Instanz-basiertes Matching mit COMA++

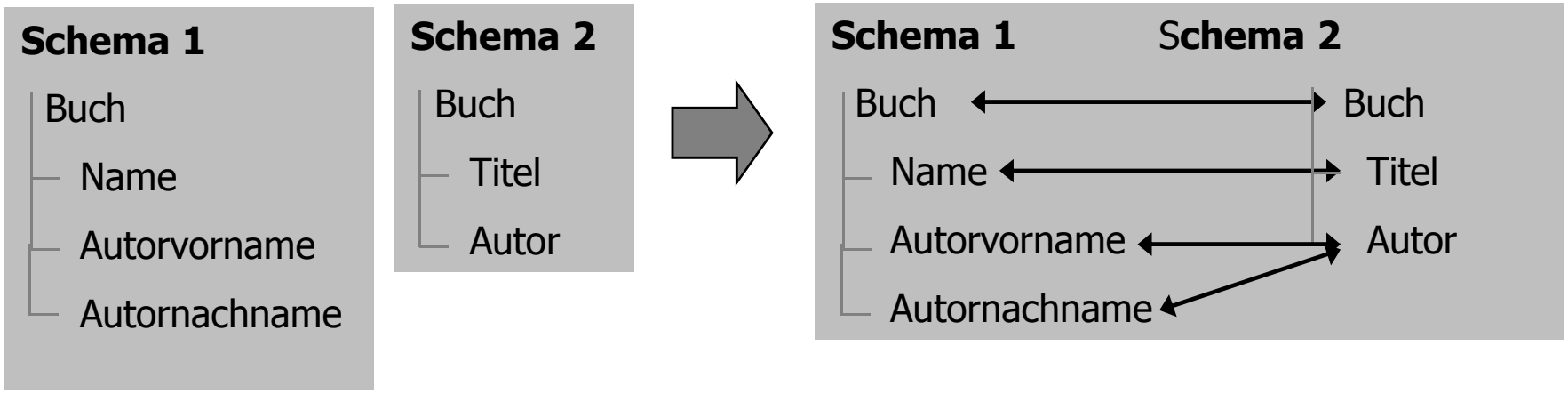
Sabine Maßmann

Zingst, 2.Juli 2008

UNIVERSITÄT LEIPZIG

Einleitung

- Situation: Austausch von Daten zwischen verschiedenen Datenquellen mit unterschiedlichen Schemata in z.B. Forschung und B2B
- Schema Matching:
  - Identifizierung von Korrespondenzen zwischen einem Ausgangsschema und einem Zielschema
  - Erster Schritt zur Datenintegration und –transformation



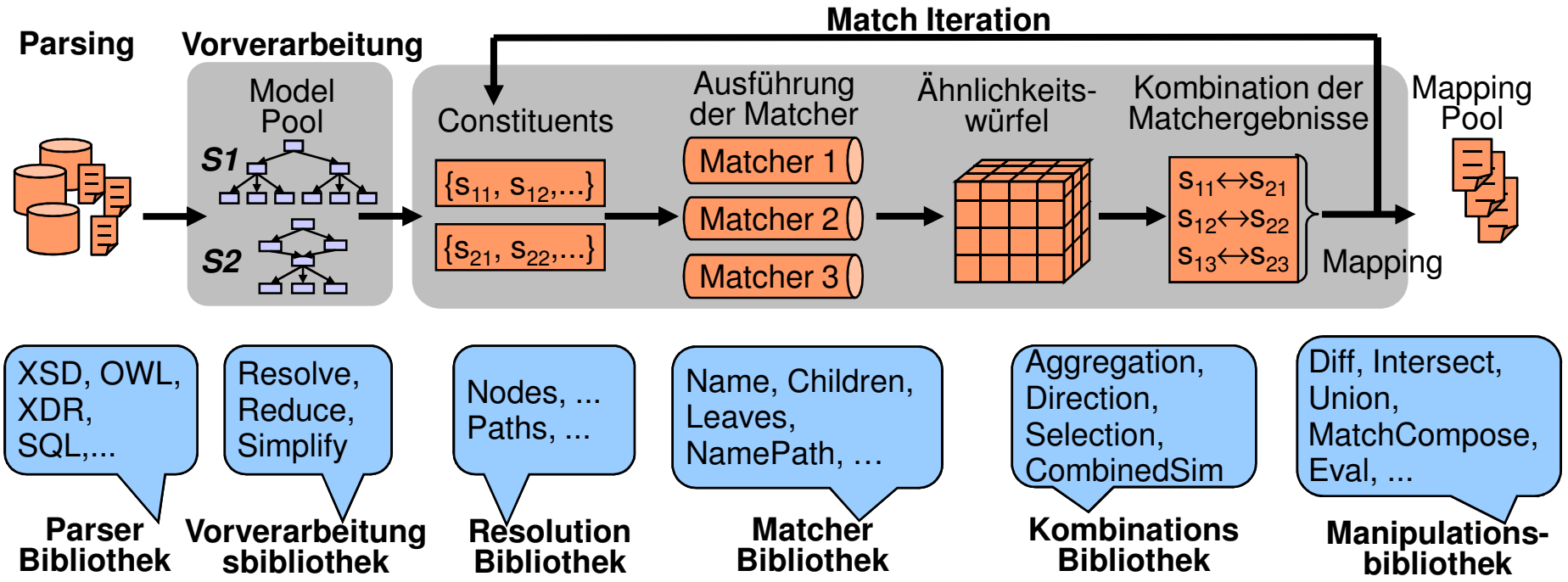
Übersicht

- Einleitung
- Matching mit COMA++
- Erweiterung zum Instanz-basierten Matching
- Bisherige Evaluation:
  - OAEI
  - Webverzeichnisse
- Produktkataloge
  - Aktueller Stand
  - Nächste Schritte

COMA / COMA++

- COMA *(VLDB 2002 –Do und Rahm)*
  - Flexible Kombination von Matchalgorithmen (composite approach)
  - Unterstützung von relationalen Schemata
  - Wiederverwendung von vorherigen Matchergebnissen
  - Umfassende Evaluation
- COMA++ *(SIGMOD 2005 – Aumüller, Do, Maßmann und Rahm)*
  - Generisches Datenmodell
  - GUI
  - Zusätzliche Unterstützung von XSD und OWL
  - Viele vordefinierte Matcher und flexible Konstruktion von neuen bzw. Änderung von vordefinierten Matchern
  - Strategien zum Umgang von großen Schemata und zur Wiederverwendung von bereits erstellten Mappings
  - Auch hier: umfassende Evaluation

