Web Mining und Farming

Thema: Business Intelligence-

Teil 2: Data Mining & Knowledge Discovery

Von Shenwei Song

Betreuer: Dipl.-Inform. Michael Haustein

1. Februar 2004

Web Mining und Farming

1 Übersicht über Web Mining und Farming

Heutzutage spielt das Internet durch seine schnelle Entwicklung eine immer größere Rolle in unserem Leben. Wie kann das Web als Informationsressource und Kommunikationsmittel besser verwendet werden? Wie können Firmen, die im Bereich des E-Commerce tätig sind, konkurrenzfähig bleiben? Es wird versucht, durch neue Ansätze wie *Web Intelligence* Lösungen zu finden. Web Intelligence ist eine Mischung aus künstlicher Intelligenz und den Informationstechniken, die im Internet eingesetzt werden.

Ziel dabei ist es, ein so genanntes *Intelligent Web Information System* auf neuen Plattformen im Internet zu entwerfen und zu implementieren. Das Intelligent Web Information System könnte die Funktionalitäten, die sich auf die menschliche Intelligenz (beispielsweise Analyse- und Lernvorgänge) beziehen, ausführen. *Web Mining* und *Web Farming* sind ein Teil der Web Intelligence.

Da Firmen, Institutionen und Individuen das Web verwenden, um ihren öffentlichen Auftritt zu gestalten und um kommerziell tätig zu sein, gibt es im Web viele potentiell verwertbare Informationen. Web Intelligence bietet einen automatischen Prozess zum Auffinden von Informationen: das Web Mining.

Web Mining ist ein Teilbereich des Data Mining. Data Mining behandelt die Extraktion brauchbarer Informationen aus Daten, die ihrerseits in Datenbanken, Data Warehouses oder anderen Datenspeichern organisiert sind. Manchmal wird Data Mining auch als Synonym für *Knowlege Discovery in Databases* (KDD) gebraucht. KDD setzt Datenanalyse- und Entdeckungsalgorithmen ein, um mit akzeptabler Recheneffizenz nützliche Muster und Informationen zu finden. Web Mining, welches die Informationen im Web ausfindig macht, ist eine Anwendung der Data-Mining-Techniken auf unterschiedlichen Web-Datenressourcen. Seit der Erfindung des Konzepts Web Mining gibt es mehr und mehr Firmen, die sich dafür interessieren. Es wird umfangreich im E-Commerce eingesetzt:

- Zum Durchforsten des Internets, um neue Kunden oder Geschäftsmöglichkeiten zu finden, um Ankündigungen auf Nachrichtenseiten zu entdecken und Interessen auf privaten Web-Seiten zu erkunden.
- Zum Beobachten und Analysieren der Konkurrenz, zum Beispiel in Hinsicht auf Preisaktualisierungen, Vorstellungen von neuen Produkten und angebotenen Serviceleistungen.
- Zur automatischen Aktualisierung von Datenbanken und zum Durchsuchen der Web-Seiten von Kunden und Interessenten, um Informationen daraus zu erhalten und damit Data Warehouses oder die Datenbanken des *Customer Relationship Management* (CRM) zu füllen，außerdem zur Automatisierung des Verkaufs und von Call-Center-Systemen.
- Zum Zusammentragen des Inhalts aus unterschiedlichen Web-Seiten, um ein zusammenfassendes Dokument wie ein Mitgliederverzeichnis, einen Newsletter, einen Produktkatalog oder ein Portal zu erstellen.

Web Farming ist definiert als die systematische Verfeinerung der web-basierten

Informationen für Business Intelligence. Im Vergleich zum Web Mining hat das Web Farming eine andere Bedeutung. Es bedeutet harte Arbeit: „Boden bestellen“, „Feldarbeit verrichten“, um eine Web-Informationsressource aufzubauen. *Systematisch* bedeutet im Informationstechnikbereich eine zentrale Sicherung und Verwaltung der Daten in Data Warehouses. Die Aufgabe des Web Farming ist die Verbesserung der Data Warehouses durch die Integration externer Informationen mit Daten aus internen Operationssystemen. Die spezifischen Aufgaben sind:

- Entdeckung von Web-Inhalten, die relevant für das Geschäft sind.
- Erwerbung von Web-Inhalten, damit sie richtig in geschichtlichem Kontext validiert werden.
- Strukturierung von Inhalt in eine nutzvolle Form, die kompatibel mit dem Data Warehouse ist.
- Unterbreitung des Inhalts einer verantwortlichen Person, damit diesen einen positiven Effekt auf die spezifischen Geschäftsprozesse hat.
- Systematische Verwaltung der obigen Schritte.

Web Farming kann die Leistung von Geschäftsaktivitäten verbessern. Viele der neuen Techniken werden im Web Farming eingesetzt, zum Beispiel XML-strukturierte Dokumente, linguistische Analyse und Informationsvisualisierung.

2 Web Mining

2.1 Klassifikation des Web Mining

2.1.1 Einführung

Es gibt einige unterschiedliche Klassifikationen des Begriffs Web Mining, die auf der Unterscheidung der Funktionalitäten oder der Anwendungsbereiche basieren. Kosala und Blockeel [1] haben die Kategorien von Etzioni [2] erweitert und vier Typen von Aufgaben definiert:

A. Lokalisierung von Ressourcen: Finden benötigter Dokumentationen, Informationen und Dienste.
B. Extraktion von Informationen: Automatische Extraktion benötigter Informationen aus den entdeckten Quellen im Netz.
C. Generalisierung: Finden generischer Muster aus individuellen Web-Seiten und aus Verknüpfungen mehrerer Web-Seiten.
D. Analyse: Validierung und/oder Interpretation der entdeckten generischen Muster.

Diese Typen enthalten Prozesse, die Schnittstellen bei der Extraktion nutzvoller Informationen aus dem Web bieten. Außerdem sind sie unabhängig von den Datentypen, die im Web Mining benutzt werden. Verschiedene Techniken werden in entsprechenden Schnittstellen eingesetzt, beispielsweise der Einsatz von wissen-basierten Wrappern oder Induktion bei der Extraktion von Informationen.

Madria et al. [3] benutzen Web Data Mining, um die Muster aus Web-Daten zu entdecken. Sie haben drei Unterbereiche des Web Mining definiert, die auf den benutzten Datentypen basieren:

A. Web Content: Die Daten, aus der die Web-Seite besteht, übermitteln Benutzerinformationen. Solche Daten sind beispielweise HTML, Video/Audio-Dateien oder Grafiken auf Web-Seiten.
B. Web Structure: Die Hyperlink-Struktur zwischen Web-Seiten.
C. Web Usage: Die Daten, die die Nutzung von Web-Ressourcen beschreiben, beispielweise Einträge im Protokoll eines Web Browsers und temporäre Internetdateien, oder die Log-Dateien von Proxy-Servern und Web-Servern.

2.1.2 Web Content Mining

Web Content Mining beschreibt die Übermittlung der nützlichen Information aus einem Web-Inhalt oder einem Dokument. Viele unterschiedliche Techniken werden im Web Content Mining verwendet. Die meisten Web-Content-Mining-Methoden basieren auf unstrukturierten Textdaten oder semistrukturierten HTML-Dokumentdaten, zum Beispiel das Text Mining [4] und die Text Kategorisierung [5].

Kosala und Blockeel [1] haben das Web Content Mining in zwei Bereiche zerlegt: *Information Retrieval View* und *Datenbank-View*. Das Ziel des Unterbereichs Information Retrieval View ist die Unterstützung des Auffindens der Informationen oder der Filterung der Information. Das Mining-Ergebnis kann in den Web-Suchmaschinen, der Web-Personalisierung und den Empfehlungssystemen angewendet werden. Das Ziel des Web Content Mining aus Datenbanksicht ist die Modellierung der Daten für Netzanwendungen und die Integration der Daten im Netz, damit komplizierte Abfragen ausgeführt werden können. Die Ergebnisse des Mining können beim Aufbau von Web Warehouses und Datenbanken eingesetzt werden.

Eine Web-Content-Mining-Applikation beinhaltet eine Klassifikation der Web-Dokumente, Methoden für ein Clustering von Web-Seiten in inhaltsbasierten Empfehlungssystemen [6], Methoden zum Vergleich der Web-Inhalte [7], eine Modellierung von Dokumentstrukturen [8] und eine Unterstützung anderer Web-Mining-Applikationen.

2.1.3 Web Structure Mining

Web Structure Mining versucht, das Modell der Verweisstrukturen des Web zu entdecken. Das Modell reflektiert die Topologie der Verweise zwischen den Web-Seiten. Durch das Web Structure Mining ist es möglich, Web-Seiten in *Authority- oder Hub-Seiten* zu klassifizieren, Informationen über die Ähnlichkeit und den Unterschied zwischen Web-Seiten zu generieren und Informationen über das Zugriffsverhalten der Benutzer auf die Seiten zu sammeln.

