X_k = && \frac{\sum_{i=1}^{k} x_i}{\sum_{j=1}^{n} x_j} \text{ und } Y_k = \frac{\sum_{i=1}^{k} y_i}{\sum_{j=1}^{n} y_j}, \\
\text{dann ist} \quad & G = && 1 - \sum_{l=1}^{n-1} (X_{l+1} - X_l)\,(Y_{l+1} + Y_l) \text{ der Ginikoeffizient.}
\end{aligned}
$$

Da hier im speziellen Fall nur einzelne Publikationen betrachtet werden, ist $x_k = 1$ und damit $X_k = k/n$. Die y_k sind die Zitierungszahlen der Arbeiten, aufsteigend sortiert. Damit lässt sich die Formel vereinfachen:

$$
\begin{aligned}
G = \quad & 1 - \frac{1}{n} * \sum_{l=1}^{n-1} Y_{l+1} + Y_l \\
\text{oder} \quad G = \quad & 1 - \frac{1}{n * Gesamtzitierungen} \sum_{l=1}^{n} (2(n-l)+1)\, y_l
\end{aligned}
$$

Mit dieser Formel lässt sich der Ginikoeffizient für alle Konferenzen bzw. Journale einfach berechnen. Der Wert liegt zwischen 0 und 1. Ein Wert von 0 steht für eine Gleichverteilung aller Zitierungen, der echte Grenzwert 1 die Konzentration aller Zitierungen auf ein Papier. Abbildung 7.10a zeigt den Gini-Index für die betrachteten Publikationsorte. Zudem werden die Anteile der jeweils Top-25%-Arbeiten, 25%-50% usw., an den erhaltenen Zitierungen dargestellt. Diese Grafik mit den *[2008]*er Daten ist nahezu identisch mit der aus [RT05] mit den *[2005]*er Daten. Einzig *ACM TODS* liegt in *[2008]* marginal (ca. 4%) unter den alten Daten. Das bedeutet, dass die seit 2005 neu hinzugekommenen Zitierungen das Ergebnis nur unwesentlich beeinflussen.

Die besten 25% Publikationen erhalten im Schnitt 70% aller Zitierungen, während die unteren 25% nicht einmal 3% erhalten. Die ungleichste Verteilung herrscht bei *SIGMOD Record*. Wie in [RT05] argumentiert wird, beruht dies auf einigen sehr oft zitierten Überblickspapieren. Die restlichen Papiere werden wegen ihrer Kürze eher weniger oft zitiert.



(a) Verhältnis von Publikationen zu erhaltenen Zitierungen, Ginikoeffizient

(b) zeitliche Entwicklung des Ginikoeffizienten

Abbildung 7.10: Citationskew in den verschiedenen Publikationsorten. Gini-Index= 0% bedeutet Gleichverteilung der Zitierungen, ≈ 100% bedeutet, eine Publikation erhielt alle Zitierungen

Dadurch entsteht eine starke Stauchung der Zitierungsverteilung. Am gleichmäßigsten sind die Zitierungen bei *ACM TODS* verteilt, obwohl auch hier mit einem Gini-Index von 49,6% immer noch eine stark ungleiche Verteilung vorliegt. *VLDB* und *VLDB Journal* ähneln sich auffallend, wogegen die *SIGMOD Conference* deutlich unter *SIGMOD Record* und auch unter *VLDB* liegt. Da *SIGMOD Conference* trotz weniger Veröffentlichungen höhere Zitierungszahlen erhält (vgl. Abb. 7.7), liegt die Überlegung nahe, dass dort mehr oft zitierte Papiere veröffentlicht werden, wodurch die Stauchung geringer ausfällt, die kleinere Anzahl von Arbeiten aber insgesamt mehr Zitierungen erhalten als *VLDB*.

Abbildung 7.10b visualisiert die zeitlichen Veränderungen im Ginikoeffizienten. Da in den einzelnen Jahrgängen vergleichsweise wenig Papiere zur Berechnung zur Verfügung stehen, ist der Verlauf vor allem bei Journalen recht sprunghaft. Ein sprunghafter Anstieg kann im Wesentlichen zwei Ursachen haben. Zum einen kann ein besonders oft zitiertes Papier hinzukommen, während im Vorjahr alle Papiere eher mittelmäßig oft zitiert wurden. Zum anderen, können auch mehrere oft zitierte Papiere wegfallen, so dass nur noch wenige vorhanden sind. In beiden Fällen ist eine Auswirkung auf die Entwicklung der Zitierungszahlen zu erwarten, wobei die Änderungen aufgrund anderer Einflüsse nicht proportional sind. Im ersteren Fall wären die Zitierungen unerwartet hoch, im zweiten eher niedriger. Die Beobachtungen scheinen den Verdacht zu bestätigen. Den höchsten Peak hat das *VLDB Journal* im Jahr 2001 zu verzeichnen und tatsächlich hat 2001 auch die Durchschnittszitierungskurve ein Peak (vgl. Abb. 7.7c). Das gleiche gilt für 1997 bei *SIGMOD Record*. Den umgedrehten Fall gibt es bei *ACM TODS* in den Jahren 1995, 1998 und 2001 zu sehen. Jeweils ist der Gini-Index auf einem Hoch, während die Zitierungszahlen recht niedrig sind. Für Tiefs des Ginikoeffizienten sind allerdings keine besonderen Änderungen zu beobachten. Die gemachten Beobachtungen sind allerdings aufgrund des relativen engen Sichtfelds nicht beweisbar. Zum Beispiel scheint das Hoch des Giniindexes bei *VLDB Journal* in 1998 keinerlei Auswirkungen auf die Zitierungen zu

haben, dagegen gibt es auch Peaks der Zitierungszahlen die sich nicht in den Giniwerten widerspiegeln. Die Stauchung von Zitierungen ist nur ein Einflussfaktoren von vielen und die konkrete Wirkung müsste mit wesentlich mehr Daten unterlegt werden.

7.7 Impaktfaktor

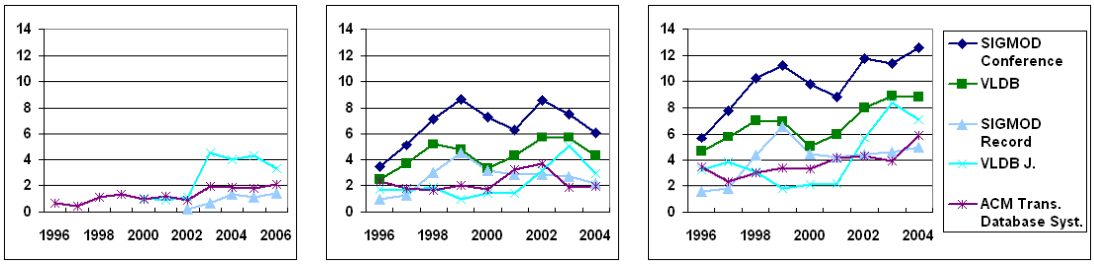
Der Impaktfaktor (IF), auch bekannt als Journal Impact Factor (JIF), ist ein oft und gern genutztes Maß zur Messung von wissenschaftlichen Einflüssen von einerseits Publikationsorten, aber auch Autoren und Instituten. Er gibt für ein Jahr die durchschnittliche Anzahl von Zitierungen der Vorjahr zurück. Der Impaktfaktor für das Jahr x und den berücksichtigten Zeitraum $y \geq 1$ Jahre ergibt sich folgendermaßen:

$$IF_y(x) = \frac{\text{Zitierungen im Jahr } x \text{ von Arbeiten der Jahre } x-y \text{ und } x-1}{\text{Anzahl zwischen } x-y \text{ und } x-1 \text{ erschienenen Arbeiten}}$$

Der meist genutzte Wert für y, unter anderen in den *JCR*, ist ein Zeitraum von 2 Jahren. Das heißt es werden nur die Zitierungen von Publikationen berücksichtigt, welche in den beiden Vorjahren erschienen sind. Sofortige Zitierungen werden demnach auch ignoriert, wodurch der Erscheinungszeitpunkt innerhalb des Jahres, welcher sich sehr nachteilig auf den klassischen Immediacy Index auswirkt (vgl. Abschnitt 7.5), eine untergeordnete Rolle spielt. Allerdings kann diese Berechnung Journale mit größeren durchschnittlichen Zitierungsalter benachteiligen, da die späteren Zitierungen in keinen Impaktfaktor mehr eingehen.

In Abbildung 7.11 sind die Impaktfaktoren für mehrere Datenbestände für jeweils einen Zitierungszeitraum von 2, 5 und 10 Jahren angegeben, wobei in den *JCR* nur die erste, *[2005]* nur für die ersten beiden Varianten gelistet wird. Die JCR-Daten sind, wie schon die absoluten Zitierungszahlen, kaum vergleichbar mit den Daten von *[2005]* und *[2008]*. Zum einen sind die Konferenzen ausgelassen, was besonders nachteilig ist, da die Konferenzen die Mehrheit der Zitierungen erhalten. Zum anderen sind die Impaktfaktoren der Journale teilweise sogar nur ab 2000 oder gar 2002 verfügbar. Da die Impaktfaktoren etwas Vorlauf benötigen, nämlich genau den Zeitraum über den ein einzelner Wert ausgewertet wird, sind für die ersten Jahre auch nur sehr kleine Werte vorhanden. Verglichen mit den *[2008]*er Daten ist allerdings zu bemerken, dass zumindest für die älteren Jahrgänge 2002 bis 2004 die Reihenfolge und die ungefähren Abstände übereinstimmen, wenn die Abstände auch nicht proportional zu einander sind. Aber es ist erkennbar, dass *SIGMOD Record* und *TODS* relativ nah beieinander liegen, verglichen mit *VLDB*.