Matchprozess bei COMA++



Motivation

- Herausforderungen beim (semi-) automatischen Matching
  - Heterogenitätsprobleme: terminologisch und konzeptuell
  - Redundanz
- Verwendung von Metadaten-basiertem Matching ist erfolgreich, hat aber Probleme im Umgang mit unverständlichen Namen, unbekannten Synonymen und verschiedenen Sprachen

  → **als Ergänzung** *(Erweiterung von COMA++)* **:**
  Verwendung von Instanzdaten für das Matching

Erweiterung für Instanz-basiertes Matching

- Import der Instanzdaten
- Erweiterte Schemarepräsentation: jedes Element kann zusätzlich zu bisherigen Daten wie Namen, Datentyp, etc. Instanzdaten beinhalten
- Import-Parser
  - Umgang mit unterschiedlichen Instanzquellen
  - Instanzdaten werden nach dem Parsen der generischen Schemarepräsentation zugewiesen

a) **Relational** → **Interner Schemagraph**

Person

firstName	**lastName**
Hong Hai	Do
Ulrike	Greiner

Name:**Person** *Type*: / *Instances*: /

- *Name*: **firstName** *Type*: string *Instances*: {Hong Hai, Ulrike}
- *Name*: **lastName** *Type*: string *Instances*: {Do, Greiner}

b) **XML** → **Interner Schemagraph**

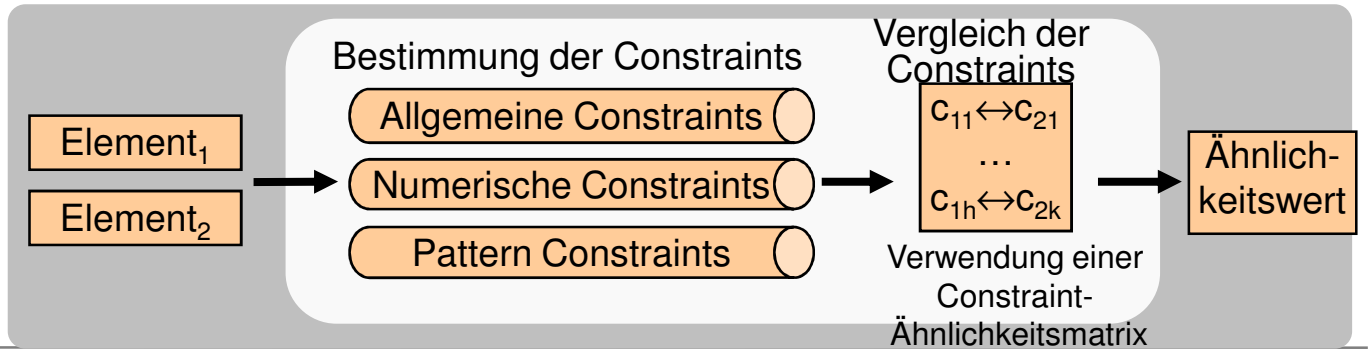
```
<Person>
  <firstName>Hong Hai</firstName>
  <lastName>Do</lastName>
</Person>
<Person>
  <firstName>Ulrike</firstName>
  <lastName>Greiner</lastName>
</Person>
```

Name:**Person** *Type*: / *Instances*: /

- *Name*: **firstName** *Type*: string *Instances*: {Hong Hai, Ulrike}
- *Name*: **lastName** *Type*: string *Instances*: {Do, Greiner}

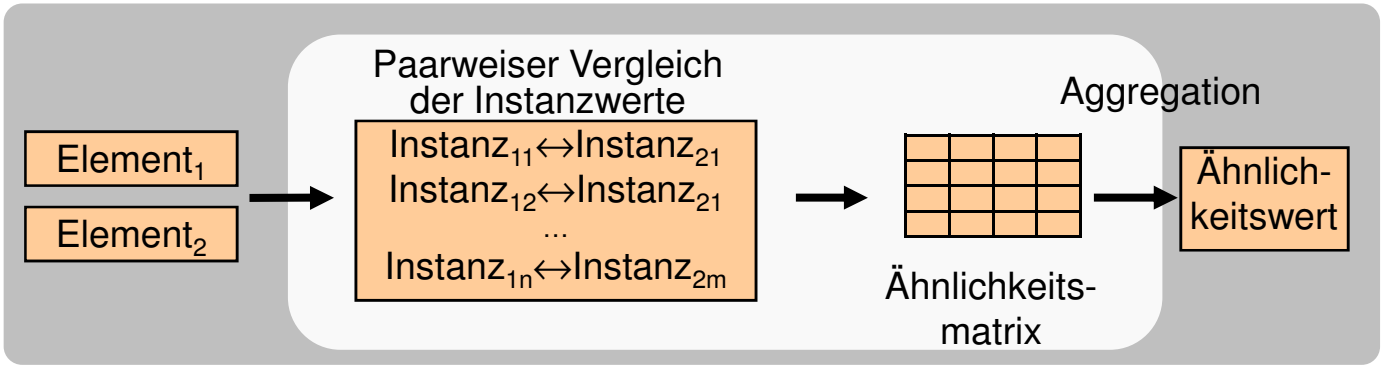
Constraint-basiertes Matching

- Constraints
  - Beschreibung von Charakteristiken oder Mustern
  - Zuweisung zu Schemaelementen
- Constraint-Gruppen:
  - **Allgemeine Constraints**: können für jede Instanz bestimmt werden
    Beispiel: durchschnittliche Länge und verwendete Zeichen (Buchstaben, Zahlen, Sonderzeichen)
  - **Numerische Constraints**: Bestimmung, ob eine Instanz eine Zahl ist und welche Art
    Beispiel : positiv oder negativ, Integer oder Fließkommazahl
  - **Pattern Constraints**: Untersuchung, ob alle Instanzen eines Elementes einem bestimmten Muster entsprechen
    Beispiel : Email und URL
- Ähnlich dem Matching von Datentypen – mit dem Wissen auf Instanzdaten basierend