2.1.4 Web Usage/Log Mining

Web Usage Mining versucht die Zugriffsmuster aus den Web-Daten zu entdecken. Aus den vom Web Usage Mining entdeckten *Usage-Mustern* kann man die Bedürfnisse der web-basierten Applikationen verstehen und besser erfüllen [9]. Durch Analyse von *Web Server Logs* können wertvolle Informationen über Benutzerverhalten gesammelt werden, damit man die Web-Seiten besser organisieren und effektiver präsentieren kann. In den Unternehmen, die ein Intranet benutzen, sind solche Informationen besonders wichtig, weil durch solche Informationen die Arbeitsgruppenkommunikationen und organisatorische Infrastrukturen besser verwaltet werden können.

Ein *Web Server Log* ist eine explizite Aufzeichnung von Benutzerabfragen bezüglich angebotener Web-Seiten oder Ressourcen auf Serverseite. Es ist die Hauptdatenquelle für Web Usage/Log Mining. Andere Web Logs, zum Beispiel Proxy Server Logs, Protokolldateien und Cookies, erfassen ebenfalls die Benutzerabfragen. Die meisten Web Usage Minings sind auf Web Server Logs [2] basiert. Srivastava et al. [9] haben das Konzept der Usage-Daten erweitert, sodass es die Logs aus entfernten Agenten enthält. Borges und Levene [10] diskutieren zwei generelle Einsatzmöglichkeiten zum Mining von Benutzernavigationsmustern aus Log Files. Die erste Möglichkeit bildet die Log-Daten in relationale Tabellen oder einen Datenkubus ab und wendet dann Data-Mining-Techniken auf die Log-Daten [11] an. Die zweite Möglichkeit wendet die Data-Mining-Techniken direkt auf Log Files [12] an. Srivastava et al. [9] schlagen einen dreistufigen Web-Usage-Mining-Prozess vor: Vorbearbeitung, Entdeckung von

Mustern und Musteranalyse. Die entstehenden schematischen Beschreibungen können in mehreren Schritten verfeinert werden.

2.1.5 Kombination von Web Content, Structure und Usage Mining

In vielen Fällen sind Webinhalt-, Struktur- und Usage-Daten in der gleichen Datei zusammengefasst. Die Informationen tauchen zum Beispiel in den Log-Daten auf, oder die Web-Structure-Daten beinhalten nutzvolle Inhaltsinformation. Die drei Kategorien müssen nicht voneinander isoliert sein. Das Web Content Mining muss manchmal Daten über die Web-Struktur zur Klassifikation einer Web-Seite benutzen. Ähnlich dazu muss das Web Usage Mining auch manchmal die Web-Inhaltdaten und die Web-Strukturdaten nutzen.

In den folgenden Abschnitten werden ein paar Web-Mining-Techniken vorgestellt. Manche Web-Mining-Techniken und -Applikationen beziehen sich gleichzeitig auf Webinhalt-, Struktur- und Usage-Mining. Durch die Kombination kann man aussagkräftige Informationen bekommen, damit das vordefinierte Ziel besser erreicht werden kann.

2.2 Web-Mining-Techniken

2.2.1 Entdeckung indirekter Beziehungen aus Web-Usage-Daten

2.2.1.1 Einführung

Web-Usage-Daten beinhalten Web-Muster, die nützliche Informationen über das Verhalten von Web-Benutzern bieten können. Mehr und mehr Forschungen werden in Entdeckung der Muster aus Web-Usage-Daten betrieben. Ein interessanter Mustertyp ist das *Beziehungsmuster*, das die Informationen über häufig zusammen in eine *Web Session* oder *Web-Transaktion* gleichzeitig auftretende Web-Seiten enthält. Die existierenden Mining-Techniken fokussieren das Auffinden der Muster mit hohem *Support*, wobei der Support eines Musters aus einer Datenmenge bestimmt wird (weitere Informationen über Support werden im folgenden Abschnitt gegeben). Andere Muster mit niedrigem Support werden als unwichtige Muster gekennzeichnet und eliminiert. Viele nützliche Informationen werden durch solche Mining-Techniken nicht entdecket, zum Beispiel die sich negativ-beziehenden Web-Seiten eines Musters. Sich negativ-beziehenden Web-Seiten sind die Web-Seiten, die selten in einer Web-Transaktion oder Web Session gleichzeitig auftreten und mittels gemeinsamen Web-Seiten, zum Beispiel gemeinsamen Navigationsseiten, miteinander indirekt verbunden sein können. Solche Muster können das Browsing-Verhalten unterschiedlicher Gruppen der Web-Benutzer repräsentieren. Um solche *negativen Beziehungen* aus Web-Usage-Daten zu entdecken, wird eine neue Mining-Technik eingesetzt, die *indirekte Beziehung* genannt wird. Die Idee hinter der indirekten Beziehung ist, ein Paar von binären Variablen zu finden, die unmittelbar miteinander negativ verbunden, aber mittels einer Menge von Elementen positiv verbunden sind. Diese Menge wird als *Mediator* bezeichnet. Ein Beispiel der indirekten Beziehung aus Web-Usage-Daten wird im folgenden Bild gezeigt:

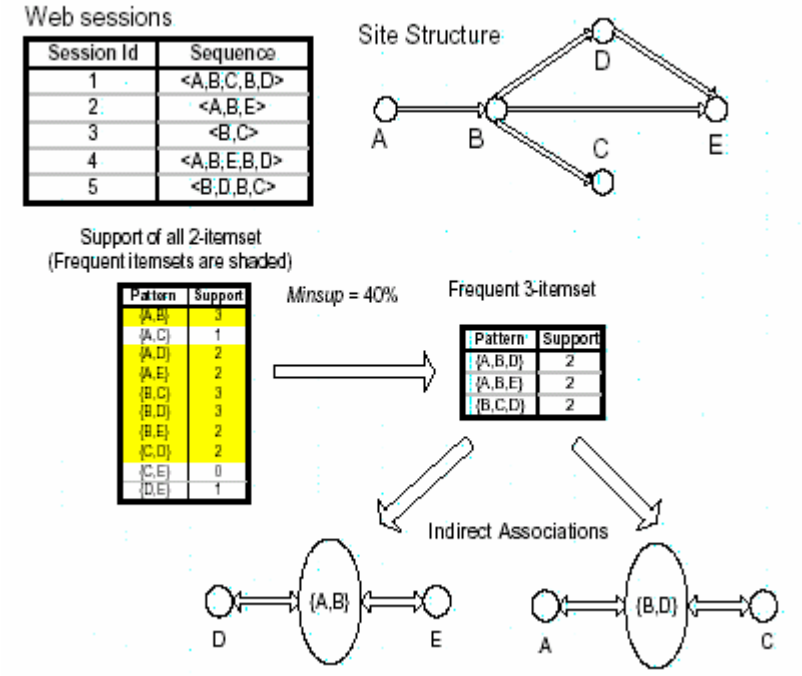


Bild 2-2-1 Beispiel der indirekten Beziehung aus Web-Usage-Daten

In Bild 2-2-1 ist eine Web Session und die Struktur der Web-Seite gegeben. Zuerst werden alle möglichen in einer Web Session verbundenden Web-Seiten gefunden und jeweils in einer Menge gruppiert. Dann werden die Muster, deren Support weniger als 2 ist, eliminiert, weil der *Minimum-Support-Grenzwert* 40% ist und es 5 Sessions in Bild 2-2-1 gibt. Der Support von häufig benutzten 2er-Mustern und 3er-Mustern wird in Bild 2-2-1 gezeigt. Offensichtlich tritt Web-Seite D häufig mit Web-Seiten A und B zusammen auf. Ähnlich tritt Web-Seite E auch häufig mit Web-Seiten A und B auf. Weil der Support zwischen Web-Seiten E und D niedriger als der Minimum-Support-Grenzwert ist, sind E und D durch die *Mediator-Menge* {A,B} miteinander negativ verbunden. Analog existiert eine indirekte Beziehung auch zwischen A und C durch die Mediator-Menge {B,D}.

Das Konzept des Support kommt aus dem *Association Rule Mining* [13]. Association Rule Mining ist eine wichtige Data-Mining-Technik, weil *Association Rules* die wichtigen Beziehungen zwischen Attributen einer Datenmenge entdecken können. In einer Association Rule werden zwei Parameter definiert: Support s und *Confidence* c. Die Definitionen von s und c sind gegeben durch:

Sei I die Menge aller Attribute einer Datenbank und T die Menge der Datenbanktransaktionen.
Sei $A \to B$ eine Association Rule, wobei $A, B \subset I$ und $A \cap B = \phi$. Die Rule hat Support s, wenn s% von den Transaktionen in T $A \cup B$ enthalten. Die Rule hat Confidence c, wenn c% von den Transaktionen, die A enthalten, auch B enthalten. Die Formeln zur Berechnung s und c sind im Folgenden definiert:

$$s(A \to B) = \frac{\sigma(A \cup B)}{|T|}, c(A \to B) = \frac{\sigma(A \cup B)}{\sigma(A)}$$

wobei $\sigma(A \cup B)$ die Anzahl der Transaktionen ist, die $A \cup B$ enthalten, $\sigma(A)$ die Anzahl der Transaktionen, die A enthalten und |T| die Anzahl aller Transaktionen ist. Das Ziel von Association Rules ist es, dass alle Rules ermittelt werden, deren Support größer als der vordefinierte Minimum-Support-Grenzwert ist, und deren Confidence größer als der vordefinierte *Minimum-Confidence-Grenzwert* ist. Ein Beispiel der Berechnung des Support ist im Folgenden gegeben:

Session	Websites
1	Homepage, Metro
2	Homepage, Metro, World, Sports
3	Homepage, Sports, TV, Metro
4	Homepage, Sports, TV, Entertainment
5	Homepage, Metro, Weather

Tabelle 2-2-1 Repräsentierung von Web Session als Transaktion

Beobachten wir die Web Sessions in Tabelle 2-2-1. Diese Datenmenge hat 5 Web Sessions (Web Transaktionen) und 7 Websites (Attribute). Der Support der Sports Website ist 0.6, weil 3 Sessions die Sports Website beinhalten. Der Support der TV Website ist 0.4. Der Support der Website-Menge {Sports, TV} ist 0.4, weil 2 Sessions diese Menge beinhalten. Deshalb hat die Association Rule TV->Sports Support 40% und Confidence 100%.