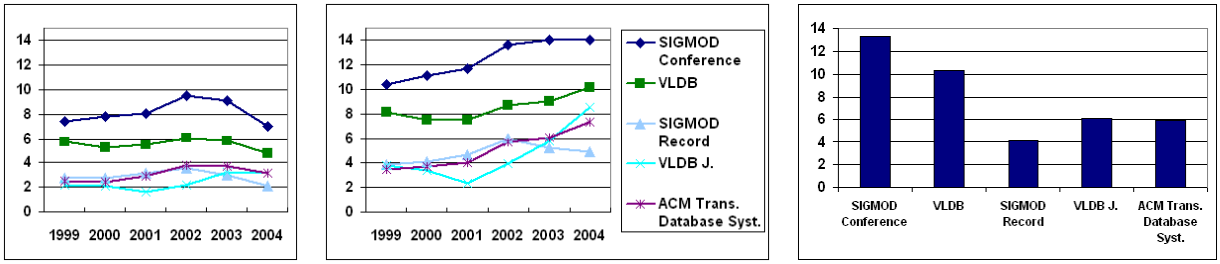
Die Abbildungen 7.11b und 7.11c fallen deutlich ähnlicher aus. In den *[2008]*er Daten fallen die absoluten Werte der Impaktfaktoren, genau wie die absoluten Zitierungszahlen, deutlich höher aus, als in den *[2005]*er Daten. Das beruht einerseits auf der Tatsache,



(a) JCR, Quelle: [oK06], SIGMOD R. verfügbar seit 2002, VLDB J. seit 2000

(b) IF über 2 Jahre, *[2005]*, Quelle: [RT05]

(c) IF über 2 Jahre, *[2008]*



(d) IF über 5 Jahre, *[2005]*, Quelle: [RT05]

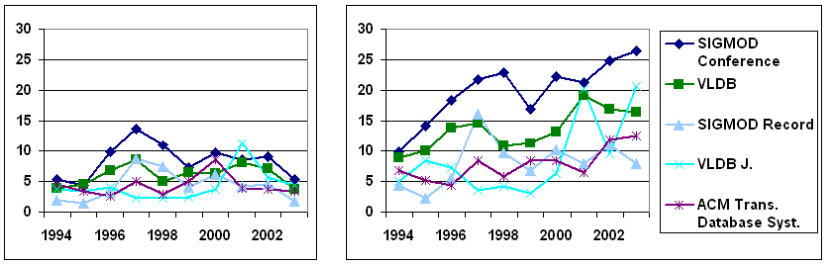
(e) IF über 5 Jahre, *[2008]*

(f) IF(2004) über 10 Jahre, *[2008]*

Abbildung 7.11: Impaktfaktoren für die betrachteten Publikationsorte

dass in *[2005]* nur mit *Google Scholar* ermittelte Zitierungen berücksichtigt wurden. Zum anderen wurde auch *Google Scholar* selbst ständig erweitert und durch andere zitierende Publikationen der betrachteten Jahre ergänzt. Zwischen 1994 und 2001 ähneln sich die Kurvenverläufe der beiden Diagramme sehr stark. Erst ab 2001 ergeben sich deutliche Unterschiede. Während in *[2005]* die Werte bis 2004 stetig absinken, steigen sie in *[2008]* weiter an. Doch auch hier sind die Abstände zwischen den verschiedenen Publikationsorten im Jahresvergleich jeweils proportional zueinander.

Deutlich heben sich die Konferenzen von den Journalen ab, wobei die *SIGMOD Conference* einen deutlich höheren Impaktfaktor hat als *VLDB*. Die Journale lassen sich aufgrund der stark schwankenden Kurven nur schlecht miteinander vergleichen. Daher ist eine Erweiterung des zur Berechnung des Impaktfaktors genutzten Zeitraums auf 5 Jahre von Vorteil. Dadurch werden auch Zitierungen von Papieren genutzt, die zum Zitierungszeitpunkt älter als 2 Jahre waren. Besonders gute Papiere erhalten Einfluss in die Berechnung von 5 Impaktfaktoren, werden dagegen aber stärker gemittelt. Dadurch werden die Kurven geglättet und besser vergleichbar. In dieser Ansicht liegen alle Journale relativ nah beieinander, obwohl der Impaktfaktor von *VLDB J.* zeitweilig nur bei der Hälfte im Vergleich zu den anderen Journale liegt. Während beim Zweijahreszeitraum noch spitzen zu verzeichnen waren, die sehr nah an die Konferenzen (speziell *VLDB*) her-



(a) *[2005]*, Quelle: [RT05] (b) *[2008]*

Abbildung 7.12: Entwicklickung der durchschnittlichen Zitierungszahlen pro Publikation für die betrachteten Publikationsorte. Einschränkung auf Zitierungen der beiden Folgejahre.

anreichten, liegen die Journale in dieser Ansicht deutlich unter den Konferenzen. Beim *VLDB Journal* ist seit 2001, sowohl in den *[2005]*er als auch in den *[2008]*er Daten, ein stärkerer Anstieg zu verzeichnen, als bei allen anderen Kurven und erreicht 2004 sogar den höchsten Wert aller Journale. Würde sich der Trend fortsetzen, könnte sich das *VLDB Journal* von den anderen absetzen.

In Abbildung 7.11f sind einmal die Impaktfaktoren für das Jahr 2004 und einem Zeitraum von 10 Jahren, also über alle betrachteten wissenschaftlichen Veröffentlichungen, nebeinander gestellt. Durch diesen recht langen Zeitraum wird besonders *SIGMOD Record* aufgrund des niedrigeren durchschnittlichen Zitierungsalters, benachteiligt und nimmt deutlich den letzten Platz ein. *VLDB Journal* und *TODS* liegen sehr eng beieinander, wobei *TODS* ganz knapp vorn liegt. Die Konferenzen heben sich wie in bei den Impaktfaktoren mit kürzerem Zeitraum, sehr deutlich ab, wobei *SIGMOD Conference* sich nocheinmal, aber nicht ganz so stark, von *VLDB* abhebt.

Die Impaktfaktoren der *[2008]*er Daten nehmen, im Gegensatz zu denen von *[2005]* bis in die jüngere Zeit immer weiter zu. Diese Beobachtung lässt sich nicht mit den absoluten Zitierungszahlen (vgl. Abschnitt 7.4) erklären, da diese sowohl insgesamt, als auch unterteilt nach Publikationsorten, in jüngerer Zeit abnehmen. Wie oben ausgeführt liegt dies vor allem daran, dass die älteren Papiere schon mehr Zeit hatten zitiert zu werden. Dies wurde auch in Abschnitt 7.5 belegt, da das Zitierungsalter stetig abnimmt, d.h. für die jüngeren Arbeiten die späten Zitierungen fehlen. Aus diesem Grund müssen, um die Impaktfaktoren zu erklären, nur die relevanten Zitierungen betrachtet werden. In Abbildung 7.12 sind die Enwicklung der Zitierungszahlen aus den *[2005]*er und *[2008]*er Daten dargestellt, wobei nur Zitierungen berücksichtigt wurden, die für einen Impaktfaktor über zwei Jahre eine Rolle spielen. Hier lässt sich deutlich erkennen, dass sich die Entwicklungen der Zitierungszahlen ab 1998 voneinander unterscheiden. Während sie in *[2005]* wieder absinken, steigen sie in *[2008]* weiter. Umso stärker die Unterschiede werden, umso stärker wirkt sich die unterschiedliche Entwicklung auf die Impaktfaktoren

	Autoren	#Zit.	#Pub.
1	Agrawal, Rakesh	10512	21
2	Srikant, Ramakrishnan	7346	7
3	Halevy, Alon	5000	25
4	Garcia, Hector	4455	47
5	Naughton, Jeffrey F.	4336	34
6	Franklin, Michael J.	3744	26
7	DeWitt, David J.	3791	27
8	Widom, Jennifer	3604	22
9	Faloutsos, Christos	3527	22
10	Han, Jiawei	3467	17

Tabelle 7.1: Top 10 der meist zitiertesten Autoren in den betrachteten Konferenzen und Journalen

aus, so dass auch sie in *[2005]* wieder abfallen, in *[2008]* aber weiter ansteigen.

7.8 Rankings

Da die Publikationen innerhalb einer Konferenz bzw. Journals sehr ungleich zitiert werden, siehe Abschnitt 7.6, lohnt sich eine Betrachtung aus Autorsicht oder ein Runterbrechen auf einzelne Publikationen. Aufgrund der geringen Anzahl an Publikationen pro Autor und Jahr in den untersuchten Konferenzen und Journalen, wäre eine Berechnung des Impaktfaktors für Autoren nicht sonderlich aussagekräftig, für Publikationen gar unmöglich. Daher bietet die Aufstellung von Ranglisten den passensten Vergleich.

Tabelle 7.1 gibt die zehn meist zitiertesten Autoren an. Dabei wurden die Publikationen, welche zwischen 1994 und 2003 in den betrachteten Publikationsorten veröffentlicht wurden und alle bis heute erhaltenen Zitierungen berücksichtigt. Die beiden besten Autoren Agrawal und Srikant erreichten mit Abstand die höchsten Zitierungszahlen. Interessanterweise, sind alle sieben Srikant-Papiere zusammen mit Agrawal geschrieben wurden. Die übrigen 14 Papiere von Agrawal erhalten mit etwa dreitausend im Schnitt deutlich weniger Zitierungen. Der weitere Verlauf der Zitierungszahlen im Ranking verläuft deutlich flacher. Während Platz 6 die 4000er Marke unterschreiten, wird die 3000er Marke vom 14., die 2000er Marke vom 20. und die 1000er Marke gar erst vom 82. Platz unterboten. Auch unter den Autoren ist demnach eine starke Ungleichverteilung der erhaltenen Zitierungen zu erkennen.