Content-basiertes Matching

- Zur Bestimmung der Ähnlichkeit: Matching der Instanzwerte selber
- Ermittelt die Ähnlichkeit zweier Elemente durch:
  1. Ausführung eines paarweisen Vergleichs der Instanzwerte unter der Verwendung einer Ähnlichkeitsfunktion
     Ergebnis: eine Ähnlichkeitsmatrix in der jeweils eine Dimension, die Instanzen eines Elementes repräsentiert
  2. Vereinigung der Matrix zu einem Wert (Aggregation)
     Ergebnis : Ähnlichkeit der Instanzmengen und somit der Elemente
- COMA++ unterstützt (neben Gleichheit) viele Ähnlichkeitsfunktionen, die auf Zeichenketten basieren
  Beispiel: Trigram, Edit Distance, Soundex



Pro & Kontra

Constraint-basiertes Matching

- *Pro*: bei großen Instanzmengen nur geringer Aufwand
- *Kontra*: nicht geeignet für Fälle, die aus vielen Elementen mit "gemischten" Zeichenketten bestehen

Content-basiertes Matching

- *Pro*: Geeignet für Fälle, in denen Element-Instanzen gleich oder sehr ähnlich sind
  Beispiel: Straßennamen→"Arthur-Straße" vs. "Arthur-Str."
- *Kontra*: hoher Aufwand → n x m Vergleiche

Evaluation 1: OAEI-Benchmark

- OAEI Contest 2006 (http://oaei.ontologymatching.org) beinhaltet einen Benchmark
  - 51 Matchaufgaben
  - Eine Referenzontologie, die mit modifizierten Versionen gematcht werden muss
  - Unterschiedliche Modifikationen: Weggelassene Kommentare, flache Hierarchien, Namen wurden durch Zufallszeichenketten ersetzt,...
  - Für alle Aufgaben muss dieselbe Strategie und Konfiguration verwendet werden
- Beschränkung auf Ontologien mit mindestens einer Instanz
  → 39 Matchaufgaben mit 2966 Korrespondenzen in ihren Referenzalignments
- Evaluierung der Instanz-basierten Matcher alleine und mit Anwendung einer Propagation (nicht hier diskutiert)

1088 von 2966 Korrespondenzen sind zwischen 2 Elementen mit Instanzen → maximal möglicher Recall für Instanz-basiertes Matching **0.367**

Algorithmus	Precision	Recall	F-Measure
Content	0.997	0.353	0.521
Constraint	0.424	0.075	0.128

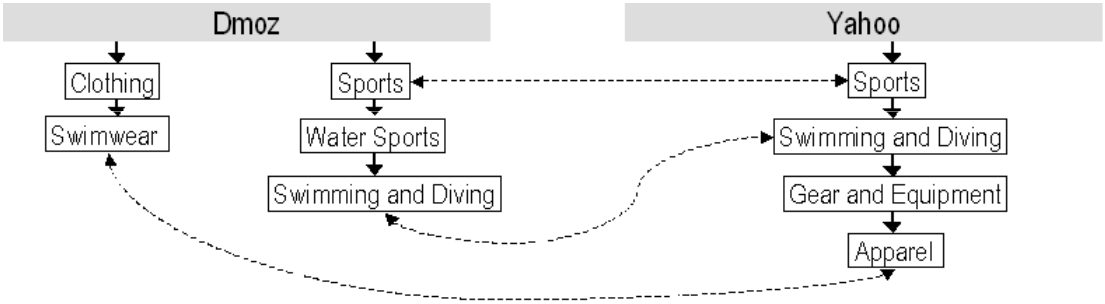
Viele Instanzen sind Zeichenketten und daher lassen sich mit Constraints nur wenige Instanz-Korrespondenzen identifizieren

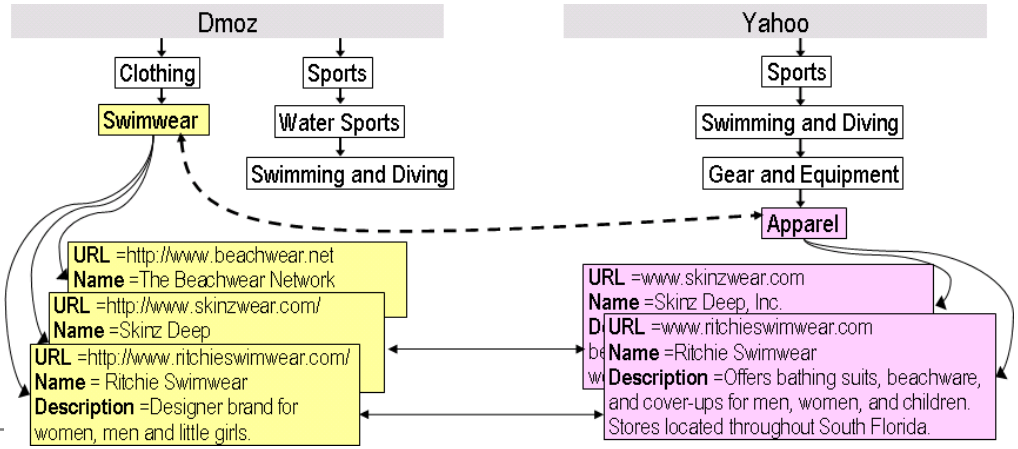
Evaluation 2: Webverzeichnisse

- Webverzeichnisse, z.B. Dmoz
  - Semantische Kategorisierung von Websites
  - Verwendung: Auffinden relevanter Websites zu einem bestimmten Interessensgebiet

Motivation zum Matching von Webverzeichnissen

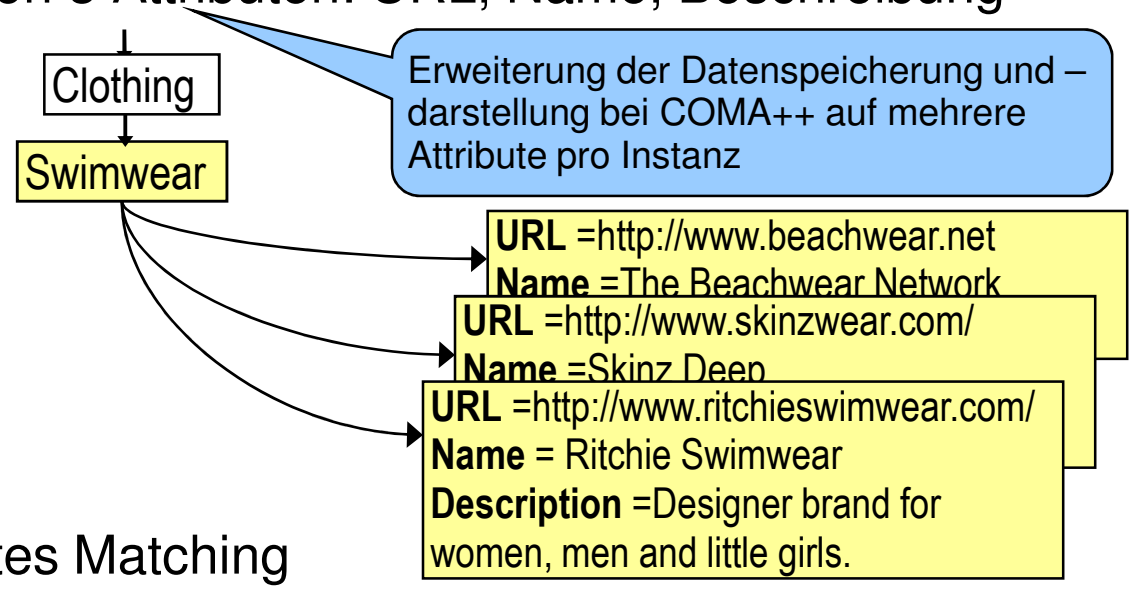
- Viele verschiedene Webverzeichnisse, e.g. Dmoz and Yahoo
- Äquivalenzmappings können genutzt werden zur:
  - Informationsintegration der verschiedenen Verzeichnisse
  - Verbesserung von Anfrageergebnissen
  - Generierung von Website-Empfehlungen