In den folgenden Abschnitten wird zuerst die Definition der indirekten Beziehungen

gegeben, dann wird ein Mining-Algorithmus der indirekten Beziehungen (*Indirekter Algorithmus*), und eine optimierte Darstellung der indirekten Beziehungen vorgestellt. Am Ende wird eine Zusammenfassung über indirekten Beziehungen gegeben.

2.2.1.2 Definition

In diesem Abschnitt, werden ein paar allgemeine Definitionen und zwei Formeln der indirekten Beziehungen beschrieben.

2.2.1.2.1 Allgemeine Definition

- I = $\{ i_1, i_2, \ldots, i_d \}$ ist die Menge aller nützlichen Items in einer Datenbank. Jede nicht leere Teilmenge von I heißt *Item-Menge.*
- $\mathcal{K}$ ist die Länge der Item-Menge. Die Länge entspricht der Anzahl der Items in der Item-Menge.
- Eine *Sequenz* ist ein geordnete Liste von Item-Mengen, S = $s_1 \text{->} s_2 \text{->} \ldots \text{->} s_n$, wobei jede Item-Menge ein Element der Sequenz ist. Die Anzahl der zur s_j gehörenden Items ist notiert als $|s_j|$. Wenn die gesamte Anzahl der zur Sequenz S gehörenden Items k ist, dann wird S eine *k-Sequenz* genannt.
- Ein Item kann mehrere Male in einer Sequenz, aber nur ein Mal in einem Element auftauchen. Wenn ein Item nur ein Mal in einer Sequenz auftaucht, wird es als *unique* oder *nicht-wiederholtes Item* bezeichnet. Eine Sequenz T = $t_1 \text{->} t_2 \text{->} \ldots \text{->} t_m$ ist eine *Subsequenz* von S, wenn jedes geordnete Element in T eine Teilmenge von einem geordneten Element in S ist. Ist beispielsweise S = {a}->{a,c}->{b} und T = {a,c}->{b}, dann ist T eine Subsequenz von S.
- Eine Sequenz W ist ein *Präfix* von S, wenn es eine nicht leere Subsequenz Y von S gibt und S = WY. W ist ein *minimales Präfix* von S, wenn W ein Präfix von S und die Länge von W eins ist. Umgekehrt ist Y ein *Suffix* von S.
- Ein Item x ist *Prefix-Item* der Sequenz s, wenn x das einzige zum ersten Element gehörende Item ist. Das heißt, dass $x \in s_1$ und $|s_1| = 1$. Umgekehrt ist x *Suffix-Item* der Sequenz s: $x \in s_n$ und $|s_n| = 1$. x ist *End-Item* der Sequenz s, wenn s Prefix- Item oder Suffix- Item ist.

Eine *Sequenzdatenbank* D besteht aus Tupeln <sid,s>, wobei sid ein *Sequenzidentifizierer* und s eine Sequenz ist. Eine Datenbank aus Web-Usage-Daten kann als Sequenzdatenbank formuliert werden. Jedes Tupel entspricht einer individuellen Session, wobei sid der Sessionidentifizierer und s die Sequenz von Web-Seiten ist, auf die innerhalb der Session zugegriffen wurde.

2.2.1.2.2 Nicht-sequenentielle indirekte Beziehung

Definition: Ein Paar von Items (a, b) bezieht sich indirekt auf die Mediator-Menge M, wenn die folgenden Bedingungen erfüllt werden:

1. Sup({a, b}) < t_s , wobei t_s als Support-Grenzwert definiert ist (*Itempaar-Support-Bedingung*).
2. Es gibt eine nicht-leere Menge M sowie:

a) Sup({a} ∪ M) $\geqslant t_f$, Sup({b} ∪ M) $\geqslant t_f$, wobei t_f als Support-Grenzwert definiert ist (*Mediator-Support -Bedingung*).
b) D({a}, M) $\geqslant t_d$, D({b}, M) $\geqslant t_d$, wobei t_d als Grenzwert definiert ist und D(P,Q) ein Maß der Abhängigkeit zwischen den Item-Mengen P und Q ist (*Mediator-Abhängigkeit-Bedingung*).

Die erste Bedingung fordert, dass der Verbindungs-Support (joint Support) zwischen a und b sehr niedrig sein soll, da die indirekte Beziehung nur dann ein sinnvolles Konzept ist, wenn die beiden Items kaum zusammen in einer Transaktion auftreten.
Bedingung 2a fordert, dass der Mediator M kooperativ häufig mit a und b zusammen auftaucht. Das ist nötig, um die statistische Signifikanz von M zu garantieren.

Bedingung 2b fordert, dass die Item-Mengen in Mediator M von sowohl a als auch b quasi abhängig sein müssen. Diese Bedingung wird zur Eliminierung der indirekten Beziehungen, die mittels uninformativem Mediator verbunden sind, benutzt. Zum Beispiel gibt es ein Item K (Homepage), der in jede Transaktion auftritt. Ohne die Bedingung 2b bezieht sich jedes Item-Paar indirekt auf K, damit würden viele uninformative indirekte Beziehungen generiert werden. Es gibt viele interessante Messverfahren [14], die in Abhängigkeitsmessungen der Bedingung 2b eingesetzt werden können. Eins davon ist *φ Koeffizient.* Der ϕ Koeffizient zwischen zwei Item-Menge X und Y ist gegeben durch:

$$\phi_{X,Y} = \frac{P(X;Y) - P(X)P(Y)}{\sqrt{P(X)(1-P(X))P(Y)(1-P(Y))}}$$

wobei P(X) und P(Y) die Wahrscheinlichkeiten sind, mit der die Item-Mengen X und Y in den Transaktionen auftreten. Die Wahrscheinlichkeiten können durch Benutzung des Support der Item-Mengen berechnet werden. Wenn P(X)<<1, P(Y)<<1 und $\frac{P(X,Y)}{P(X)P(Y)}$ >> 1 sind, können die Beziehung zwischen X und Y, $\phi_{X,Y}$ in Termen ihres Interessensfaktors beschrieben werden: I(X,Y) ≡ $\frac{P(X,Y)}{P(X)P(Y)}$ [15], und ihre Beziehungswahrscheinlichkeit : $\phi_{X,Y} \approx \sqrt{I(X,Y) * P(X,Y)}$.

2.2.1.2.3 Sequenentielle indirekte Beziehung

Sei A ein nicht-wiederholtes End-Item aus der Sequenz $s_1 = A_1 \text{->} A_2 \text{->...->} A_n$ und B ein nicht-wiederholtes End-Item aus der Sequenz $s_2 = B_1 \text{->} B_2 \text{->...->} B_n$. Sei A* das Element in s_1 , das A beinhaltet, und B* das Element in s_2 , das B beinhaltet. Weil A und B beide End-Items sind, ist A* = A_1 ={A} oder A* = A_n ={A} und B* = B_1 = {B} oder B* = B_n = {B}.
Definition:
Ein Paar von End-Items (A,B) bezieht sich indirekt auf die *Mediator-Sequenz* W, wenn s_1 = WA* oder s_1 = A*W, s_2 = B*W oder s_2 = WB*, und die folgenden Bedingungen erfüllt werden:
1. Sup({a, b}) < t_s , wobei t_s als Support-Grenzwert definiert ist (Item-Paar-Support-Bedingung).

2. Es gibt ein nicht leere Sequenz W so wie:
   a) Sup(S1) $\geqslant t_f$, Sup(S2) $\geqslant t_f$, wobei t_f als Support-Grenzwert definiert ist (Mediator-Support-Bedingung).
   b) D(A*, W) $\geqslant t_d$, D(B*, W) $\geqslant t_d$, wobei t_d als Grenzwert definiert ist und D(P,Q) ein Maß von der Abhängigkeit zwishcen die Item-Menge P und Q ist (Mediator-Abhängigkeit-Bedingung).

Die Bedingung 1 vernachlässigt die Reihenfolge, in der A und B in der Datensequenz auftreten. Der Wert von t_s wird nahe zu null gewählt, um dadurch zu garantieren, dass A und B selten in der gleichen Sequenz auftreten. Bedingung 2a garantiert, dass nur frequente Pfad in die Analyse eingebracht werden. Bedingung 2b hat ähnliche Anforderungen, wie sie in der nicht-sequenentiellen indirekten Beziehung definiert sind.