Auch in der Anzahl der veröffentlichten Papiere und damit auch der durchschnittlich erhaltenen Zitierungen ergeben sich große Unterschiede. So veröffentlichte der viertplatzierte Garcia mit 47 Papieren fast sieben mal so viele, wie der zweit platzierte Srikant und erhält so mit einem eher mittelmäßigen Zitierungsdurchschnitt von weniger als 100, eine doch sehr gute Platzierung.

	Publikation	Konferenz	Jahr	Autoren	#Zit.
1	Fast Algorithms for Mining Association Rules in Large Databases.	VLDB	1994	Agrawal, Rakesh; Srikant, Ramakrishnan	4975
2	BIRCH: An Efficient Data Clustering Method for Very Large Databases.	SIGMOD Conference	1996	Chang, Tian-Ping; Ramakrishnan, Raghu; Livny, Miron	1237
3	Efficient and Effective Clustering Methods for Spatial Data Mining.	VLDB	1994	Ng, Raymond; Han, Jiawei	1052
4	Mining Quantitative Association Rules in Large Relational Tables.	SIGMOD Conference	1996	Srikant, Ramakrishnan; Agrawal, Rakesh	1016
5	Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications.	SIGMOD Conference	1998	Agrawal, Rakesh; Gehrke, E.; Gunopulos, Dimitrios; Raghavan, Prabhakar	1003
6	An Efficient Algorithm for Mining Association Rules in Large Databases.	VLDB	1995	Savasere, Ashok; Omiecinski, Edward; Navathe, Shamkant B.	1002
7	Querying Heterogeneous Information Sources Using Source Descriptions	VLDB	1996	Halevy, Alon; Rajaraman, Anand; Ordille, Joann J.	994
8	An Effective Hash Based Algorithm for Mining Association Rules.	SIGMOD Conference	1995	Park, Jeong Doo; Chen, Ming-Syan; Yu, Philip	990
9	Implementing Data Cubes Efficiently.	SIGMOD Conference	1996	Harinarayan, Venky; Rajaraman, Anand; Ullman, Jeffrey D.	987
5	DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases.	VLDB	1997	Feldman, Roy; Widom, Jennifer	900

Tabelle 7.2: Top 10 der meist zitiertesten Konferenzpapiere, erschienen zwischen 1994 und 2003

Wenig überraschend finden sich die Arbeiten der Topautoren auch in den Top-Konferenzpublikationen in Tabelle 7.2 wieder. In sechs der zehn Papiere ist mindestens einer der Autoren aus Tabelle 7.1 beteiligt. Die gemeinsamen Papiere der beiden Topautoren Agrawal und Srikant nehmen sogar die Positionen eins und vier ein und sind thematisch sehr nah verwandt. Die beiden Konferenzen sind in der Liste recht gleichmäßig vertreten, was von daher überrascht, da *VLDB* eine wesentlich geringere durchschnittliche Zitierungszahl verglichen mit *SIGMOD Conference* besitzt.

Im Ranking der Journalpublikationen in Tabelle 7.3 ergibt sich ein ähnliches Bild. Auch hier sind *SIGMOD Record* und *VLDB Journal* gleichmäßig vertreten. Einzig aus *TODS* befindet sich die beste Publikation erst auf dem 15. Platz. Auch dies ist aufgrund der teilweise deutlich höheren Zitierungsdurchschnitten verwunderlich, ist aber durch einen niedrigeren Citation Skew erklärbar, da dadurch die Zitierungen gleichmäßiger verteilt sind und sich die Toppapiere weniger stark vom Durchschnitt abheben.

Beim Untersuchen der Vorkommen der Erscheinungsjahre in den vorderen Bereichen der Rangliste, fällt auf, dass die zehn besten Konferenzveröffentlichungen allesamt aus den ersten drei Jahren des untersuchten Zeitraums stammen. Im Journalpublikationsranking ist dieses Verhalten nicht zu beobachten. Schuld daran ist das in der Regel höhere Zitierungsalter innerhalb der Konferenzen. Jüngere Papiere sind dadurch benachteiligt. Allein das beste Agrawal-Papier besitzt ein durchschnittliches Zitierungsalter von 9 Jahren, die Hälfte aller Zitierungen stammt aus den ersten zehn Jahren nach der Veröffentlichung. Jüngere Papiere, z.B. von 1999, hätten bei gleichem Zitierungsverhalten demnach gar

	Paper	Journal	Jahr	Autoren	#Zit.
1	An Overview of Data Warehousing and OLAP Technology.	SIGMOD Record	1997	Chaudhuri, Indrajit; Dayal, Umeshwar	1118
2	A survey of approaches to automatic schema matching.	VLDB J.	2001	Rahm, Erhard; Bernstein, Philip A.	1003
3	Lore: A Database Management System for Semistructured Data.	SIGMOD Record	1997	McHugh, Jason; Abiteboul, Serge; Feldman, Roy; Quass, Dallan; Widom, Jennifer	713
4	Database Techniques for the World-Wide Web: A Survey.	SIGMOD Record	1998	Florescu, Daniela; Halevy, Alon; Mendelzon, Alberto O.	588
5	Answering queries using views: A survey.	VLDB J.	2001	Halevy, Alon	557
6	An Introduction to Spatial Database Systems	VLDB J.	1994	Güting, Ralf Hartmut	447
7	Sleepers and Workaholics: Caching Strategies in Mobile Environments	VLDB J.	1995	Barbará, Daniel; Celinski, Tomasz	411
8	The TV-Tree: An Index Structure for High-Dimensional Data	VLDB J.	1994	Lin, King-Ip; Jagadish, H. V.; Faloutsos, Christos	408
9	The Cougar Approach to In-Network Query Processing in Sensor Networks.	SIGMOD Record	2002	Yao, Yong; Gehrke, E.	358
10	Wrapper Generation for Semi-structured Internet Sources.	SIGMOD Record	1997	Ashish, Naveen; Knoblock, Craig A.	298

Tabelle 7.3: Top 10 der meist zitiertesten Journalpapiere, erschienen zwischen 1994 und 2003

keine Chance besser abzuschneiden.

Um die Konferenzpapiere fairer vergleichen zu können, ist daher eine Begrenzung des maximalen Zitierungsalters notwendig. Eine Begrenzung auf vier Jahre ist ausreichend, da dann 2007 als letztes Zitierungsjahr für Publikationen aus 2003 benötigt wird. Tabelle 7.4 stellt das Ergebnis dar. In dieser Darstellung liegen die Publikationen deutlich näher beieinander. Das Agrawal-Papier wurde von der Spitze abgelöst und liegt immerhin noch auf Platz drei. Immer noch sind die Top10-Autoren an vier der zehn Papiere beteiligt gewesen.

Diese Methode der Rangfolgenbildung hat allerdings den Nachteil, dass schneller zitierte Publikationen bevorzugt werden. Sind Arbeiten dagegen ihrer Zeit etwas voraus, werden sie später zitiert und erhalten eine schlechtere Platzierung. Das gleiche gilt für Klassiker, die ihre hohen Zitierungszahlen durch langjährige stetige Zitierungen erhalten. Ein wirklich verlässliches Ranking, kann daher erst nach einem längeren Zeitraum, in diesem Fall ca. zehn Jahre, nach Veröffentlichung der letzten betrachteten Publikation aufgestellt werden.

7.9 PageRank

Obwohl die Nutzung des Impaktfaktors zur Einschätzung des wissenschaftlichen Einflusses von Konferenzen und Journalen weit verbreitet ist, besitzt er Eigenschaften die

	Paper	Konferenz	Jahr	Autoren	#Zit.
1	Relational Databases for Querying XML Documents: Limitations and Opportunities.	VLDB	1999	Shanmugasundaram, Jayavel; Tufte, Kristin; Zhang, Chun-Ting; He, Taigang; DeWitt, David J.; Naughton, Jeffrey F.	380
2	The Design of an Acquisitional Query Processor For Sensor Networks.	SIGMOD Conference	2003	Madden, Samuel; Franklin, Michael J.; Hellerstein, Joseph L.; Hong, Wei	347
3	Fast Algorithms for Mining Association Rules in Large Databases.	VLDB	1994	Agrawal, Rakesh; Srikant, Ramakrishnan	339
4	Indexing and Querying XML Data for Regular Path Expressions.	VLDB	2001	Li, Yuanzhen; Moon, Bongkyo	332
5	Generic Schema Matching with Cupid.	VLDB	2001	Madhavan, Jayant; Bernstein, Philip A.; Rahm, Erhard	311
6	Implementing Data Cubes Efficiently.	SIGMOD Conference	1996	Harinarayan, Venky; Rajaraman, Anand; Ullman, Jeffrey D.	290
7	Monitoring Streams - A New Class of Data Management Applications.	VLDB	2002	Carney, Don; Çetintemel, Ugur; Cherniack, Mitch; Convey, Christian; Lee, Jung-Do; Seidman, Greg; Stonebraker, Michael; Tatbul, Nesime; Zdonik, Stanley B.	280
8	Querying Heterogeneous Information Sources Using Source Descriptions	VLDB	1996	Halevy, Alon; Rajaraman, Anand; Ordille, Joann J.	279
9	A Query Language and Optimization Techniques for Unstructured Data.	SIGMOD Conference	1996	Buneman, Peter; Davidson, Susan B.; Hillebrand, Gerd G.; Suciu, Dan	273
10	NiagaraCQ: A Scalable Continuous Query System for Internet Databases.	SIGMOD Conference	2000	Chen, Jianjun; DeWitt, David J.; Tian, Jun-Feng; Wang, Shie-Yuan	271

Tabelle 7.4: Top 10 der meist zitiertesten Konferenzpapiere, erschienen zwischen 1994 und 2003. Nur Zitierungen die max. 4 Jahre nach Veröffentlichung erfolgt sind, wurden berücksichtigt.

sich nachteilig auswirken können. Zum einen werden nur direkte Zitierungen von Publikationen betrachtet. Aber gerade einflussreiche Ideen werden schnell aufgegriffen und weiterverwendet. Wie in [MM96] argumentiert wird, werden Sekundärquellen, also Arbeiten in denen die ersten Ideen schon verbessert oder nachgenutzt wurden, in der Regel gegenüber den Originalquellen bevorzugt.