Instanzen der Webverzeichnisse

- Instanzen sind Blättern und inneren Kategorien zugeordnet
- Verwendung von 3 Attributen: URL, Name, Beschreibung (Description)



- Instanz-basiertes Matching
  - 3 Instanzmengen: URL, Name, Beschreibung
  - Namen- und Beschreibungs-basiertes Matching :
    - Namen von Websites sind viel kürzer als Beschreibungen und beinhalten zum Großteil zusammengesetzte Worte und Personennamen
    - Vorverarbeitung, z.B. Entfernung einzelner Zeichen und Sonderzeichen
    - Ähnlichkeitsberechnung unter Verwendung von TFIDF (Gewichtung)

Abspeicherung der Instanzen in Repository

- Jedes Schema/Ontologie bekommt eigene Instanz-Tabelle
  (zuvor nur temporär und nur eine Instanzmenge pro Element)

a)

```
<foaf:Person rdf:id="#a712561038">
   <rdfs:label>Marc Ehrig</rdfs:label>
   <foaf:name rdf:datatype="&xsd;string">Marc
Ehrig</foaf:name>
   <foaf:firstName
rdf:datatype="&xsd;string">Marc</foaf:firstName>
   <lastName rdf:datatype="&xsd;string">Ehrig</lastName>
</foaf:Person>
```

b)

```
<TFreizeit_Basteln_Window_Color
rdf:ID="http://www.bastelstore.de/">
   <rdfs:label>Bastelstore</rdfs:label>
   <rdfs:comment>Neben dem unfangreichen Window
Color Sortiment werden Farben und Lacke fuer die
Serviettentechnik angeboten.</rdfs:comment>
</Freizeit_Basteln_Window_Color>
```

zu a)

id	connect	elementid	Instance_id	attribute	value
11	a712561038	798	11	id	a712561038
12	a712561038	798	12	label	Marc Ehrig
13	a712561038	811	-1	[NULL]	Ehrig

zu b)

id	connect	elementid	Instance_id	attribute	value
20	http://www.bastelstore.de/	412	20	id	http://www.bastelstore.de/
21	http://www.bastelstore.de/	412	21	comment	Neben dem unfangreichen Window Color Sortiment werden Farben und Lacke fuer die Serviettentechnik angeboten.
22	http://www.bastelstore.de/	412	21	label	Bastelstore

Instanz-basiertes Matching mit Verwendung der URLs

- Motivation: URLs verschiedener Verzeichnisse überlappen sich
  → Grad der Überlappung wird zur Ähnlichkeitsberechnung genutzt
- URL-Vergleich basiert auf Gleichheit:
  - Grund: selbst kleine Unterschiede in URLs führen zu total verschiedenen Bedeutungen, z.B. Tuch vs. Buch vs. Busch
  - Vorverarbeitung wird ausgeführt, um mit unterschiedlichen Notierungen von URLs umzugehen

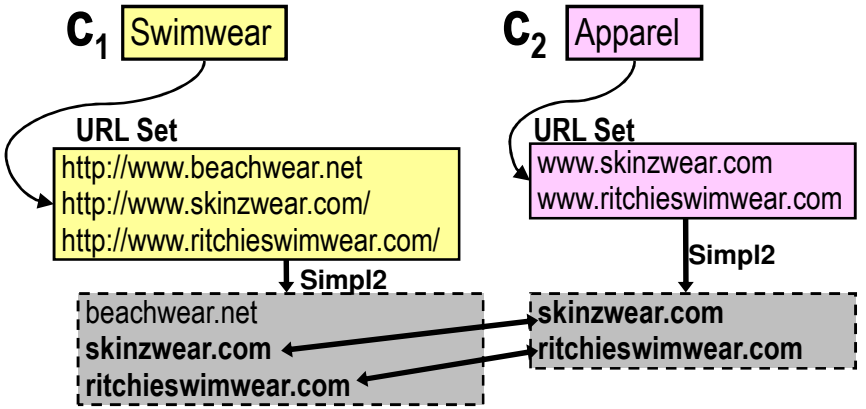
5 mögliche Vorverarbeitungsschritte:

- **Original**: komplette URLs, wie sie im Verzeichnis auftauchen
  http://www.Test.com/Shop/
- **Simpl1**: "/" am Ende und Parameter nach "?" entfernen, Kleinschreibung
  http://www.test.com/shop
- **Simpl2**: *Simpl1* + am Anfang "http://" and "www." entfernen
  test.com/shop
- **Simpl3**: *Simpl2* + alles nach dem ersten "/" entfernen
  test.com
- **Simpl4**: *Simpl3* + Domäne entfernen (alles nach dem letzten ".")
  test