2.2.1.3 Indirekter Algorithmus

Der indirekte Algorithmus zum Mining indirekter Beziehungen zwischen Item-Paaren ist definiert als:

1.Extrahiere die Artikelmengen L_1 , L_2 ,..., L_n durch standardisierten Mining-Algorithmus.
2. P = Ø
3. For k = 2 to n do
4. C_{k+1} <- join(L_k , L_k)
5. Für jede (a,b,M) $\in C_{k+1}$ tue folgendes:
6. If (sup({ a,b}) $< t_s$ und d({a},M) $\geqslant t_d$ und d({b}, M) $\geqslant t_d$)
7. P = P $\cup$ (a, b, M)
8. Ende
9. Ende

Zunächst wird dieser Algorithmus für das Mining zur Entdeckung der nicht-sequentiellen indirekten Beziehungen eingesetzt. Es gibt zwei Phasen in diesem Algorithmus:
1. Generierung der Kandidaten.
2. Durchsieben der Kandidaten.

Während der ersten Phase wird ein standardisierter Algorithmus wie beispielsweise Apriori [16] eingesetzt, um frequente Item-Mengen zu generieren. Der Apriori Algorithmus benutzt folgende Heuristik: Wenn eine k-Item-Menge frequent ist, dann müssen alle ihre Teilmenge auch frequent sein und durchsucht rekusiv die ganze Datenbank, bis keine frequente Item-Menge mehr gefunden wird. Dann werden die Item-Mengen zur Generierung der Kandidaten für eine indirekte Beziehung kombiniert, die die Länge k+1 haben. Jeder Kandidat in C_{k+1} ist ein Tripel (a, b, M), wobei a und b indirekte Item-Mengen sind und sich auf M beziehen. Während des Join-Abschnitts wird jeweils ein Paar von den k-Item-Mengen $(a_1, a_2, .., a_k)$ und $(b_1, b_2, .., b_k)$ zusammengebaut, um eine indirekte Beziehung zu erzeugen (a, b, M), wenn die beiden Item-Mengen gleiche k-1 Items haben. Weil C_{k+1} aus frequenten

Item-Mengen generiert ist, wird die Mediator-Support-Bedingung schon erfüllt. Die Durchsiebungsphase (von Schritt 5 bis 7) ist zur Eliminierung der Kandidaten, die die Item-Mengen-Support-Bedingung und Mediator-Abhängigkeit-Bedingung nicht erfüllt haben.

Der Einsatz des indirekten Algorithmus im Mining für sequentielle indirekte Beziehungen ist ähnlich wie im Mining für nicht-sequentielle indirekte Beziehungen. L_k ist eine frequente k-Sequenz, die durch die Anwendung eines sequentiellen Musterentdeckungsalgorithmus wie GSP [17] erzeugt wird. Während der Join-Schritte wird ein Paar von frequenten Sequenzen s_1 und s_2 zu einem indirekten Beziehungskandidaten (a, b, W) kombiniert, allerdings nur, wenn a und b nicht-wiederholte End-Items von s_1 und s_2 sind. Es gibt ein paar Optimierungsmöglichkeiten, um die Anzahl der Join-Operationen und Anzahl der indirekten Beziehungskandidaten zu reduzieren. Weil nur die indirekten Beziehungen zwischen End-Items interessant sind, kann die Join-Operation nur zwischen Sequenzen, deren Länge größer als 2 ist und deren Längenunterschied kleiner als 2 ist, durchgeführt werden.

2.2.1.4 Kombination von indirekten Beziehungen

Obwohl indirekte Beziehungen die interessanten negativen Beziehungen zwischen Items effizient entdecken können, könnte die Anzahl der gefundenen Muster sehr groß sein. In diesem Abschnitt wird dargestellt, wie negative Beziehungen in kompakte Muster zusammengebaut werden können. Es bringt zwei große Vorteile: erstens wird die Anzahl der zur Analyse präsentierten Sequenzen stark reduziert, zweitens sind die kombinierten Muster viel informativer als individuelle negative Beziehungen. Es gibt zwei Wege zum Zusammenbau von negativen Beziehungen: Zusammenbau von negativen Beziehungen zwischen gleichen Items und Zusammenbau von negativen Beziehungen zwischen Items mit gleichen Mediatoren.

Im Zusammenbau von negativen Beziehungen zwischen gleichen Items, der *Item-Paar Sicht* heißt, wird die Anzahl der Muster auf die Anzahl der einzigartigen (unique) indirekten Item-Paare in der Datenmenge reduziert. Zum Beispiel, wenn (a, b) sich indirekt auf {c}, {c, d} und {d, e} bezieht, dann kann eine Item-Menge durch Gruppierung aller Item-Mengen des Mediators ({c}, {c,d}, {d,e}) generiert und als Mediator in ein kompaktes Muster eingesetzt werden. Es wird in Schaubild 2-2-2 (a) gezeigt. Durch Item-Paar Sicht ist es eindeutig, mit welchen Mediatoren Items miteinander indirekt verbunden sind.

Im Zusammenbau von den negativen Beziehungen zwischen Items mit gleichen Mediatoren, die *Mediator Sicht* heißt, werden alle indirekten Item-Paare, die den gleichen Mediator haben, zusammen in einer Struktur gruppiert, und die Anzahl der Muster wird auf die Anzahl der einzigartigen Mediatoren in der Datenmenge reduziert. Schaubild 2-2-2 (b) zeigt ein Beispiel von der Mediator Sicht. Die durchgezogenen Linien bedeuten, dass der Support zwischen zwei Item-Gruppen hoch ist. Die gestrichelten Linien bedeuten, dass der Support zwischen zwei Gruppen niedrig ist. Durch die Mediator Sicht kann man die verschiedenen Item-Gruppen mit gleichem Mediator überprüfen lassen.

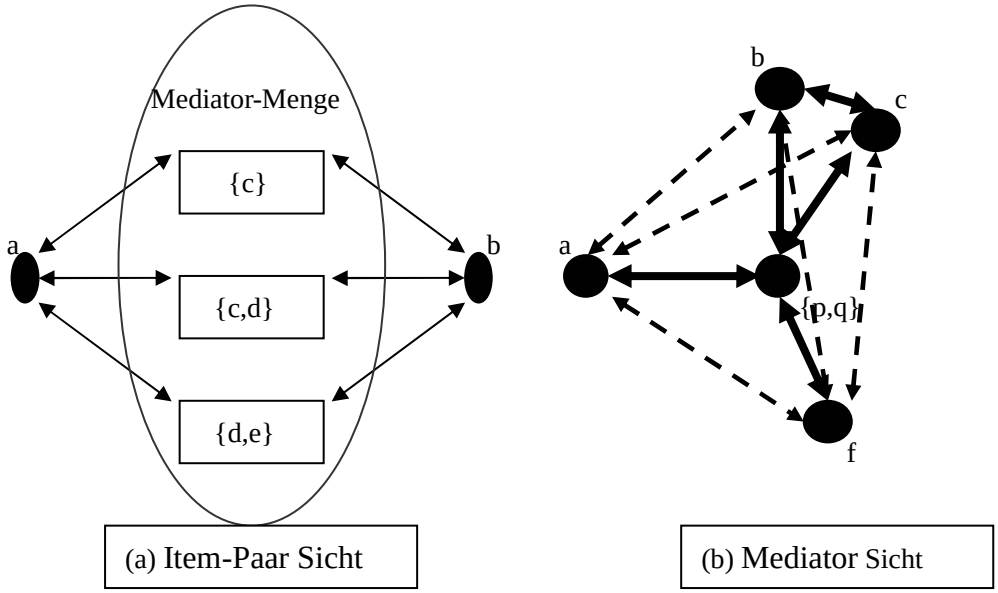


Bild 2-2-1 Zusammenbau von indirekten Beziehungen

2.2.1.5 Zusammenfassung

In diesem Kapitel wurde das Mining der Beziehungsmuster in Web-Usage-Daten dargestellt. Ein neues Muster (indirekte Beziehung) wird formuliert, und eine Mining-Technik zur Entdeckung von indirekten Beziehungen wird dargestellt. Die indirekte Beziehung kann zur Identifikation unterschiedlicher Gruppen von Web-Benutzern dienen, die ähnliche Pfade während ihrer Web-Besuche nutzen. Durch Einsatz dieser Technik werden die interessanten Item-Paare entdeckt, die sich negativ beziehen. Wenn die indirekten Beziehungen mit demselbem Mediator zusammenfasst werden, wird die Anzahl der Beziehungsmuster signifikant reduziert.

2.2.3 Wissensbasierte Wrapper-Induktion für intelligente Web-Informationsextraktion

2.2.3.1 Einführung

Informationsextraktion ist das Wiedererkennen und die Extraktion spezifischer Datenfragmente aus Dokumentensammlungen. Die Prozesse der Informationsextraktion hängen von einer Menge von Extraktionsregeln ab, die *Wrapper* heissen. Ein Wrapper ist eine Regel oder eine Prozedur, die die von einer spezifischen Quelle angebotenen Informationen versteht und sie in eine Regularität übersetzt. Solche Regularitäten können zur Extraktion besonderer Attributwerte oder Eigenschaften benutzt werden. Ein Wrapper ist auf die Informationsquelle einer einzigen Art spezialisiert.