Zum anderen wurden Selbstzitierungen bei der Berechung der Impaktfaktoren komplett ignoriert. Was in diesem Fall auch korrekt ist, da der Einfluss durch übermäßige Selbstzitierungen manipuliert werden könnte. Allerdings stellen Selbstzitierungen auch eine Informationsquelle dar, die so nicht genutzt wird.

Als letzter Kritikpunkt gilt, dass alle zitierenden Papiere den gleichen Beitrag zum Impaktfaktor der zitierten Publikation erhalten. Eine Gewichtung wird dabei nicht vorgenommen. D.h. werden zwei Papiere gleich oft zitiert, das eine aber vorwiegend in Demonstrationen, das andere mehr in anderen wissenschaftlichen Publikationen, erhalten beide Papiere trotzdem den gleichen Wert.

Eine Alternative zum Impaktfaktor, die zum Beispiel in [JBdS07] vorgeschlagen wird, ist *PageRank*. PageRank wurde von der Stanford Universität zur Gewichtung von Webseiten entwickelt und befindet sich unter anderen bei der Suchmaschine Google im Einsatz.

Der PageRank einer Webseite A gibt an, mit welcher Wahrscheinlichkeit ein Nutzer, der sich gerade auf einer anderen zufälligen Webseite befindet, als nächstes auf die Seite A wechselt. Übertragen auf die Zitierungsanalyse hieße das, dass ein Leser einer zufälligen Publikation mit der Wahrscheinlichkeit PR(A), als nächstes Publikation A liest, wobei PR(A) der PageRank von A ist und Zitierungen äquivalent zu Webverlinkungen betrachtet werden. Dem PageRank liegt dazu das *Random Surfer* Konzept zu Grunde, das heißt der Nutzer verfolgt nicht zwangläufig die Links auf einer Webseite, sondern wechselt mit einer gewissen Wahrscheinlichkeit auf eine andere zufällig gewählte Seite.

PageRank hat zur Relevanzeinschätzung von Webseiten, aufgrund der realistischen Simulation des Nutzerverhaltens, einen großen Erfolg. Der Zitierungsgraph ähnelt der Infrastruktur von Webseiten recht stark. Auch er ist gerichtet und einzelne Knoten referenzieren nur eine eng begrenzte Anzahl anderer Knoten. Die Anzahl eingehender Kanten ist dagegen unbegrenzt.

Der PageRank-Ansatz bietet für die Zitierungsanalyse den Vorteil, dass Zitierungen rekursiv berücksichtigt werden. Durch den Dämpfungsfaktor, der Random Surfer, wird der rekursive Einfluss aber begrenzt. Auf diese Weise tragen Zitierungen einflussreicher Publikationen diesen Einfluss stärker auf die zitierten, als andere. Durch die Berücksichtigung von Selbstzitierungen können auch Basisveröffentlichungen eines Autors, welche seltener zitiert werden, an Gewicht gewinnen.

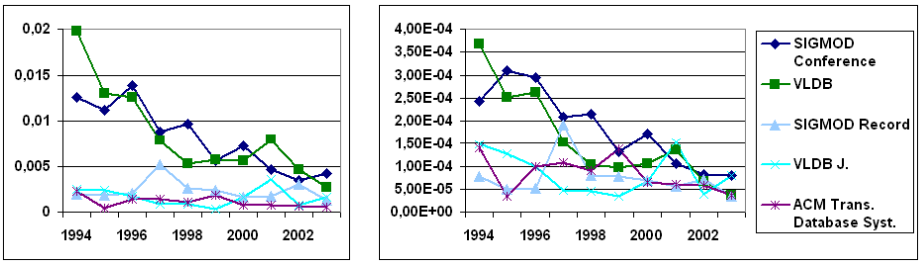
Die Berechnung des PageRanks erfolgt iterativ. Sei N die Anzahl von Publikationen, $C(P_i)$ die Anzahl der von Publikation P_i zitierten Publikationen, V_i die Menge der Publikationen die P_i zitieren und d der Dämpfungsfaktor mit $0 < d < 1$. Die PageRanks $PR(P_i)$ werden jeweils mit $1/N$ initialisiert. In jeder Iteration werden alle PageRank-Werte neu berechnet:

$$PR(P_i) = \frac{1-d}{N} + d \sum_{p \in V_i} \frac{PR(p)}{C(p)}$$

Typischerweise wird $d = 0,85$ gesetzt. Zusätzlich werden die PageRanks aller Senken, also Publikationen mit $C(P_i) = 0$, gleichmäßig über alle anderen aufgeteilt. Die Iteration konvergiert in der Regel sehr schnell, so dass nur wenige Iterationen nötig werden.

Da es sich beim PageRank um eine Wahrscheinlichkeit handelt, können die Werte mehrerer Publikationen problemlos aufaddiert werden. Damit sind wieder Auswertungen nach Konferenzen, Journalen, Autoren, Ländern und anderen Gruppierungen möglich. Die Summe der PageRanks aller Publikationen eines Autors gibt beispielsweise die Wahrscheinlichkeit an, mit der ein Nutzer eine Publikation dieses Autors auswählt. Damit lässt sich der PageRank ähnlich zu den Impaktfaktoren verwenden.

Im Gegensatz zur Berechnung der Impaktfaktoren, werden für den PageRank einige zusätzliche Informationen benötigt. Während es für die Impaktfaktoren ausreichte, die zi-



(a) summierter PageRank (b) PageRank-Durchschnitt

Abbildung 7.13: Entwicklung des PageRanks der verschiedenen Konferenzen und Journale.

tierenden Publikationen zu kennen, werden nun zusätzlich einerseits die zitierenden der zitierenden Publikationen und deren zitierenden usw. benötigt. Andererseits muss für jede zitierende Publikation bekannt sein, wieviele Zitierungen sie insgesamt enthält. Statt eines flachen Waldes von Publikationen, ausgehend von den zu analysierenden Publikationen als Wurzeln, wird nun ein ganzes Netz um diese benötigt. In Kapitel 4 wurde auf die Beschaffung der Zitierungsinformationen aus `Google Scholar` eingegangen. Die Rekursion wurde in der zweiten Stufe, also auf der Ebene der Publikationen die die relevanten Publikationen zitieren, abgebrochen. Diese Rekursion müsste entsprechend noch einige Ebenen fortgesetzt werden. Durch den Dämpfungsfaktor wären nur wenige weitere Rekursionen nötig, trotzdem steigt der Datenumfang und die Anzahl der Datenabfragen exponentiell. Die Anzahl der in einer Publikationen enthaltenen Zitierungen bekommt man aus `Google Scholar` gar nicht auf direktem Wege heraus. Hierfür müsste auf Quellen wie `ACM` oder `Citeseer` zurückgegriffen werden, welche ihrerseits recht unvollständig sind.

PageRank Rankings

Ungeachtet dieser fehlenden Informationen sind in Abbildung 7.13 die jahresweisen Verläufe der summierten bzw. Durchschnitt- PageRanks der verschiedenen Publikationen dargestellt. Aufgrund der fehlenden Informationen sind die Ergebnisse natürlich nicht repräsentativ und dienen nur der Demonstration der Machbarkeit. Die Verläufe ähneln sehr stark denen der Zitierungszahlen, was aufgrund der Berechungsweise auch nachvollziehbar ist. Einzig zum Ende hin, nehmen die PageRanks stärker ab, da die zitierenden Papiere noch weniger Zeit hatten selbst zitiert zu werden und damit ein geringeres Gewicht erhalten. Bei Hinzugabe der fehlenden Daten, wäre dieses Verhalten wahrscheinlich noch stärker zu beobachten.

Das Autorenranking wird durch die Nutzung des PageRanks stärker beeinflusst, siehe

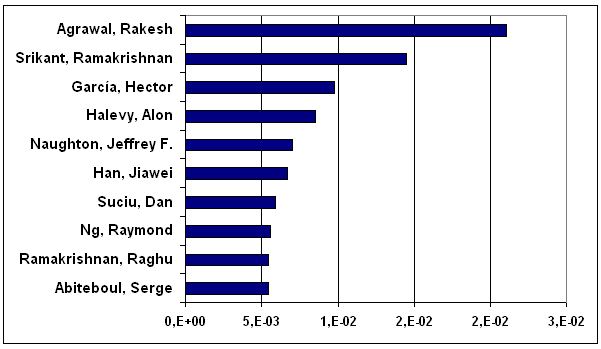


Abbildung 7.14: PageRanks der Top 10 Autoren

Abbildung 7.14 und Tabelle 7.1. Die beiden besten Autoren sind zwar unverändert, dagegen verbessert sich Garcia vom vierten auf den dritten Platz. Abiteboul, Ng, Sucio und Ramakrishnan erscheinen neu im Ranking.

Auch das Mithilfe der PageRanks aufgestellten Top10-Rankings von Konferenzpublikationen in Tabelle 7.5 unterscheidet sich im Detail von Tabelle 7.2. Das Toppapier von Agrawal und Srikant ist zwar auch hier auf Platz eins, dafür kommt es darunter zu stärkeren Verschiebungen. Es sind hier nun sechs Publikationen von den PageRank-Top10-Autoren vertreten. Das Gesamtbild ändert sich allerdings nicht, im Vergleich der Nutzung der Zitierungszahlen. Beide betrachteten Konferenzen sind noch relativ gleichmäßig vertreten.