Ähnlichkeitsmaße für das URL-basierte Matching

Ähnlichkeitswerte $\in [0\ldots1], \forall c_1 \in C_{D1}, c_2 \in C_{D2}$

- $Sim_{Base-k}(c_1,c_2) = \begin{cases} 1, if \left|I_{c1} \cap I_{c2}\right| \geq k \\ 0, if \left|I_{c1} \cap I_{c2}\right| < k \end{cases}$
- $Sim_{Min}(c_1,c_2) = \dfrac{\left|I_{c1} \cap I_{c2}\right|}{\min(\left|I_{c1}\right|,\left|I_{c2}\right|)}$
- $Sim_{Dice}(c_1,c_2) = \dfrac{2 \cdot \left|I_{c1} \cap I_{c2}\right|}{\left|I_{c1}\right| + \left|I_{c2}\right|}$
- $Sim_{Max}(c_1,c_2) = \dfrac{\left|I_{c1} \cap I_{c2}\right|}{\max(\left|I_{c1}\right|,\left|I_{c2}\right|)}$



$$Sim_{Base-1}(c_1,c_2) = 1$$
$$Sim_{Min}(c_1,c_2) = \frac{2}{2} = 1$$
$$Sim_{Dice}(c_1,c_2) = \frac{2 \cdot 2}{3+2} = 0.8$$
$$Sim_{Max}(c_1,c_2) = \frac{2}{3} = 0.67$$

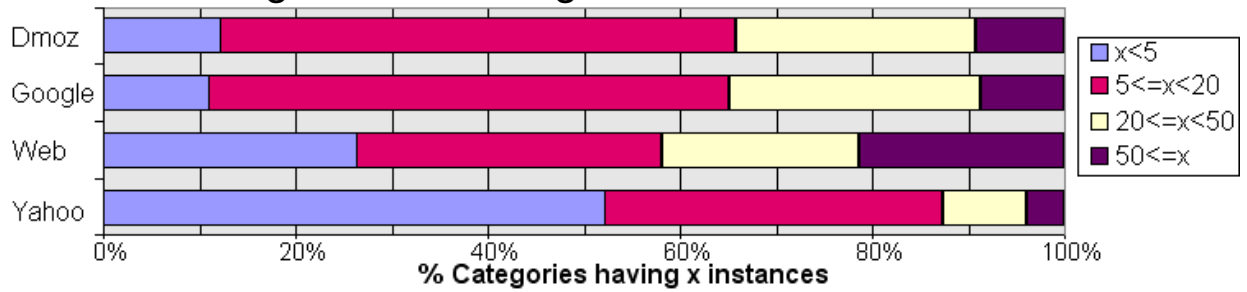
$$Sim_{Base-1}(c_1,c_2) \geq Sim_{Min}(c_1,c_2) \geq Sim_{Dice}(c_1,c_2) \geq Sim_{Max}(c_1,c_2)$$

Datensätze

- Vier Webverzeichnisse, Beschränkung auf Onlineshops

	Dmoz	Google	Web	Yahoo
#Kategorien	746	728	418	3,234
# Kategorien mit direkt assoziierten Instanzen	738	720	380	3,143
#Instanzen	15,304	15,082	13,673	34,949
# Direkte Assoz. pro Kat.	21	21	36	11
Durschnittliche Länge URL / Name / Beschr.	28 / 21 / 119	29 / 20 /119	28 / 26 / 92	28 / 11/ 70

- Instanzverteilung über die Kategorien



- Sechs Matchaufgaben → Sechs Referenzmappings (von Hand erstellt)

	Dmoz ↔ Google	Dmoz ↔ Web	Dmoz ↔ Yahoo	Google ↔ Web	Google ↔ Yahoo	Web ↔ Yahoo
# Korresp.	729	218	436	211	416	235
Abgedeckte Kategorien	98% ↔ 100%	29% ↔ 50%	55% ↔ 13%	29% ↔ 48%	55% ↔ 12%	52% ↔ 7%

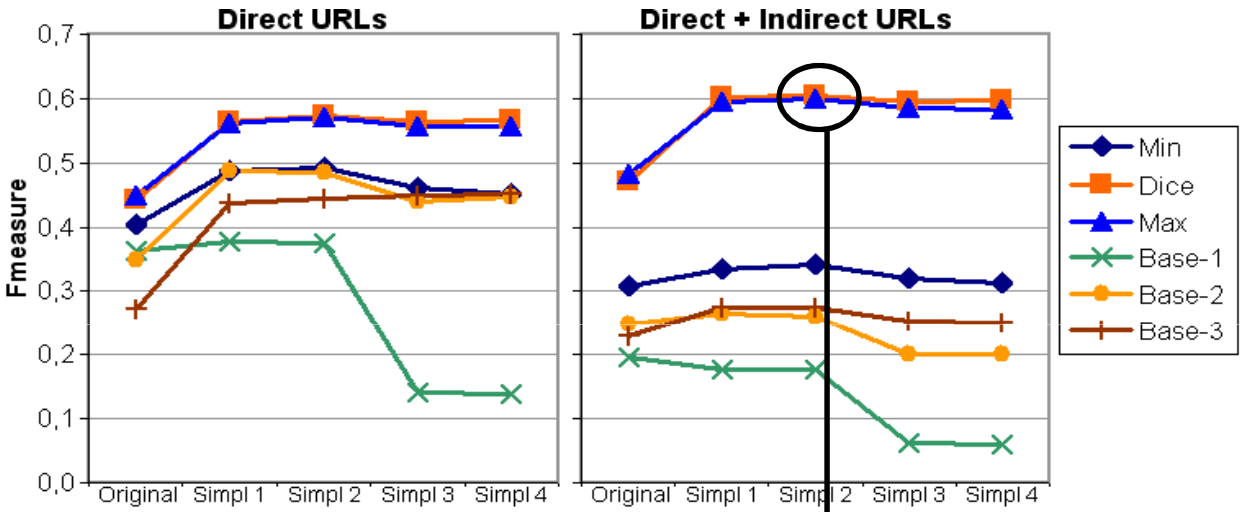
Überlappung der URLs

- Insgesamt um die 60,000 verschiedenen URLs
- Anzahl der URLs, die sich in allen Verzeichnissen befinden: ~ 500

	Dmoz ↔ Google	Dmoz ↔ Web	Dmoz ↔ Yahoo	Google ↔ Weby	Google ↔ Yahoo	Web ↔ Yahoo
URL Überlappung (ohne Duplikate)	12,963	1,650	1,485	1,679	1,561	1,695
=Verzeichnis-URLs (mit Duplikaten)	85% ↔ 87%	11% ↔ 12%	10% ↔ 5%	11% ↔ 13%	10% ↔ 5%	13% ↔ 5%