Die Techniken der Wrapper-Generierung für Web-Informationsextraktion können in drei Kategorien klassifiziert werden: manuelle Wrapper-Generierung, heuristische

Wrapper-Induktion und wissensbasierte Wrapper-Induktion. Bei der Methode der manuellen Wrapper-Generierung sind die Extraktionsregeln von Menschen nach einer sorgfältigen Überprüfung von Beispiel-Web-Seiten festgelegt. Für jede unterschiedliche Informationsquelle muss eine neue Extraktionsregel beschrieben werden, ebenso wenn eine Informationsquelle erzeugt wird oder sich ihre Struktur verändert. Deswegen, obwohl sie eine hochpräzise Leistung besitzt, passt sie sich nicht einem intelligenten Web Information Management an, weil sie nicht skalierbar ist. Zur Vermeidung dieses Nachteils wurde die Wrapper-Induktion eingesetzt, die automatisch einen Wrapper durch Lernen aus Beispiel-Web-Seiten einer Informationsquelle bildet. Wrapper-Induktion kann durch Heuristiken oder Domänenkenntnis vorgenommen werden. Heuristische Wrapper-Induktion ist in den meisten traditionellen Systemen eingesetzt worden. Dieser Einsatz ist manchmal nicht effizient, da die Heuristiken meistens einfach und naiv sind.

Eine Domänenkenntnis beschreibt Terme, Konzepte und Beziehungen, die in einer spezifischen Anwendungsdomäne benutzt werden. Sie beseitigt die Schwachpunkte, die die lexikalisch orientierte Analyse mit sich bringt und spielt eine Kernrolle bei der Erkennung semantischer Fragmente eines Dokuments in einer gegebenen Domäne. Durch Definition und Anwendung der Domänenkenntnis versucht die wissensbasierte Wrapper-Induktion, das Problem der manuellen Wrapper-Generierung und der heuristischen Wrapper-Induktion zu lösen. Durch Benutzung der Domänenkenntnis bildet der Wrapper-Generator automatisch einen Wrapper.

2.2.3.2 XTROS wissensbasiertes Informationsextraktionssystem

XTROS ist eine Implementierung eines wissensbasierte Informationsextraktions-Systems. XTROS generiert automatisch einen Wrapper für jede Informationsquelle und extrahiert die spezifische Information durch Anwendung des generierten Wrappers auf die entsprechende Quelle. Durch Benutzung der Domänenkenntnis erkennt der Wrapper-Generierungsalgorithmus die Bedeutung logischer Linien eines Musterdokuments, um die am häufigsten auftretenden Muster aus der Sequenz der logischen Linien zu finden.

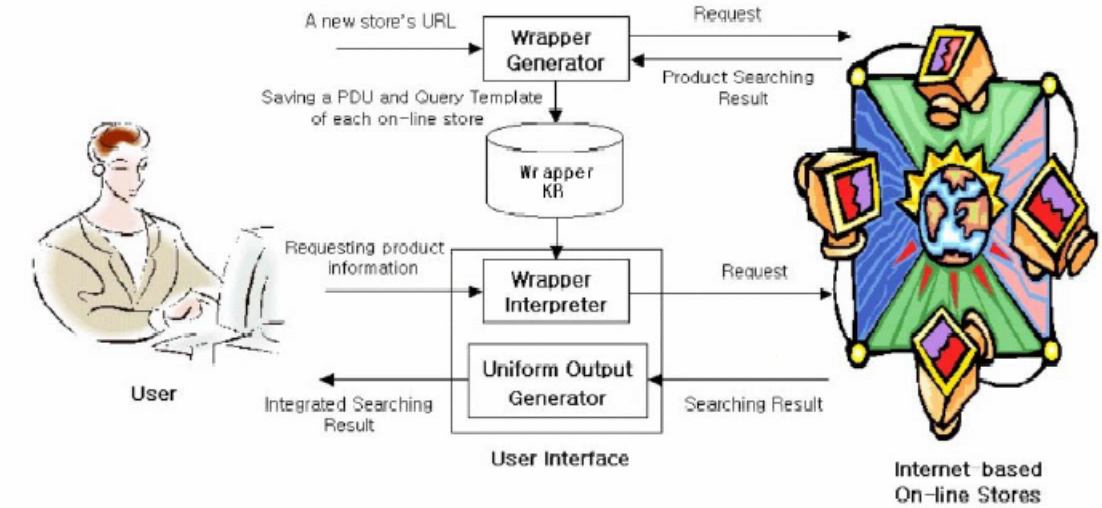


Bild 2-3-1 Übersicht des XTROS Systems

Schaubild 2-3 zeigt die Übersicht des XTROS Systems. Der Wrapper-Generator und der Wrapper-Interpreter sind zwei Hauptmodule von XTROS. Der Wrapper-Induktionszyklus ist im obigen Schaubild gezeigt. Der Wrapper-Interpretations-prozess ist in der unteren Abbildung dargestellt.

XTROS ist zur Behandlung semistrukturierter Dokumente entworfen. Ein semistrukturiertes Dokument beinhaltet strukturierte und unstrukturierte Komponenten [18]. Der unstrukturierte Teil beinhaltet Menüs, Headers oder Adressen, die meistens nichts mit Semantikinformation zu tun haben und während des Extraktionsprozesses vernachlässigt werden sollen. Der strukturierte Teil besteht aus der Information, die aus Datenbankanfragen gewonnen werden. In einem ausgezeichneten Dokument ist jede Portion von Daten, die extrahiert werden soll, durch ein *Label-Value*-Paar dargestellt, wobei das Label die Bedeutung des zugeordneten Wertes beschreibt. In der Domäne von Immobilien beispielsweise hat die Beschreibung jedes Hauses viele Label-Value-Paare, wie Preis: $3.195.000, BR:5 und Bäder:1. $, BR und Bäder sind Label, die die Dollarsumme, die Anzahl der Schlafzimmer und die Anzahl der Bäder notiert. Ein Label kann vor oder nach seinem Wert auftauchen.

In XTROS werden die Domänenkenntnis und die Wrapper durch XML-Dokumente repräsentiert. Die XML-Repräsentation verbessert die Modularität und Flexibilität, weil XML mehrere Arten zur Repräsentation der Extraktionsregeln bietet. Außerdem werden durch den Einsatz von XML-Parsern und -Interpretern einfache Implementierungen realisiert. Wegen der Interoperabilität von XML besitzt der XML-basierte Wrapper eine Portabilität und Mitbenutzbarkeit in verteilten Umgebungen.

In XTROS ist die Kenntnis einer Domäne innerhalb einer <Knowledge> ... </Knowledge> Struktur repräsentiert. Die <Knowledge> Struktur enthält die <Objects>...</Objects> Struktur, die Eigenschaften, die Objekte genannt und deren Wert extrahiert werden kann, auflistet. Ein XML-Konstruktor besteht aus zwei Elementen: <Ontology> und <Format>. <Ontology> listet die Terme auf, die zur Anerkennung der Existenz von einem Objekt benutzt werden. <Format> beschreibt die Datentypen der Objektwerte und die positionelle Beziehung zwischen den ontologischen Termen und den Werten. Domänenkenntnis wird durch Untersuchungen zahlreicher Testseiten aus der gleichen Domäne erhalten. Die <Ontology> listet alle mögliche Labels, die aus Testseiten gesammelt werden, als <Term> Elemente auf. Das <Format> eines Objekts spezifiziert mögliche Ordnungen zwischen den Labels und seinen Werten als <Form> Elemente. Durch Einfügung von neuen <Term> Elementen in der XML-Repräsentation kann die Domänenkenntnis leicht erweitert werden. Diese Flexibilität ermöglicht, dass die XML-basierte Repräsentation anpassungsfähig für unterschiedliche Domänen ist.

2.2.3.3 Wissensbasierte Wrapper-Generierung

Eine Kernfunktion des Wrapper-Generators ist das Erlernen des Web-Seiten-Formats aus verschiedenen Beschreibungen der erfolgreichen Suchergebnisse. Die Ausgabe der HTML-Seite einer Datenbanksuche (PG) besteht aus drei Teile: dem Kopf (K), der Liste der Item-Beschreibungen (L) und dem Fuß (F). Die PG kann als <K, L, F> beschrieben werden. K und F sind irrelevant und können vernachlässigt werden. L ist

eine Liste aus Item-Beschreibungen (ID) und kann als L = $< ID_1, ID_2 ,..., ID_n >$ geschrieben werden. Eine ID besteht aus mehreren Attributen und wird als $< A_1 , A_2 ,..., A_m >$ geschrieben. In XTROS gibt es kein separates Modul zur Entfernung von K und F. Während des Lernvorgangs des Wrapper findet er die Start- und Endposition von L und erkennt die Patterns einer Item-Beschreibung. Dieser Prozess hat drei Phasen: Konvertierung der HTML-Quelle in logische Linien, Bestimmung der Bedeutung der logischen Linien und Finden der am häufigsten auftretenden Muster.