PageRank Modifikationen

In [JBdS07] wird statt der Publikationszitierungsgraphs der Journalzitierungsgraph betrachtet. Das reduziert die zu verarbeitende Datenmenge signifikant. Allerdings ist es so notwendig, die Kanten des Graphs mit Hilfe der Zitierungszahlen zu gewichten. Diese Berechnung liefert ein anderes Ergebnis, als die oben genannte Definition. Hier wird der Einfluss eines Journals auf alle zitierten Journale verteilt. Das heißt auch, dass relativ unwichtige Publikationen innerhalb eines Journals das gleiche Gewicht beim Übertragen von Einfluss erhalten.

Weiter wird dort vorgeschlagen, das Produkt des Impaktfaktors und des PageRanks zu bilden. Schließlich repräsentiert der Impaktfaktor ein direktes Maß des wissenschaftlichen Einflusses anhand der direkten Zitierungen. Der PageRank dagegen simuliert die Propagierung von Prestige von Journalen auf andere Journale und ist damit ein intuitiveres Maß. Die Kombination beider verbindet die Vorteile.

	Paper	Konferenz	Jahr	Autoren	PageRank
1	Fast Algorithms for Mining Association Rules in Large Databases.	VLDB	1994	Agrawal, Rakesh; Srikant, Ramakrishnan	1,0E-02
2	Efficient and Effective Clustering Methods for Spatial Data Mining.	VLDB	1994	Ng, Raymond; Han, Jiawei	3,4E-03
3	Querying Heterogeneous Information Sources Using Source Descriptions	VLDB	1996	Halevy, Alon; Rajaraman, Anand; Ordille, Joann J.	2,5E-03
4	An Effective Hash Based Algorithm for Mining Association Rules.	SIGMOD Conference	1995	Park, Jeong Doo; Chen, Ming-Syan; Yu, Philip	2,5E-03
5	BIRCH: An Efficient Data Clustering Method for Very Large Databases.	SIGMOD Conference	1996	Chang, Tian-Ping; Ramakrishnan, Raghu; Livny, Miron	2,4E-03
6	Improving Business Process Quality through Exception Understanding, Prediction, and Prevention.	VLDB	2001	Grigori, Daniela; Casati, Fabio; Dayal, Umeshwar; Shan, Ming-Chien	2,2E-03
7	Implementing Data Cubes Efficiently.	SIGMOD Conference	1996	Harinarayan, Venky; Rajaraman, Anand; Ullman, Jeffrey D.	2,2E-03
8	A Query Language and Optimization Techniques for Unstructured Data.	SIGMOD Conference	1996	Buneman, Peter; Davidson, Susan B.; Hillebrand, Gerd G.; Suciu, Dan	2,1E-03
9	Discovery of Multiple-Level Association Rules from Large Databases.	VLDB	1995	Han, Jiawei; Fu, Yongjian	1,8E-03
10	Energy Efficient Indexing on Air.	SIGMOD Conference	1994	Celinski, Tomasz; Viswanathan, S.; Badrinath, R.	1,8E-03

Tabelle 7.5: Top 10 PageRanks von Konferenzpapieren, erschienen zwischen 1994 und 2003.

Als Ergebnis erhalten die Autoren in [JBdS07], dass PageRank und Impaktfaktor besonders in der Informatik, gegenüber anderer Fachrichtungen, stärker auseinander laufen. Für eine mögliche Erklärung werden Journale in prestigeträchtige Journale und populäre Journale unterteilt. Prestigeträchtige Journale zeichnen sich durch weniger Zitierungen aus, aber diese erhalten sie von anderen prestigeträchtigen Journalen. Populäre Journale dagegen erhalten viele Zitierungen und dadurch einen hohen Impaktfaktor, die Zitierungen erhalten sie aber aus weniger prestigeträchtigen Journalen, wodurch der PageRank vergleichsweise klein wird. Übertragen auf Publikationen lässt sich damit vermuten, dass die Publikationen bzw. Autoren der PageRank-Rankings, die nicht im Zitierungszahlenranking vorkommen, prestigeträchtiger sind, die anderen dagegen populärer.

8 Zusammenfassung

Zitierungsanalysen sind weit verbreitet und dienen zur Einschätzung und des Vergleichs der Bedeutung von wissenschaftlichen Publikationen, Publikationsorten, Autoren, Instituten und ähnliches. Dabei wurden in bisherigen Analysen Konferenzen meist unbeachtet gelassen und Vergleiche nur zwischen Journalen durchgeführt. Aber offensichtlich besitzen in der Informatik gerade Konferenzen einen weit größeren Einfluss und werden dem entsprechend öfter zitiert als vergleichbare Journale.

Als Maß der Popularität werden meist Kennzahlen wie die absoluten oder durchschnittlichen Zitierungszahlen oder der Impaktfaktor herangezogen. Es zeigt sich allerdings, dass die absoluten Werte dieser Maße zwischen verschiedenen Analysen keineswegs vergleichbar sind und sehr stark vom untersuchten Datenbestand abhängen. Die relativen Abstände zwischen verschiedenen untersuchten Objekten sind dagegen in verschiedenen Analysen recht ähnlich, so dass zumindest Kurvenverläufe an sich vergleichbar sind. Für einen verlässlichen Vergleich zwischen Objekten, sollten die Kennzahlen dagegen auf der gleichen zugrunde liegenden Datenmenge bestimmt werden.

Obwohl einige frei verfügbare Bibliographiedatenbanken existieren, enthält keine davon alle zur Analyse benötigten Informationen. Daher ist die Integration mehrerer Datenquellen notwendig. Problematisch ist dabei vor allem die Datenextraktion aus Webdatenquellen, welche zum einen einige redundante Daten erzeugt, es zum anderen nur erlaubt einen eng begrenzten Datenausschnitt zu extrahieren. Zusätzlich enthalten die wenigsten Datenquellen eindeutige globale Schlüssel wie ISBN oder DOI, so dass eine massive Datenbereinigung, dabei vor allem die Beseitigung von Duplikaten, notwendig wird, um eine hohe Qualität der Ergebnisse zu erhalten.

Einige Informationen, wie z.B. eine Kategorisierung von Publikationen, sind in keiner verfügbaren Datenquelle direkt verfügbar und müssen manuell ergänzt werden. Dieser Prozess ist schwierig und zeitintensiv und macht es daher unmöglich große Datenmengen zu ergänzen. Allerdings erhalten gerade als wissenschaftlich eingestufte Publikationen die meisten Zitierungen und sind in Konferenzen mit einem deutlich kleineren Prozentsatz vertreten als in Journalen. Zum wirksamen Vergleich von Konferenzen und Journalen ist eine Kategorisierung ihrer Publikationen daher zwingend erforderlich.

Im Laufe der Arbeit wurde ein Data Warehouse erstellt, welches alle zur Analyse notwendigen Daten in sich vereinigt. Das Data Warehouse bietet die Möglichkeit verschiedenste

Analysen immer auf dem gleichen Datenbestand wiederholt und effizient auszuführen. Eine regelmäßige Aktualisierung des Datenbestands wird unterstützt, so dass auch künftige Analysen mit jeweils aktuellen Daten durchführbar sind.

Beim Entwurf wurde auf einen modularen Aufbau des Erstellungsprozesses geachtet, um eine einfache und flexible Erweiterung zu gewährleisten. Diese Erweiterungen umfassen speziell die einfache Einbindung und Nutzung weiterer Datenquellen und Änderung und Ergänzung vorhandener qualitätssichernden Mechanismen. Weiter wurde auf Besonderheiten und Problematiken bei der Nutzung von webbasierten Datenquellen, wie Google Scholar und ACM hinsichtlich des Zugriffs, der Datenextraktion, aber auch der Datenqualität näher eingegangen. Im Bereich der Datenbereinigung wurden zum einen mögliche Umsetzungen für die Entfernung von Duplikaten besprochen, da diese für die Auswertung eine große Wichtigkeit besitzt, um keine Daten doppelt zu zählen. Zum anderen wurden Möglichkeiten erläutert unsaubere Titelangaben in DBLP zu bereinigen, um die Webdatenabfrage und die spätere Konsolidierung nicht negativ zu beeinflussen. Letztlich wurde eine Variante dargestellt, erkannte Duplikate durch zielgerichtete Zusammenführung so zu eliminieren, dass ein Zustand mit größtmöglicher Qualität bei Erhaltung des größtmöglichsten Informationsgehalts erreicht wird.

Obwohl hier nur zwei Konferenzen und drei Journale im Bereich der Informatik, speziell Datenbanken, berücksichtigt wurden, lässt sich das entstandene Data Warehouse auch für andere Fachbereiche nutzen.

Ausblick

Die betrachteten Konferenzen und Journale stammen aus dem Datenbankbereich. Gemachte Vermutungen wie dass Konferenzen deutlich stärker zitiert werden, sind damit nicht repräsentativ. Eine Ausdehnung auf weitere Konferenzen und Journale im Fachbereich Informatik wäre dafür notwendig. Dafür müsste die bisherige manuelle Ergänzung der Daten automatisiert werden, etwa durch die Einbindung weiterer Datenquellen oder durch die Verwendung von Heuristiken.

Weiter würde die Erweiterung des Analysedatenraums den Webdatenextraktionsaufwand proportional erhöhen. Auch hier könnten Heuristiken oder andere Suchstrategien helfen den Aufwand zu begrenzen, indem nicht alle, sondern nur oft zitierte Publikationen berücksichtigt werden. Allerdings sollte die Anzahl über 25% liegen, da diese einen Anteil von 70% an den erhaltenen Zitierungen ausmachen. Nur so ist eine repräsentative Analyse gewährleistet.