Ergebnisse der Instanz-basierten Matcher

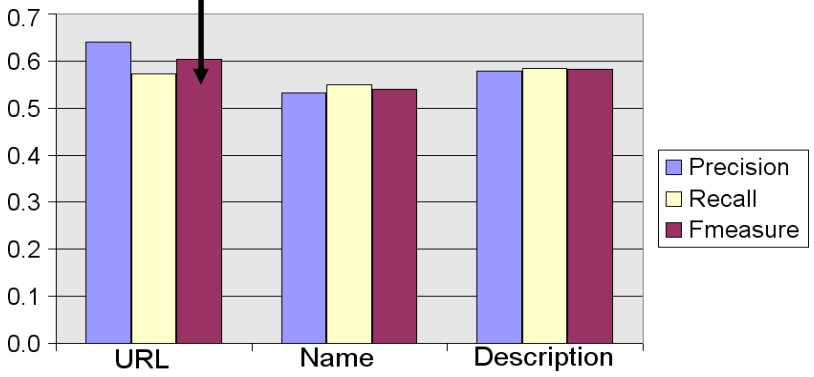
- URL-basierten Matching: **durchschnittlicher Fmeasure von 6 Matchaufgaben** für 5 Varianten der URL-Vorverarbeitung und 6 Ähnlichkeitsmaße



Für Dice und Max

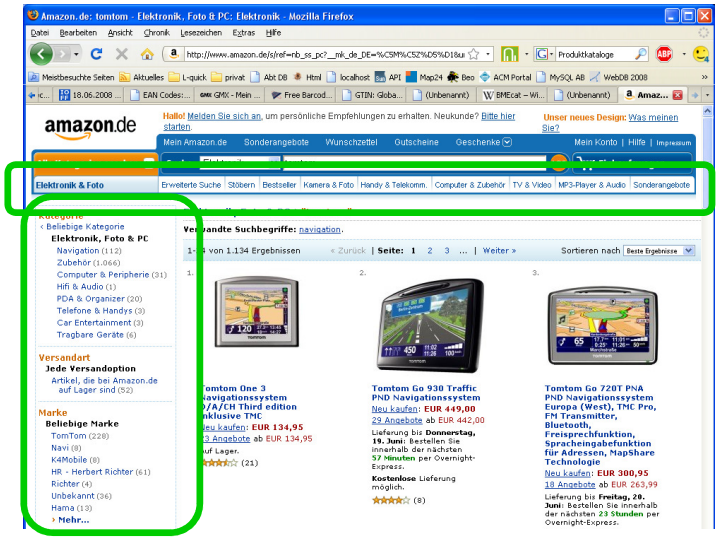
- Besser als Min und Base-k
- URL-Vorverarbeitung mit Vereinfachung (Simpl1-Simpl4) besser als Original-URLs
- Profitieren von der zusätzlichen Verwendung von indirekten URLs

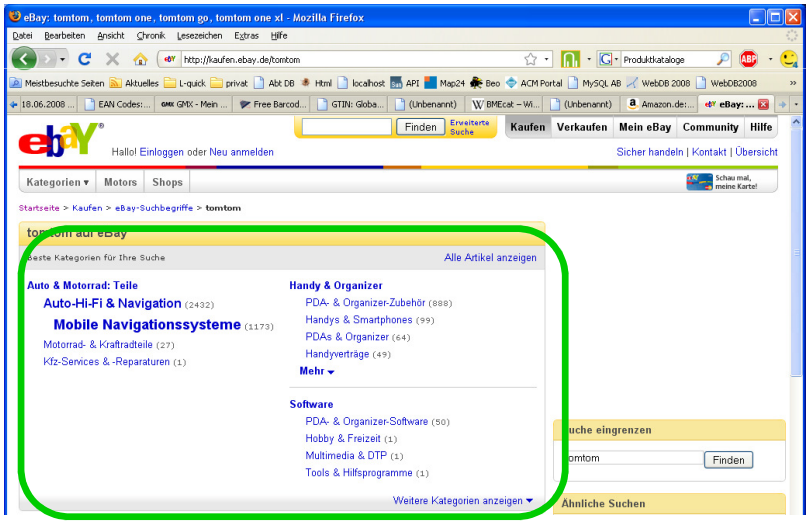
Höchster durchschnittlicher Fmeasure **0.6**: Dice, Simpl2, Direkte + Indirekte URLs



Produktkataloge - Aktueller Stand

- Produktkataloge: systematisch geordnete Sammlung von Informationen zu Produkten oder Dienstleistungen
- Auflistung erfolgt meist nach bestimmten Systematiken und Gruppierungen nach Eigenschaften
- Es existieren viele verschiedene Produktkataloge, z.B. Amazon oder Ebay





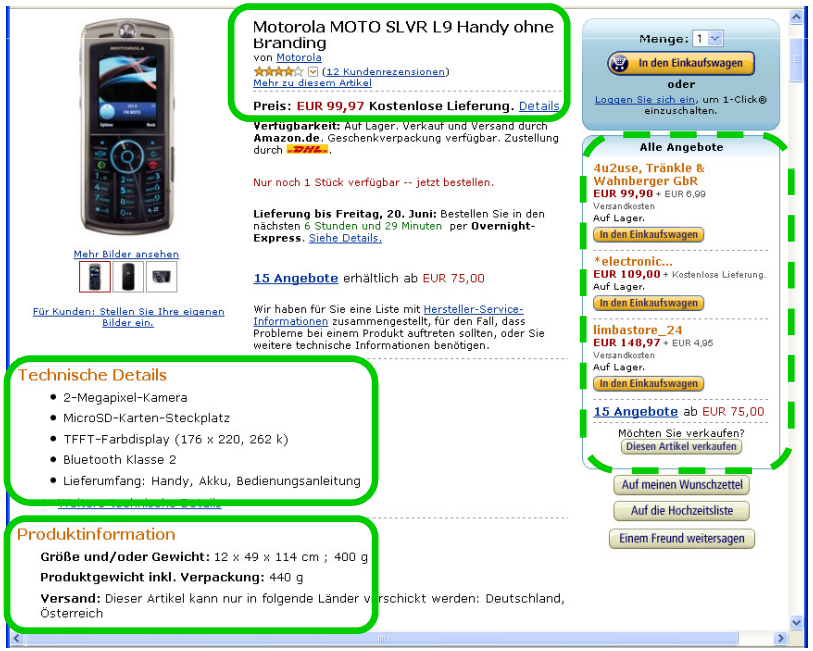
- Verschiedene Standardformate zum Austausch von Produktkatalogen, z.B. BMEcat und xCBL