2.2.3.3.1 Konvertierung der HTML-Quelle in logische Linien

In der ersten Phase wird die HTML-Quelle von der Suchergebnisseite in eine Sequenz logischer Linien zerlegt. Eine logische Linie ist ähnlich wie die Linie, die der Benutzer im Browser sieht. Der Lerner identifiziert sie durch Wahrnehmung der HTML-Element wie
, <p>, <li>, <td> und <tr>. Im Schaubild 2-4 wird ein Vergleich zwischen zwei HTML-Quellen gezeigt. Alle HTML-Element außer <a href =...> und <img src = ...> sind irrelevant und werden entfernt. Die erste HTML- Quelle ist original, die zweite HTML-Quelle ist aus Eliminierung redundanter Tags der ersten HTML-Quelle entstanden.

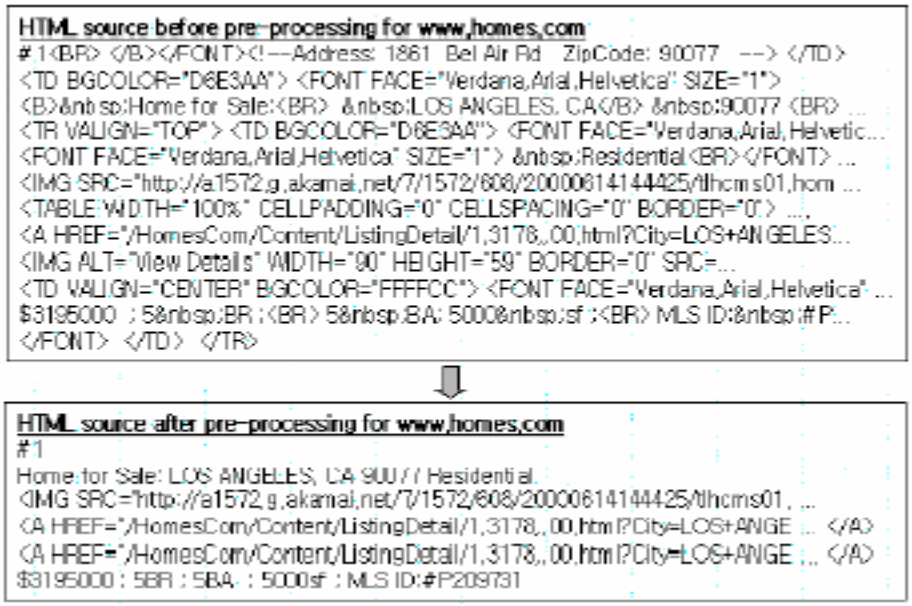


Bild 2-3-2 Beispiel der Konvertierung in logischen Linien

2.2.3.3.2 Bestimmung der Bedeutung logischer Linien

Die zweite Phase dient zur Bestimmung der Bedeutung von jeder logischen Linie, indem die Existenz jedes Objekts in der Domänenkenntnis überprüft wird. Eine gegebene logische Linie wird als ein Objekt anerkannt, wenn sie irgendeinen entsprechenden Term in ihrer <Ontology> Spezifikation enthält und sie zu irgendwelchen entsprechenden Formen in der <Form> Definition passt. Wenn eine logische Linie mehr als ein Objekt enthält, wird sie vor der Anwendung des Moduls in Subteile zerlegt. Nach Bestimmung der Bedeutung wird jede logische Linie durch vordefinierte Datenstrukturen repräsentiert. Diese Datenstruktur besteht aus fünf Teilen: Objekt, Linie, Kategorie, Typ und Format. Objekt beschreibt die Bedeutung der Linie. Linie beinhaltet die ursprüngliche Linie mit Kennzeichnungen für

Ontologie und Format. Kategorie ist die Katalognummer der entsprechenden Objekte. Typ ist der Datentyp des Objektwerts. Format definiert die positionelle Beziehung zwischen der Ontologie und dem Wert. Tabelle 2-3-1 zeigt in einem Beispiel die Katalognummer zum entsprechenden Objekt.

Katalognummer	Objekte
0	PRICE
1	BED
2	BATH
3	CITY
4	MLS
5	DETAIL
6	IMG
9	GENERAL TEXT

Tabelle 2-3-1 Beispiel der Katalognummern und Objekte

No.	Objekt	Linie	Katalog	Typ	Form
1	{(IMG)}	<A HREF=“ „> {(<IMG[{ALT=“.. „}]>)} {{IMG}}	6	IMGURL	[ONTOLOGY] IMGURL
2	{[PRICE]}	{[[{$}]3195000]}	0	DIGITS	[ONTOLOGY] DIGITS
3	{[BED]}	;{[5[{BR}]]} {{BED}}	1	DIGITS	DIGITS [ONTOLOGY]
4	{[BATH]}	;{[5[{BR}]]} {{BATH}}	2	DIGITS	DIGITS [ONTOLOGY]
5	{[MLS]}	; 5000sf ; {[[{MLS ID:#}]P209731]} {{MLS}}	4	DIGITS	[ONTOLOGY] DIGITS
6	{[DETAIL]}	{[<A HREF =“ „> [{View Property}]]} {{DETAIL}}	5	URL	URL [ONTOLOGY]

Tabelle 2-3-2 Aus logischen Linien resultierende Datenstruktur

In Tabelle 2-3-2 werden die aus den logischen Linien resultierenden Datenstrukturen gezeigt.

2.2.3.3.3 Finden der meisten benutzten Muster

Nach der Katalogisierungsphase wird die Seite durch eine Sequenz von Katalognummern dargestellt. Die dritte Phase findet die häufigsten Muster in dieser Sequenz. Ein Pseudokode dieses Suchalgorithmus ist nachfolgend gezeigt:

```
FindMostFreqPattern (Sequenz)
    KandPatterns ← Hole alle Kandidate Patterns in Sequenz;
    MFPgesamtNumAtt ← 0; // MFP: most frequente Pattern
    For jede Pattern in KandPatterns do
```

```
        freq ← hole die Frequenz von Patter in Sequenz;
        NumAttr ← hole die Nummer von Attribute in Pattern;
        gesamtNumAttr ← freq * numAttr;
        if ( gesamtNumAttr > MFPgesamtNumAtt ) then
            MFPgesamtNumAtt ← gesamtNumAttr;
            MostFreqPattern ← Pattern;
        End
    End
Return mostFreqPattern;
```

Diese Prozedur findet alle Kandidatenmuster aus der Katalogsequenz. Ein Kandidatenmuster ist ein beliebiger maximaler Substring der Sequenz mit mindesten drei Attributen und ohne duplizierte Attribute. Zum Beispiel sind bei einer gegebenen Sequenz 601260126015 die mögliche Kandidatenmuster 6012, 0126, 1260, 2601, 6015, 015. 60126 ist kein Kandidatenmuster, weil es ein dupliziertes Attribut in diesem Muster (6) gibt. 601 ist auch kein Muster, weil der String nicht maximal ist (6012 oder 6015). Der nächste Schritt ist die Bestimmung der Frequenz von einem Kandidatenmuster in der Sequenz (Fp). Sie zeigt, wieviele Male das Muster P in der Sequenz auftritt. Im obigen Beispiel ist F_{6012} = 2, F_{6015} =1. Die Anzahl der Attribute im Muster ist als Ap definiert. Die gesamte Anzahl der Attribute des Muster ist als Tp definiert und Tp = Fp * Ap. Durch den Vergleich in der For-Schleife wird letztendlich das am häufigsten auftretende Pattern mit den meisten Attributen in einer Sequenz bestimmt.

Tabelle 2-3-3 zeigt eine Statistik einer anderen Beispielsequenz von allen möglichen Kandidatenmustern mit ihren Fp-, Ap- und Tp-Werten. 601245 wird als Muster dieser Seite ausgewählt, weil es den maximalen Tp-Wert hat.

Kandidatenmuster (P)	Muster Frequenz (Fp)	Anzahl der Attribute im Muster (Ap)	Gesamte Anzahl der Attribute des Muster (Tp=Ap*Fp)
601245	5	6	30
012456	4	6	24
124560	4	6	24
245601	4	6	24
456012	4	6	24
560124	4	6	24
56012	5	5	25
60125	1	5	5
01256	1	5	5
12560	1	5	5
25601	1	5	5
01245	5	5	25
1245	5	4	20
245	5	3	15