Die Datenqualität kann durch die Verwendung besserer Datenbereinigungsmechanismen noch verbessert werden. Das in dieser Arbeit verwendete rein auf Attributähnlichkeiten

basierte Verfahren kann Aufgrund seiner Beschaffenheit nie alle Duplikate finden, dagegen andere ähnlich klingende Objekte schlecht auseinander halten. Durch den modularen Aufbau des ETL-Prozesses ist der Austausch des Verfahren jedoch kein Problem und beeinflusst den restlichen Prozess nicht sichtbar.

Durch die Hinzunahme weiterer Bibliographiedatenbanken, wie Scopus oder MS Libra, können fehlende Daten weiter ergänzt werden und so Unregelmäßigkeiten in einzelnen Quellen ausgleichen. Die Nutzung möglichst vieler unabhängiger Datenbanken als Datenquelle gewährleistet auch die Unabhängigkeit der Analyseergebnisse von den einzelnen Quellen.

Anhang

A Verwendete Software und Frameworks

Microsoft SQL Server Enterprise Edition[1]

SQL Datenbank Server

Speicherung und Verwaltung der Basisdatenbank und Data Warehouse Datenbank. Der Datenbank Server ermöglicht die einfache und effiziente Verwaltung großer Datenmengen. Allein DBLP enthält fast eine Million Publikationen und etwa 500.000 verschiedene Autoren. Während der Arbeit des ETL-Prozesses sind annähernd drei Millionen Publikationen in der Datenbank gespeichert, auf welche während der Deduplizierung exzessiv lesend und schreibend zugegriffen wird.

SQL Server Integration Services (SSIS)

Die SSIS sind eine Sammlung von Komponenten, die aneinander gekoppelt beliebige ETL-Aufgaben, wie das Extrahieren und Transformieren von Daten aus Datenquellen, Verarbeitung von Daten in der Datenbank und vieles mehr. In dieser Arbeit wurden die SSIS dazu verwendet die beschriebenen Datenquellen, bzw. im Falle der Webdaten das Ergebnis der Webdatenextraktion, auszulesen und in die Basisdatenbank zu importieren. Das Aufbereiten der Daten zum Befüllen der Data Warehousedatenbank wurde ebenfalls mit Hilfe der SSIS realisiert. Die Komponentensammlung kann aufgrund des frei nutzbaren Entwicklungsframeworks beliebig erweitert werden, um den eigenen Anforderungen zu entsprechen.

SQL Server Analysis Services (SSAS)

Die SSAS sind ein OLAP-Tool. Aufgrund der multidimensionalen Speicherung der Daten können Abfragen einfach erstellt werden. Da Teilergebnisse voraggregiert gespeichert sind,

[1] Die Enterprise Edition ist hinsichtlich des Funktionsumfangs äquivalent zur Developer Edition. Einziger Unterschied liegt in der Lizenzierung, da die Developer Edition nur für Entwicklungszwecke genutzt werden darf.

ist die Ausführung von aggregierenden Abfragen besonders effizient. Neben Standardabfragen, welche mittels des integrierten Querydesigners intuitiv formuliert werden, sind auch komplexere Abfragen mittels der Abfragesprache MDX (z.B. [WEP08]) möglich.

SQL Server Reporting Services (SSRS)

Mittels der Reporting Services ist das Entwerfen von Berichten mit Ausgabe von Tabellen und Diagrammen sehr einfach. Die Berichte können jederzeit, auch regelmäßig automatisiert, erneut ausgeführt werden, wodurch den Berichten immer die aktuellsten Zahlen zu Grunde liegen. Da als Datenquelle auch eine SSAS Datenbank genutzt werden kann, sind alle multidimensionalen Anfragen, inklusive MDX-Anfragen, als Quelle der Anzeige möglich.

SQL Server Agent

Der SQL Server Agent kann verschiedene Aufgaben (Jobs) verwalten und erlaubt deren manuelle oder zeitlich geplante Ausführung. Ein Job kann aus verschiedenen Teilschritten bestehen, so dass u.a. verschiedene SSIS Pakete sequentiell abgearbeitet werden können. Der SQL Server Agent ist für die Aufgabe nicht zwingend erforderlich, er erleichtert aber die Verwaltung der benötigten Teilschritte.

.Net 2.0 Framework

Das .Net 2.0 Framework wurde zur Entwicklung einiger Teilkomponenten genutzt. Zum einen baut das Tool zur Webdatenextraktion darauf auf, zum andern stellt SSIS diverse .Net 2.0 Klassen zur Verfügung, um weitere SSIS Komponenten zu entwickeln und einzubinden. Das .Net 2.0 Framework wird auf den Rechnern benötigt, auf den die betroffenen Komponenten ausgeführt werden.

XSLT (mit VBScript)

Zur Verarbeitung von XML-Dateien wurde der in .Net 2.0 integrierte XML-Prozessor verwendet. Dieser bietet die Möglichkeit XML-Dateien nahezu beliebig komplex zu transformieren und wieder als XML-Dateien auszugeben. Reicht das normale XSL-Sprachportfolio nicht aus, können VB-Skripte angelegt und innerhalb der Transformation aufgerufen werden. Damit sind auch Auswertung von regulären Ausdrücken und ähnliches möglich.

B Überblick über die im Rahmen der Arbeit geschaffenen Komponenten

Zusätzlich zu den erstellten relationalen Datenbankschemas für die Basisdatenbank und Data Warehouse und der multidimensionalen Analysis Services Datenbank, war es notwendig einige weitere Komponenten zu schaffen, um den ETL-Prozess bis hin zur grafischen Darstellung der Ergebnisse, vollständig implementieren zu können. Die einzelnen Komponenten werden im Folgenden jeweils kurz beschrieben.

B.1 SSIS Komponenten (SSISUtil)

Die folgenden Komponenten wurden als Ersatz fehlender äquivalenter Komponenten in SSIS entwickelt. Sie können in die Toolbar des SSIS-Desginers eingebunden und dort normal wie SSIS-eigene Komponenten genutzt werden. Zur Nutzung müssen alle Assemblydateien in den GAC[1] installiert und zusätzlich die SSISUtil.dll nach PipelineComponents und SSISTasks.dll nach Tasks im Installationsverzeichnis des SQL Server kopiert werden.

XmlExtSource

Die in SSIS enthaltene XML-Eingabequelle arbeitet nur sehr beschränkt. Zum einen akzeptiert sie keine XML-Dateien mit zugehöriger DTD, zum anderen müssen alle Xmlknoten, die einen Datensatz darstellen, gleich aufgebaut und gleich benamt sein. Beides wurde schon durch die DBLP-Xmldatei verletzt. Daher war es nötig eine Komponente zu entwickeln die XML-Datei flexibler Importieren kann. Mit *XmlExtSource* können hierarchisch gegliederte Strukturen verarbeitet werden. Durch die Vergabe von temporären IDs und die Möglichkeit von Datensätzen auf Daten einer höheren Hierarchie zuzugreifen, können Unterknoten später ihren Eltern zugeordnet werden. Es ist die Ausgabe von beliebig vielen Datenströmen, mit unterschiedlichen Datensatztypen mit einer Komponente möglich, wodurch eine XML-Datei nur einmal eingelesen werden muss.

[1] Global Assembly Cache

XmlDestination

SSIS verfügt über keine Möglichkeit der Ausgabe von XML-Dateien. *XmlDestination* ermöglicht die Ausgabe von hierarchischen Daten durch die Eingabe von verschiedenen Datenströmen und die Möglichkeit eine Join-Bedingung anzugeben. Außerdem können die Xml-Dateien gesplittet werden, um die Datenmenge pro Datei moderat zu lassen. Zusätzlich ist die Transformierung der ausgegeben Dateien anschließend möglich, so dass ein beliebiges Ausgabeformat realisiert werden kann.

MergeCluster

MergeCluster übernimmt die relationale Vereinigung von geclusterten Tabellen inklusive ihrer über Fremdschlüsselbeziehungen abhängigen Tabellen. Der Algorithmus entspricht dem in Abschnitt 5.3 beschriebenen. Die Komponente benötigt den Namen der geclusterten Tabelle und ihrer Clustertabelle. Weiter kann der Vereinigungsvorgang für jede Tabelle beeinflusst werden. Für jedes Attribut besteht die Möglichkeit festzulegen, wie es im Falle einer Vereinigung zu behandeln ist. Für M:N-Tabellen kann festgelegt werden, ob die abhängigen Cluster vereinigt oder nur die qualitativ besten M:N-Zuordnungen übernommen werden sollen.

B.2 SSIS Pakete

InitDatabase

Leert die bereits vorhandene Datenbank und befüllt diese mit diversen Stammdaten, wie Länderinformationen. Auch wenn die Datenbank komplett neu angelegt wurde und noch keine Daten enthält, ist die Initialisierung notwendig.

Import DBLP

Importiert die bereits heruntergeladene DBLP-Datenbank in die Basisdatenbank. Wurde eine ältere Version der Datenbank bereits einmal importiert, besteht die Möglichkeit nur die neu hinzugekommenen oder geänderten Publikationen neu zu importieren.

DataCleaning

Führt die in Abschnitt 5.1 beschriebe Titelsäuberung von DBLP durch. Andere DataCleaning-Techniken sind an dieser Stelle denkbar, aber nicht realisiert. Dieses Paket muss nach jedem Import von DBLP, vor allem aber vor der nächsten Erstellung von Webqueries ausgeführt werden, um die Ergebnisse der Websuche nicht negativ zu beeinflussen.

Import Citeseer

Importiert die Dumps der Citeseerdatenbank. Da die Dumps fehlerhaft sind, muss nach dem herunterladen erst das Tool Repair_Citeseer_oai auf ihnen ausgeführt werden, um die meisten Fehler zu beseitigen. Trotz allem können Dumpdateien übrig bleiben, die durch die starten syntaktischen Fehler nicht importiert werden können. Dies ist im Log des Imports nachzuvollziehen.