Produktinformation

- Name
- Identifikationsnummern
  - Catalogintern, beispielsweise bei Amazon ASIN (Amazon Standard Identification Number) oder bei Ebay die Artikelnummer
  - Strichcode
  - Modellnummer
- Preis
- Beschreibung
- Größe
- Farbe
- Bewertungen
- Rangfolge beim Verkauf
- Bild(er)
- …

ASIN: B000JNYWBG

Artikelnummer: 160249046656



Instanz-basiertes Matching über eindeutige ID

- Vorteil: Test auf Gleichheit (nicht Ähnlichkeit) → schnell, eindeutig
- Nachteil: Manche Produktkataloge verwenden UPC, andere EAN, manche beides
- Idee:
  - Nutzen des Wissens im Internet um “Wissenslücke” zu füllen
    →EAN, UPC, ASIN, Modellnummer,
    Produktname, Hersteller,…

www.upc-codes.com
www.ean-codes.com
www.isbn-codes.com
www.model-numbers.com

0683728121587 - Apple iPod touch 16 GB without Software Updates

683728121587
Powered By IDAutomation.com

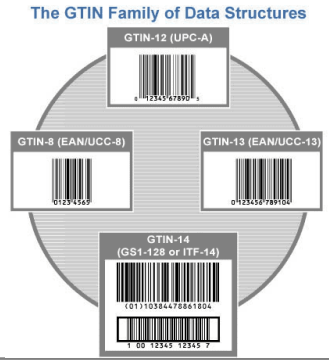
Publisher:	Apple Computer
Manufacturer:	Apple Computer
Manufacturer Part:	MA627LL/A
ASIN:	B000JNYWBG
UPC:	683728121587
EAN:	0683728121587
Prices:	Compare Prices on Apple iPod touch 16 GB without Software Updates at EAN Prices
Description:	Does not ship with the new software applications announced at MacWorld 2008; software upgrades are available via the online Apple store for $20

Strichcodes

- Produktkennzeichnung für Handelsartikel
- **UPC**
  - *Universal Product Code*, 12-stellig
  - 1973 in USA eingeführt

- **EAN**
  - *European Article Number*, 13-stellig (bzw. 8 für kleine Artikel)
  - Kompatibel zu UPC
    - Ergänzung von einer UPC-Nummer um eine führende Null ergibt gültige EAN
    - UPC wird deshalb von international tätigen Firmen genutzt
  - Kompatibel zu ISBN (Bücher) und ISSN (Zeitschriften)
    - EAN wird mit Hilfe der EAN-Ländernummer und Prüfziffer erzeugt

- **GTIN**
  - *Global Trade Item Number,* 14-stellig
  - Kompatibel zu UPC und EAN (führende Nullen)
  - Zukünftiger Einsatz



Datensatz (bisher)

- Amazon und Yahoo (US) mit Beschränkung auf Elektronik

	Amazon	Yahoo
#Kategorien	2250	1855
#Kategorien mit direkt assoziierten Instanzen	1956	1525
#Direkte Instanzen, mit	201618	6144
UPC	157964	4843
EAN	161061	0
ISBN	290	0
ASIN	201618	0

Erste Ergebnisse

- Qualität des Zusatzwissen:
  je mehr die IDs miteinander verlinkt sind, umso besser
- Folgende Daten basieren auf Daten von Amazon, Yahoo und Zusatzwissen

www.upc-codes.com
www.ean-codes.com
www.isbn-codes.com
www.model-numbers.com

	Anzahl	Nicht gefunden
EAN	164025	267
UPC	159156	429
ManufacturerPart	162547	-
ISBN	285	5
ASIN	207479	2153

- Weitere Erkenntnisse der Untersuchung:
  - Die EAN, die einer UPC zugeordnet ist, entspricht nicht der UPC mit führender Null 3,1% (4929)
  - ASIN ohne UPC, EAN und ManufacturerPart sind nur 3,2% (6740)

	EAN	UPC	ASIN	ISBN	ManufacturerPart
EAN	-	91,1%	100%	0%	85,4%
UPC	93,3%	-	100%	0%	86,8%
ASIN	81,4%	79,5%	-	0,1%	83,6%
ISBN	100%	8,8%	100%	-	19,6%
ManufacturerPart	83,9%	83,4%	100%	0%	-