Tabelle 2-3-3 Statistik der Frequenz der Kandidatenmuster

2.2.3.3.4 Konstruktion eines XML-basierten Wrappers

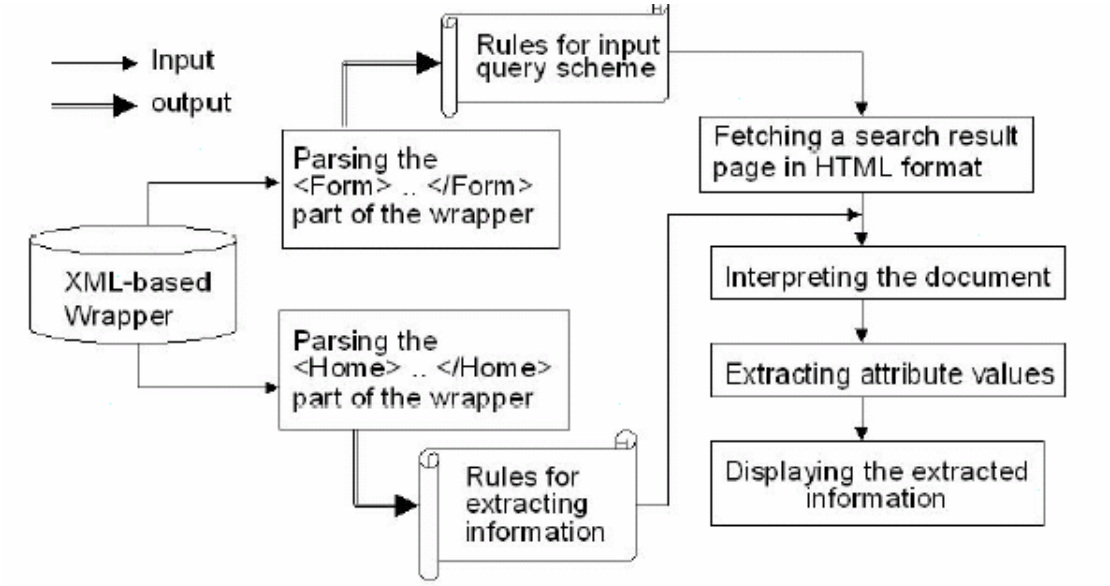
Ein XML-basierter Wrapper ist auf das am häufigsten auftretende Muster basiert. Ein Wrapper ist durch die <Wrapper> Struktur repräsentiert, die aus zwei Substrukturen besteht: <Form> und <Home>. Das <Form> Konstrukt beschreibt das Input-Query-Schema, um die Suchergebnisseite zu holen. Informationen in der <Form> Struktur enthalten die URL des CGI-Modells der Seite, die Anzahl der HTML Form Labels und so weiter. Das <Home> Konstrukt beschreibt das Muster, das vom Pattern-Suche-Modul geliefert wird. Jedes Konstrukt besteht aus <Ontology>, <Ident>, <Format>, und <Operation> Elementen. <Ontology> spezifiziert einen Term, der die Existenz eines Objekts bestimmt. <Ident> ist ein Delimiter, der Objekte voneinander abgrenzt. <Format> spezifiziert den Datentyp des zu extrahierenden Werts. <Operation> zeigt, ob ein ontologischer Term vor oder nach dem Objektwert auftaucht. Ein XML-basierter Wrapper für eine Immobilien-Homepage wird wie folgend geschrieben:

```
<Wrapper>
    <Form> ---omitted---</Form>
    <Home>
        <Img>
            <Ontology>ALT=“View Details“</Ontology>
            <Ident> SRC </Ident>
            <Format> URL </Format>
            <Operation> Ontology*Format</Operation>
        </Img>
        <Price>
            <Ontology>$ </Ontology>
            <Ident> NULL</Ident>
            <Format> Digts</Format>
            <Operation> Ontology*Format </Operation>
        </Price>
        <Bed>
            <Ontology>BR</Ontology>
            <Ident> NULL</Ident>
            <Format> Digts</Format>
            <Operation> Ontology*Format </Operation>
        </Bed>
        <Bath>
            <Ontology> BA</Ontology>
            <Ident>NULL </Ident>
            <Format>Digts </Format>
            <Operation> Ontology*Format </Operation>
        </Bath>
        <Mls>
            <Ontology>MLS ID:# </Ontology>
            <Ident>NULL </Ident>
            <Format> Digits</Format>
            <Operation> Ontology*Format </Operation>
        </Mls>
        <Detail>
```

```
            <Ontology>View Property </Ontology>
            <Ident>A HREF </Ident>
            <Format>URL </Format>
            <Operation>Ident* Ontology*Format </Operation>
        </Detail>
    </ Home>
</Wrapper>
```

2.2.3.3.5 Interpretierung der Wrapper

Die Extraktion der Information wird vom Interpreter von XTROS vollzogen. Die Informationen werden vom XML-Wrapper zerlegt, um Extraktionsregeln aufzubauen und um solche Regeln an den Suchergebnisseiten anzuwenden. Der Interpretierungsprozess ist in folgender Graphik dargestellt:



2.2.3.4 Zusammenfassung

Das intelligente Web-Informationsextraktionssystem XTROS repräsentiert die Domänenkenntnis und Wrapper in XML-Dokumenten. Die Benutzerschnittstelle, der Wrapper-Generator und der Wrapper-Interpreter von XTROS sind in Java implementiert. Im Vergleich zu manueller Wrapper-Generierung besitzt XTROS eine bessere Skalierbarkeit und ist besser geeignet für heterogene Umgebungen. Im Vergleich zur heuristischen Wrapper-Induktion ist XTROS präziser und flexibler. Eine vorläufige Beschränkung von XTROS ist es, dass XTROS nur ausgezeichnete Dokumente verarbeitet und nicht mit non-Label-Dokumenten wie Table-Type-Beschreibungen (in Table-Type-Beschreibungen werden alle Informationen/Daten in Tabelleform geschrieben) funktioniert. Um dieses Problem zu lösen, werden im Augenblick viele Forschungen betrieben.

3 Web Farming

3.1 Übersicht des Web-Farming-Systems

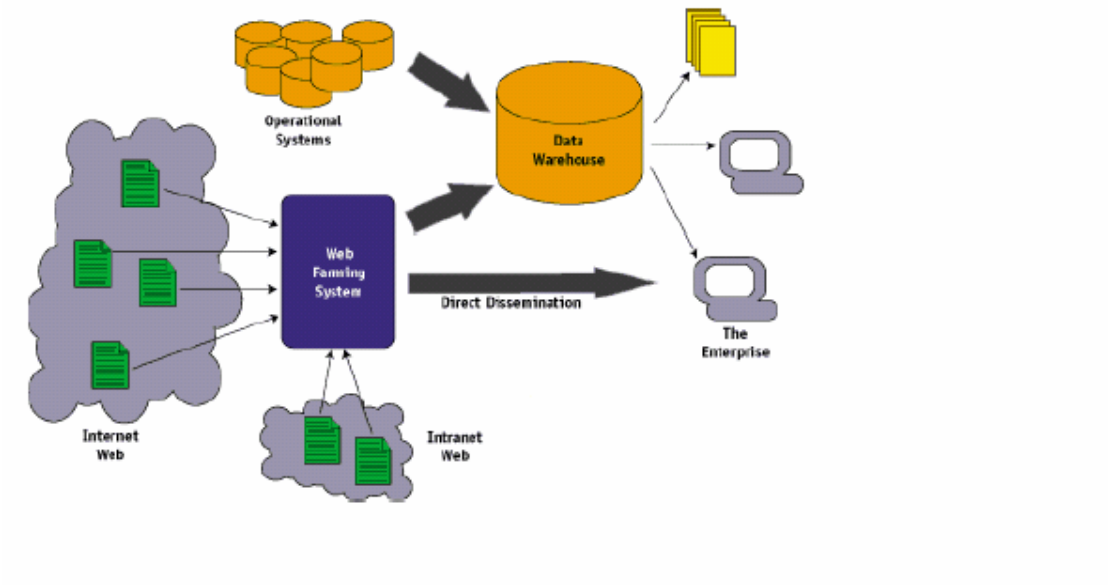


Bild 3-1 Web-Farming-System

Bild 3-1 zeigt, dass das Data Warehouse eine zentrale Position im Informationsfluss des Web-Faming-Systems besitzt. Das Web-Faming-System liefert die verfeinerten Informationen, die aus dem Web gefunden und gesammelt wurden, an ein Data Warehouse oder direkt an das Unternehmen.

Die primäre Ressource des Inhalts für Web-Farming-Systeme ist das *World Wild Web*. Die meisten aus dem Web entdeckten Informationen können nicht direkt vom Data Warehouse des Unternehmens aufgenommen werden. Sie sind unstrukturierte Hypertexte oder unverfeinerte tabellarische Werte. Vor Einlagerung der Informationen in das Data Warehouse muss die Verfeinerung der entdeckten Web-Informationen durchgeführt werden.

3.2 Verfeinerung der Web-Informationen

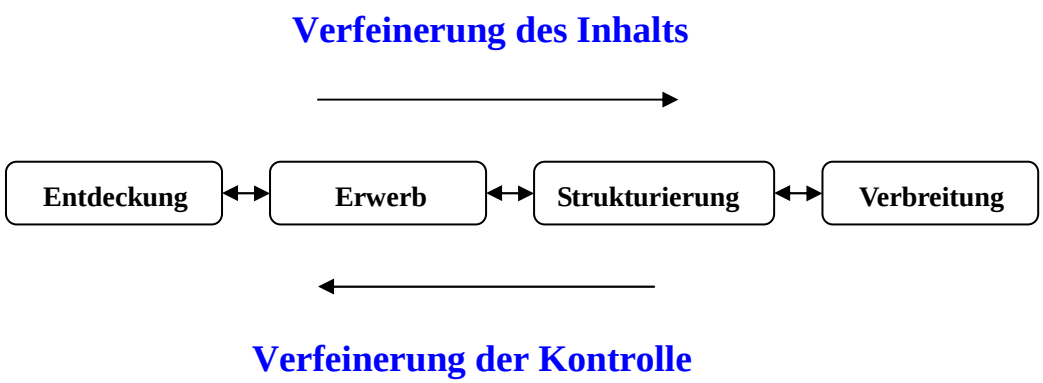


Bild 3-2 Prozess der Verfeinerung der Web-Information

Die Kernaufgabe des Web Farming ist Verfeinerung von Web-Inhalt, damit die Web-Informationen dem Geschäftsprozess zur Verfügung gestellt werden können. Bild 3-2 zeigt den Prozess der Verfeinerung der Informationen, der aus vier Einheiten (Entdeckung, Erwerbung, Strukturierung und Verbreitung) besteht. Inhaltsfluss zwischen diesen Einheiten läuft in beide Richtungen. Der Linke-zu-Rechte-Fluss verfeinert den Informationsinhalt und der Rechte-zu-Linke-Fluss verfeinert die Kontrolle über den Prozess.