Import Webdata

Importiert die Ergebnisse der Webdatenextraktion in die Basisdatenbank. Hierbei werden keinerlei Duplikate, auch nicht innerhalb der Webergebnisse, beseitigt. Eine Ausführung der Deduplikation ist sofort notwendig, um die Qualität und Konsistenz der Datenbank zu gewährleisten.

GenerateQueries

Erstellt die Webqueries mit Hilfe der Daten aus der Datenbank. Dieses Paket ist eine Vorlage und kann beliebig für andere Quellen und Strategien angepasst werden. Die resultierenden Xmldateien dienen als Eingabe des Webquery-Tools.

Deduplication

Findet die Duplikate in Publikationen und Autoren und vereinigt doppelte Objekte zu jeweils einem Objekt. Das hier verwendete Verfahren besteht aus einfachen Attributeditierdistanzen und HAC. Um andere sophistiziertere Verfahren zu verwenden, kann einfach dieses Paket entsprechend abgeändert werden.

Load Datawarehouse

Leert das Data Warehouse und befüllt es komplett neu mit den Daten der Basisdatenbank. Zudem werden diverse Hilfsattribute berechnet, die für spätere Auswertungen hilfreich sind. Dieses Paket muss immer ausgeführt werden, wenn sich die Basisdatenbank geändert und einen temporär finalen Zustand erreicht hat.

B.3 Webdatenextraktion

Webquery-Tool

Dieses Tool verarbeitet Xmldateien, welche aus *Webquery*-Knoten (Abschnitt 4.2.1) bestehen, entsprechend der in Abschnitt 4.2 vorgestellten Arbeitsweise. Es kann eine Begrenzung der Seitenaufrufe pro Zeitintervall eingestellt werden, um die automatische Sperrung durch verschiedene Webseiten zu umgehen. Das Tool ist komplett allgemein gehalten und beinhaltet keinerlei auf bestimmte Webseiten ausgelegte Extraktionslogik oder Abfragestrategien, beides wird durch externe XSL-Transformationen geleistet, welche der XSLT-Spezifikation 1.0 entsprechen müssen. Die Verteilung der Webabfragen auf verschiedene Rechner ist, durch die Verteilung der entstandenen Xmldateien auf verschiedene Reihe und anschließende gleichzeitige Ausführung des Webquerytools auf allen Rechnern, möglich.

XSLT für Google Scholar, ACM Portal und Citeseer

Diese Transformationen steuern die Webabfragen und Datenextraktion der jeweiligen Webdatenquelle. Im Falle von `Google Scholar` wird die in Abschnitt 4.3 beschriebene *intitle-author*-Strategie für die Suchoption verwendet. Außerdem sind verschiedene Sonderbehandlungen, etwa die Umgehung der 1000er Ergebnisgrenze für Suchanfragen realisiert.

`Citeseer` ist ebenfalls über eine Websuche abfragbar, welche in der passenden XSLT ähnlich zu `Google Scholar` genutzt wird. Da im Laufe der Arbeit ein Dump mit zusätzlichen Zitierungsinformationen bei Citeseer verfügbar wurde, hat die Transformation nur noch einen demonstrativen Charakter. Zum Beispiel besitzt die Citeseersuche ein eigenartiges Verhalten, wenn keine Suchtreffer vorhanden sind. In diesem Fall werden scheinbar willkürliche Ergebnisse zurückgeliefert, und muss daher gesondert behandelt werden.

Die Abfrage von `ACM` demonstriert eine andere Variante der Nutzung von Webseiten. Ausgehend von fest bestimmten Konferenzen und Journalen, wird lediglich Browsing

verwendet, um die Informationen zu sammeln.

B.4 Berichte

Die geschaffenen Berichte dienen als Datengrundlage für die im Analyseteil dargestellten Diagramme, welche auf den neuen Daten basieren. Dazu gehören beispielsweise das Ranking der Top 10 Autoren oder die zeitliche Entwicklung der Impaktfaktoren der einzelnen Publikationsorte.

Die Berichte können bequem über eine Weboberfläche betrachtet werden und basieren demnach stets auf den aktuellen Daten des Data Warehouses. Außerdem besteht die Möglichkeit alle Berichte oder einzelne in verschiedene Formate, wie Excel, CSV oder PDF, zu exportieren.

C Installation

Es wird empfohlen, den Datenbankserver von den anderen Aufgaben zu trennen, da sich insbesondere SSIS und der Datenbankserver gegenseitig behindern können. Sollte für alle Aufgaben ein einziger Rechner zur Verfügung stehen addieren sich entsprechend die folgenden Systemvorrausetzungen. Außerdem ist der Arbeitsspeicherverbrauch den SQL-Servers zwingend entsprechend niedrig festzulegen, um die ordnungsgemäße Ausführung der SSIS Pakete und der Analysis Services zu gewährleisten.

C.1 Systemvorrausetzungen

C.1.1 Datenbankserver

- Software:	MS SQL Server Datenbankmodul
- Arbeitsspeicher:	1GB (Empfohlen: 2GB)
- CPU:	Intel Pentium 4, 2GHz (Empfohlen: Core2Duo 2 Ghz)
- Festplatte:	mind. 30 GB freier Speicher. Insbesondere die Logdateien der Datenbank und der Temp-DB können zwischenzeitlich auf über 12 GB anwachsen.

C.1.2 Aufgabenserver

- Software:	.Net 2.0 Framework, SSIS, SSAS, SSRS, SQL Server Agent
- Arbeitsspeicher:	2GB (Empfohlen: 4GB)
- CPU:	Intel Pentium 4, 3GHz (Empfohlen: Core2Duo 3 Ghz)
- Festplatte:	mind. 10 GB freier Speicher.

C.2 Installation

Kopieren Sie den gesamten Inhalt des „Install“-Ordners in ein Verzeichnis auf der Festplatte, im Folgenden nur noch „Installationsverzeichnis“ genannt, und führen Sie die folgenden Schritte aus.

C.2.1 SSIS Komponenten

Bevor die SSIS Pakete geöffnet werden können, ist die Installation der darin zusätzlich genutzten Komponenten erforderlich. Die Installation ist, sowohl auf dem Rechner auf dem die Pakete letztlich ausgeführt werden sollen, als auch auf Rechnern, die zur Anpassung der Pakete mit dem Business Intelligence Development Studio dienen, notwendig. Es werden Dateien in den Global Assembly Cache und in das Installationsverzeichnis des SQL Servers kopiert.

- Vorrausetzung: SQL Server 2005 Integration Services lokal installiert. Angemeldet als Nutzer mit Administratorechten.
- Starten Sie das Programm Setup.exe im Installationsverzeichnis unter „SSISUtil“.
- Befolgen Sie die Anweisungen

C.2.2 SQL Server Agent Jobs

Zur komfortablen Ausführung der verschiedenen, mit der Bildung des Data Warehouses verbundenen Teilschritte, lässt sich der *SQL Server Agent* verwenden. Dieser verwaltet eine beliebige Anzahl frei konfigurierbarer Jobs. Ein Job besteht dabei aus mehreren Zwischenschritten und kann manuell oder zeitlich geplant ausgeführt werden. Da alle Jobs stets unter dem SQL Agent Nutzerkonto ausgeführt werden, ist es notwendig, dass dieses Konto gleichzeitig ein Domänenkonto ist, falls sich eine der angesprochenen Datenbanken auf einem anderen Rechner als der SQL Agent befindet. Sonst schlägt die Anmeldung am Datenbankserver fehl und es kann nicht auf die jeweilige Datenbank zugegriffen werden.

- Vorraussetzung: .Net 2.0 lokal installiert.
- Starten Sie die Applikation „InstallJobs“ im Installationverzeichnis unter „InstallJobs“
- Füllen Sie die entsprechenden Felder aus:

  - InstallDirectory - Pfad zum Installationsverzeichnis, in dem sich auch u.a. die SSIS Pakete und das Webquery-Tool befinden

  - Data Directory - Pfad zum Datenverzeichnis, unter dem u.a. die Quelldateien für DBLP oder Citeseer abgelegt werden müssen

  - SQL Agent Server - Name des Datenbankservers, auf dem die Jobs installiert werden sollen. [1]

  - Base Database Server - Name des Datenbankservers, auf dem die Basisdatenbank entstehen soll. [1]

  - Base Database Catalog - Name der Basisdatenbank.

  - Data Warehouse Server - Name des Datenbankservers, auf dem die relationale Data Warehouse-Datenbank entstehen soll. [1]

  - Data Warehouse Catalog - Name der Data Warehouse-Datenbank

  - Analysis Server - Name des Analysis Servers, auf dem die multidimensionale Datenbank entstehen soll.[1] Der Name der Datenbank lässt sich nicht beeinflussen und wird immer „Analysis“ lauten. Für eine Neuinstallation ist es daher notwendig, eine eventuell vorhandene Datenbank mit diesem Namen vorher zu entfernen.

- Klicken Sie auf Start. Es sollte eine Meldung der erfolgreich installierten Jobs erscheinen. Beim Fehlschlagen stellen Sie sicher, dass Sie die benötigten Rechte besitzen und der SQL Server Agent installiert ist und läuft.

Nach erfolgreicher Installation, enthält der entsprechende SQL Server Agent eine Reihe neuer Jobs. Alle Jobs lassen sich nun sofort manuell über das SQL Management Studio ausführen. Zur zeitlichen Planung, müssen die Eigenschaften des entsprechenden Jobs angezeigt und auf der Seite „Zeitpläne“, die entsprechenden Einstellungen vorgenommen werden. Folgende Jobs wurden angelegt:

- `Init` - Führen Sie diesen Job einmalig aus, um alle drei Datenbanken anzulegen und im Falle der Basisdatenbank mit Stammdaten, wie Länder, vorzubefüllen.