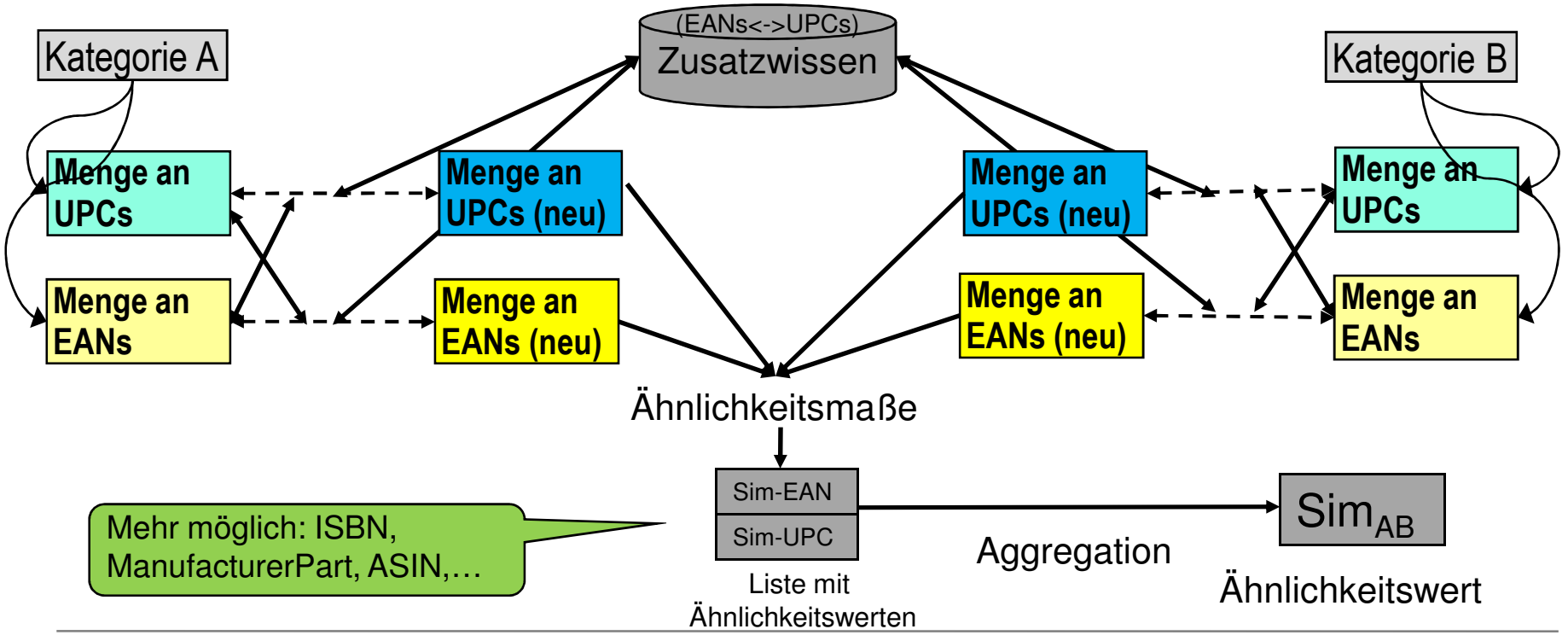
Ausnutzung des Zusatzwissens

UPC	**Originalwissen**		**Mit Zusatzwissen**		**Zusätzlich**
Amazon	154237	0,7% (max. möglich: 3,1%)	155644	0,7%	+1407
Yahoo	4843		5060		+217
zusammen	158044	21,4%	159568	22,5%	+1524
überlappend	1036		1136		+100

EAN	**Originalwissen**		**Mit Zusatzwissen**		**Zusätzlich**
Amazon	158061	0%	160749	0,7%	+2688
Yahoo	0		4604		+4608
zusammen	158061	0%	164240	24,2%	+6179
überlappend	0		1113		+1113

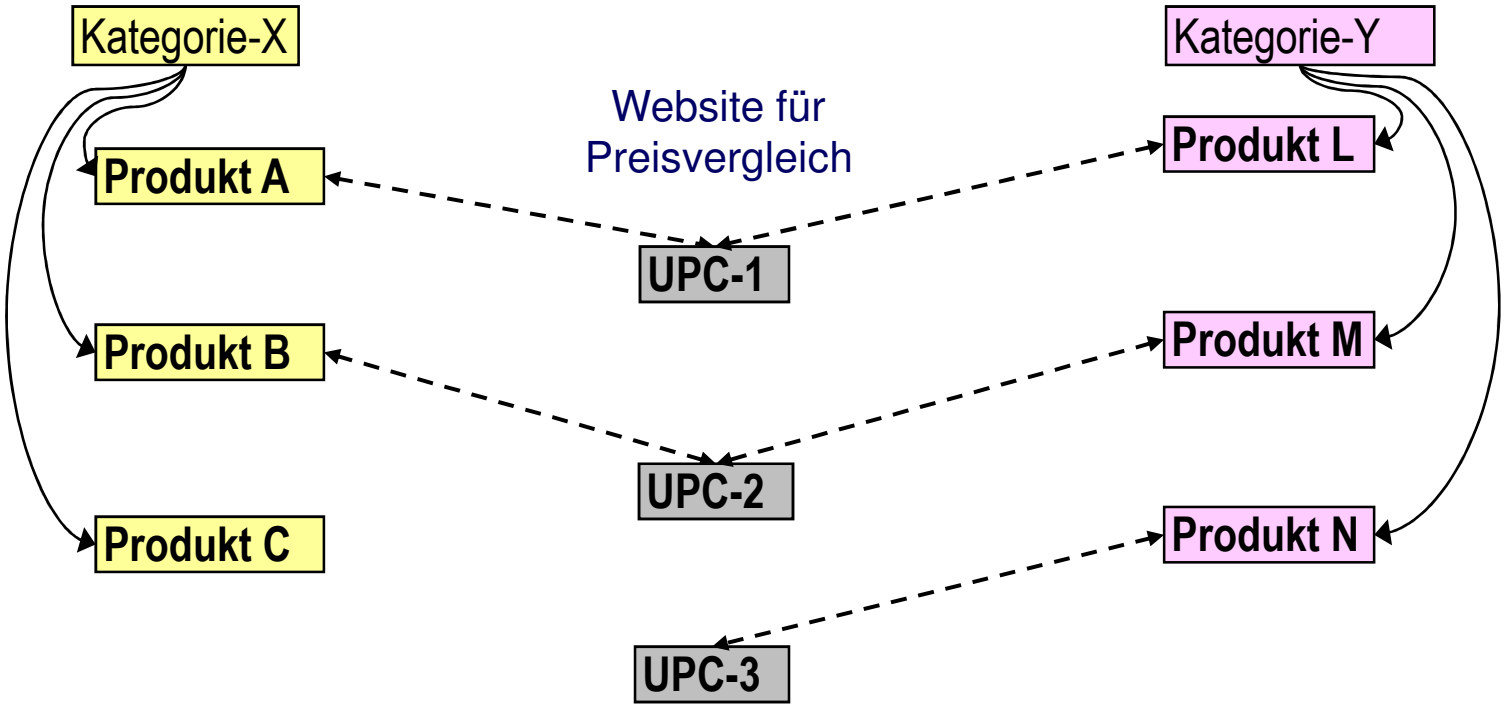
Nutzung der IDs:
1. Instanz-basiertes Matching der Produktkataloge

- Extrahieren aus den Produktinformationen: EAN, UPC
- Für alle EANs und UPCs Bestimmen der zugehörigen UPC bzw. EAN
  - Verwendung des Wissens von Online-Datenbanken (mit Datenwiederverwendung von früheren Vorgängen)
  - Beachte: nicht jede EAN hat UPC (und umgedreht)
- Verwendung verschiedener Ähnlichkeitsfunktionen: Base-k, Min, Dice, Max



Nutzung der IDs:
2. Instanzassoziationen über weitere Quellen nutzen

- Webseiten, die Preisvergleich anbieten, verlinken zu einem Produkt bei verschiedenen Anbietern
- Diese Instanzassoziationen können über Compose zum Matching verwendet werden



Nächsten Schritte

- Extraktion weiterer Produktkataloge
- Test, ob mit Verwendung von Zusatzwissen neue Korrespondenzen mit Base-K generiert werden (ohne Mappings möglich)
  - wenn ja, dann Überprüfung ob diese korrekt
- Erstellung eines Mappings (n:m, Teilmengenbeziehungen?)
  → Ausnutzung der Metadaten und Instanzen
- Auswertung von Preisvergleichs-Websites (z.B. PriceSpider.com)
- ASIN führt zu ManufacturerPart, diese hat eigene Seite mit EAN bzw. UPC (und anderer ASIN)

Danke für die Aufmerksamkeit.

Fragen? Anmerkungen?