- Entdeckung (Discovery) ist die Suche in verfügbaren Web-Ressourcen, um die Information bezüglich eines spezifischen Themas zu finden. Dies beginnt bei Suchmaschinen wie Google oder Verzeichnissen wie Yahoo, geht aber darüber hinaus. Das Ziel der Entdeckung ist Lokalisierung von Individuen und Organisationen, die relevante Informationen für das Unternehmen bereitstellen. Weil im Web kontinuierlich neue Ressourcen auftreten, muss die Entdeckung ein kontinuierlicher Prozess sein.
- Der Erwerb (Acquisition) ist die Sammlung und Erhaltung von identifiziertem Inhalt. Das Ziel des Erwerbs ist Erhaltung von historischen Informationskontexten, damit der Inhalt im Kontext seiner Vergangenheit analysiert werden kann. Eine Kernanforderung des Erwerbs ist der effiziente Einsatz menschlicher Beurteilung in der Validierung des Inhalts.
- Strukturierung (Structuring) ist die Analyse und Transformation des Inhalts in nützliche Form und Struktur. Die Formen können Websites, Dokumente oder Datenbanktabellen sein. Wenn die geladenen Daten in Data Warehouses speichert werden, müssen die unstrukturierten textuellen Daten aus dem Web in tabellarische Daten verfeinert werden, damit die kritische Business Analyse unterstützt wird.
- Verbreitung (Dessemination) ist Verpackung und Lieferung der Information zum entsprechenden Verbraucher. Ein effektives Web-Farming-System muss viele Verbreitungsmechanismen unterstützen, zum Beispiel vordefinierte Abläufe und Ad-Hoc-Abfragen. Wichtig ist hierbei, dass die Information unabhängig von der Form der Verbreitung gespeichert wird, damit sie wieder verwendet werden kann.

3.3 Vier-Stufen-Methodik

Um ein erfolgreiches Web-Farming-System zu implementieren und das Risiko einer erfolglosen Web-Farming-Systemimplementierung zu reduzieren, wird eine Vier-Stufen-Methodik in den meisten Implementierungen eingesetzt. Wie die Vier-Stufen-Methodik funktioniert, wird im folgenden Abschnitt dargestellt.

Stufe 1-Getting Startet. In dieser Stufe werden die Geschäftsfälle des Web Farming festgelegt, die auf den Aufgaben und den Markt der Unternehmung basieren. Die Critical External Factors (CEF) werden dokumentiert, der Entdeckungsplan wird formuliert, die Inhaltsverwerter werden identifiziert, die anfänglichen Informationen werden erworben und der Geschäftsfall wird erstellt. Es ist sehr wichtig, dass die externen Faktoren deutlich beschrieben werden, die positiven oder negativen Einfluss auf den Geschäftsfall haben können.

Stufe 2-Getting Serious. Die Web-Farming-Aktivitäten werden innerhalb der Geschäftsorganisation eingebettet, und die Infrastruktur im Datenzentrum wird für

Produktionsoperationen aufgebaut. Die CEF-Liste wird verfeinert, und der historische Kontext wird festgehalten.

Stufe 3-Getting Smart. Die Entdeckungs- und Strukturierungstechniken der Information werden verwertet und die Pipelines zu den Inhaltsverwertern werden aufgebaut. Die Auswahl- und Extraktionsfilter werden implementiert. Die Extraktionsfilter sind die Suchstrategien für das Auffinden relevanter Daten und die Algorithmen für Extraktion der Daten aus Web-Ressourcen.

Stufe 4-Getting Tough. Die Informationen für Data Warehouses werden nach Geschäftsaufgaben strukturiert. In dieser finalen Stufe wird die Zuverlässigkeit der Informationen überprüft und die korrekten Entities werden in Data Warehouses abgebildet.

3.3 Zusammenfassung

Web Farming bezeichnet die systematische Verfeinerung von web-basierter Information für Geschäftszwecke, die den Schwerpunkt der Arbeit auf das Nachbearbeiten der Information legen und in vier Phasen (Entdeckung, Erwerbung, Strukturierung und Verbreitung) eingeteilt ist.

Ein Web-Farming-System muss kontinuierlich und systematisch der richtigen Person zur richtigen Zeit die relevante Information liefern. Es beobachtet die Außenwelt und nimmt die wichtigen Änderungen in der Geschäftsumgebung wahr. Um ein Web-Farming-System aufzubauen, wird eine Vier-Stufe-Methodik eingesetzt, die schrittweise detailliert, wie man ein Web-Farming-System erfolgreich implementieren kann.

Literatur

- [1] R.Kosala, H. Blockeel: Web Mining Research: A Survey, SIGKDD Exploration.2, 1-15(2000)
- [2] O. Etzioi, The World Wide Web: quagmire or gold mine? Communications of the ACM, 39, 65-68(1996)
- [3] S.K Madria, S.S Bhowmick, W.K.Ng, E.-P. Lim: Research issus in web data mining, Proc. *Data Warehousing and Knowledge Discovery, Ist International Conference*(1999) pp. 303-312
- [4] A.-H. Tan: Text Mining: the state of the art and the challenges, *Proc. PAKKD'99 Wprkshop on Knowledge Discovery from advanced Database*(1999) pp. 65-70
- [5] Y.Yang, J.O. Pedsersen: A comparative study on feature selection in text categorization, *Proc. The 14th International Conference on Machine Learing*(1997) pp.412-420
- [6] M.Balabanovic: An adaptive web page recommendation service, *Proc. The 1st International Conference on Autonomous Agents*(1997) pp. 378-385
- [7] B. Liu, Y.M Ma, P.S. Yu: Discovering unexpected information from your competiors'web sites, *proc KDD'01(*2001) pp.144-153
- [8] G. Arocean, A. Mendelzon: webQQL: restructuring documents, databases and webs. *Proc. IEEE International Conference on Data Engineering*(1998) pp. 24-23
- [9] J. Srivastava, R. Cooley, M. Deshpande, P.-N. Tan: Web usage mining: discovery and applications of usage patterns from web data. SIGKDD Exploration, 1, 12-23(2000)
- [10] I. Borges, M.Levene: Data mining of user navigation patterns, *Proc. WEBKDD'99: Workshop on Web Usage Analysis and User Profiling*(1999) pp. 92-111
- [11] Z:-X. Hang, J.Ng, D.W. Cheng. M.K.Ng, W.-K. Ching: A cube model and cluster analysis for web access sessions, LNAI 2356(Springer, 2002)
- [12] R. Cooley, B. Mobasher, J. Srivastava: Web mining: information and pattern discovery on the World-Wild Web, *Proc. the 9th IEEE Interantional Conference on Tools with Artificial Intelligence*(1997) pp. 558-567
- [13] R. Agrawal, T. Imielinski, A. Swami: Database mining: a performance perspective. *IEEE Trans. Know. Data Eng.*, 5, 914-925(1993)
- [14] P.N. Tan, V. Kumar, J. Srivastava: Selecting the right interestingness measure for association patterns. In: D.Hand, D. Keim, R. Ng(eds), Proc. *8th Int. Conf. On Knowledge Discovery and Data Mining*, Edmonton, 23-26 July 2002(ACM Press) pp. 32-41
- [15] S. Brin, R. motwani, C. Silverstein: Beyond market baskets: generalizing association rules to correlations . In: J. Peckham(ed), *Proc. ACM SIGMODInt. Conf. On Management of Data*, Tucson, 13-15 May 1997(ACM Press, 1999) pp. 254-276
- [16] R. Agrawal, R. Srikant: Fast algorithms for mining association rules. In: J. B. Bocca, M. Jarke, C. Zaniolo(eds), *Proc. 20th VLDB Conference*, Santiago de

Chile, 12-15 September 1994(Morgan Kaufmann, 1994) pp. 487-499

- [17] R. Srikant, R. Agrawal: Mining sequncial patterns: generalitions and performance improvements. In: P. M. G. Apers, M. Bouzeghoub, G.. Gardarin(eds.), *Proc. 5th Int'l Conf. On Extending Database Technology(EDBT)*, Avignon, 25-29 March 1996, pp. 3-17
- [18] A. Blum, T. Mitchell: Combing labeled and unlabeled data with co-training. IN: *Proc. 11th Conf. On Computational Learning Theory*, Madison, WI, 24-26 July 1998(ACM Press 1998), pp. 92-100