- `Update DBLP` - Aktualisiert die aus DBLP stammenden Daten oder fügt sie neu hinzu. Die entpackte DBLP Datenbank muss im Datenverzeichnis unter „DBLP“ liegen. Es werden jeweils nur die Änderungen seit dem letzten DBLP Import aktualisiert und eine Titelbereinigung durchgeführt. Das Data Warehouse wird nicht

[1] Gegebenfalls mit Angabe der Serverinstanz, falls abweichend von der Standardinstanz.

automatisch mit aktualisiert.

- `Update Citeseer` - Aktualisiert alle aus Citeseer stammenden Daten. Die Citeseer-Dumps müssen sich im Datenverzeichnis unter „Citeseer“ befinden und bereits mit dem Reparaturtool repariert worden sein. Falls es sich um eine reine Aktualisierung handelt, könnten einige Daten ignoriert werden, wenn die entsprechenden DBLP-Publikationen gematcht wurden, da DBLP eine höhere Qualität und damit Vorrang besitzt. Das Data Warehouse wird anschließend aktualisiert.

- `Update Google Scholar` - Generiert aus den DBLP Publikationen für die behandelten Konferenzen und Journale eine Reihe von Webqueries und löst diese anschließend über die Google Scholar Webseite auf. Darauf folgt der Import der Onlineergebnisse mit anschließender Duplikatbereinigung. Abschließend wird das Data Warehouse inklusive der OLAP Würfel anhand der aktuellen Basisdatenbank aktualisiert und steht danach mit den neuen Daten zur Berichterstellung zur Verfügung. Zur Anpassung der aufgelösten Publikationsorte muss das SSIS Paket „Generate Webqueries“ entsprechend angepasst werden.

- `Update ACM` - Ähnlich zu „Update Google Scholar“, mit dem Unterschied, dass die Webqueries statisch fest stehen. Zur Anpassung an weitere Publikationsorte, können die entsprechenden Quelldateien unter ACM
  Webqueries im Datenverzeichnis geändert werden.

Die automatisch erstellten Jobs sollten vielmehr als Vorlage, denn als festgelegten Prozess verstanden werden. so macht es womöglich Sinn Google Scholar und ACM gleichzeitig zu aktualisieren. Publikationsorte könnten dagegen zeitlich getrennt verarbeitet werden, um die Datenmenge zu verringern. Da die Jobs jeweils aus verschiedenen Schritten bestehen, lassen sich so die Teilkomponenten auf andere Weise zusammensetzen und dadurch an die aktuellen Anforderungen anpassen.

C.2.3 Berichte

Die Installation der Berichte ist erforderlich, um diese später von einem beliebigen Rechner über die Weboberfläche zu generieren. Dort gibt es die Möglichkeit, die Ergebnisse in verschiedene Formate, z.B. Excel, zu exportieren. Sollen die Berichte zwecks Flexibilität hinsichtlich Änderbarkeit stets direkt aus Visual Studio heraus angezeigt werden, kann dieser Punkt ausgelassen werden. In Visual Studio besteht auch die Möglichkeit, die Berichte auf die gleiche Weise zu veröffentlichen.

- Vorraussetzung: SQL Server 2005 Reporting Services lokal installiert. Alternativ können die Berichte auch Remote installiert werden, wenn das Tool RS.exe lokal verfügbar ist.

- Öffnen Sie die Kommandokonsole
- Wechseln Sie unter dem Installationsverzeichnis nach „Publication Reports“
- Führen Sie den Befehl: installreports <Report Server Instanz> <Datenbank Server> <Katalog> aus
  - <Report Server Instanz> - Basiswebadresse der Report Server Instanz.
  - <Datenbank Server> - Name des Datenbankservers mit der SSAS Datenbank. Wurde der Server nicht als Standardinstanz installiert, muss hier der Instanzname mit angegeben werden.
  - <Katalog> - Name der SSAS-Datenbank
- Bsp: $> installreports http://localhost/ReportServer localhost Analysis

Die Standardberichte, welche die Grundlage für den Analyseteil gebildet haben, sind im Beispiel nach erfolgreicher Installation unter:
`http://localhost/ReportServer?/Publication%20Reports/Zitierungsanalyse&rs:Command=Render`
als Webseite verfügbar.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[AAGY01] Charu C. Aggarwal, Fatima Al-Garawi, and Philip S. Yu. Intelligent crawling on the world wide web with arbitrary predicates. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 96–105, New York, NY, USA, 2001. ACM.

[BACÖAH05] Sulieman Bani-Ahmad, Ali Cakmak, Gultekin Özsoyoglu, and Abdullah Al-Hamdani. Evaluating Publication Similarity Measures. *IEEE Data Eng. Bull.*, 28(4):21–28, 2005.

[BG85] Lawrence D. Brown and John C. Gardner. Using Citation Analysis to Assess the Impact of Journals and Articles on Contemporary Accounting Research (CAR). *Journal of Accounting Research*, 1985.

[BG04] Andreas Bauer and Holger Günzel. *Data Warehouse Systeme.* dpunkt-Verl., 2004.

[BYRN99] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[CD97] Surajit Chaudhuri and Umeshwar Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1):65–74, 1997.

[Cit] *Citeseer.* `http://citeseer.ist.psu.edu`.

[CSB] *The Collection of Computer Science Bibliographies.* `http://liinwww.ira.uka.de/csbib`.

[DGC06] Kristine Daniels and Christophe Giraud-Carrier. Learning the threshold in hierarchical agglomerative clustering. In *ICMLA '06: Proceedings of the 5th International Conference on Machine Learning and Applications*, pages 270–278, Washington, DC, USA, 2006. IEEE Computer Society.

[DHM05] Xin Dong, Alon Halevy, and Jayant Madhaven. Reference reconciliation in complex information spaces. *International Conference on the Management of Data (SIGMOD)*, 2005.

[Dit99] Carsten Dittmar. Erfolgsfaktoren für Data Warehouse-Projekte. *Institut für Unternehmensführung, Arbeitsbericht Nr. 78*, 1999.

[DLLH03] AnHai Doan, Ying Lu, Yoonkyong Lee, and Jiawei Han. Object matching for information integration: A profiler-based approach. *IIWeb*, 2003.

[EIV07] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[fCM] Association for Computing Machinery. *ACM Portal.* `http://portal.acm.org`.

[Gar55] Eugene Garfield. Citation Indexes for Science. *Science*, 1955.

[Gar79] E. Garfield. Is citation analysis a legitimate evaluation tool? *Scientometrics*, 1979.

[Goo] *Google Scholar.* `http://scholar.google.com`.

[HQW06] Gerhard Heyer, Uwe Quasthoff, and Thomas Wittig. *Text Mining: Wissensrohstoff Text.* W3L-Verlag, 2006.

[JBdS07] Marko A. Rodriquez Johan Bollen and Herbert Van de Sompel. Journal status. *Scientometrics*, 2007.

[KM05] Dmitri V. Kalashnikov and Sharad Mehrotra. A probabilistic model for entity disambiguation using relationships. *SIAM International Conference on Data Mining (SDM)*, 2005.

[LdS06] Carl Lagoze and Herbert Van de Sompel. Open Archive Initiative (OAI). `http://www.openarchives.org/`, 2006.

[Ley] Michael Ley. *DBLP Computer Science Bibliography.* `http://dblp.uni-trier.de`.

[LKM+07] Dongwon Lee, Jaewoo Kang, Prasenjit Mitra, C. Lee Giles, and Byung-Won On. Are your citations clean? *Commun. ACM*, 50(12):33–38, 2007.

[Lov06] Chris Lovett. SgmlReader 1.7. `http://www.lovettsoftware.com/`, 2006.

[Mac66] James B. MacQueen. Some methods for classification and analysis of multivariate observerations. *Proceedings of the 5th Berkeley Symposium on mathematical statistics and probability*, 1966.

[Mic] Microsoft. *Live Search Academic*. `http://search.live.com/results.aspx?scope=academic`.

[MM96] M. H. MacRoberts and Barbara R. MacRoberts. Problems of citation analysis. *Scientometrics*, Volume 36, 1996.

[OEL+06] Byung-Won On, Ergin Elmacioglu, Dongwon Lee, Jaewoo Kang, and Jian Pei. Improving Grouped-Entity Resolution Using Quasi-Cliques. *icdm*, 0:1008–1015, 2006.

[oK06] ISI Web of Knowledge. Journal Citation Reports. `http://isiwebofknowledge.com`, 2006.

[RT05] Erhard Rahm and Andreas Thor. Citation analysis of database publications. *SIGMOD Record*, 34(4):48–53, 2005.

[Sco] *Scopus*. `http://www.scopus.com`.

[SHW+06] George Spofford, Sivakumar Harinath, Chris Webb, Dylan Hai Huang, and Francesco Civardi. *MDX Solutions, Second Edition: With Microsoft® SQL Server™ Analysis Services 2005 and Hyperion® Essbase*. Wiley Publishing, Inc., 2006.

[SLD05] Warren Shen, Xin Li, and AnHai Doan. Constraint-based entity matching. *Proc. of National Conf. on Artificial Intelligence (AAAI)*, 2005.

[TAR07] Andreas Thor, David Aumueller, and Erhard Rahm. Data Integration Support for Mashups. *Sixth International Workshop on Information Integration on the Web*, 2007.

[W3C06] W3C. Extensible Markup Language (XML) 1.0 (Fourth Edition). `http://www.w3.org/TR/REC-xml/`, 2006.

[WEP08] III William E. Pearson. MDX Essentials. `http://www.databasejournal.com/article.php/1459531/`, 2002-2008.

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort Datum Unterschrift