(Semi-)Automatisches Ontologie Matching

Natanael Arndt

Universität Leipzig
Abteilung Datenbanken
Seminar Ontologie-Management

13. Januar 2009

Tutorin Frau Sabine Maßmann

Problemstellung

The distributed nature of ontology development has led to dissimilar ontologies for the same or overlapping domains. Thus, various parties with different ontologies do not fully understand each other. To solve these problems, it is necessary to use ontology mapping geared for interoperability. Choi u. a. [2006]

Begriffe

Matching ist die Suche nach Correspondences.

Correspondence ist eine Relation zwischen Objekten verschiedener Ontologien.

Alignment ist eine Menge von Correspondences zwischen zwei Ontologien. Das Alignment ist die Ausgabe des Matching Prozesses.

Definition

*The **matching process** can be seen as a function f which, from a pair of ontologies to match O and O′, an input alignment A, a set of parameters p and a set of oracles and resources r, returns an alignment A′ between these ontologies:*

$$A' = f(O, O', A, p, r)$$

Euzenat u. Shvaiko [2007]

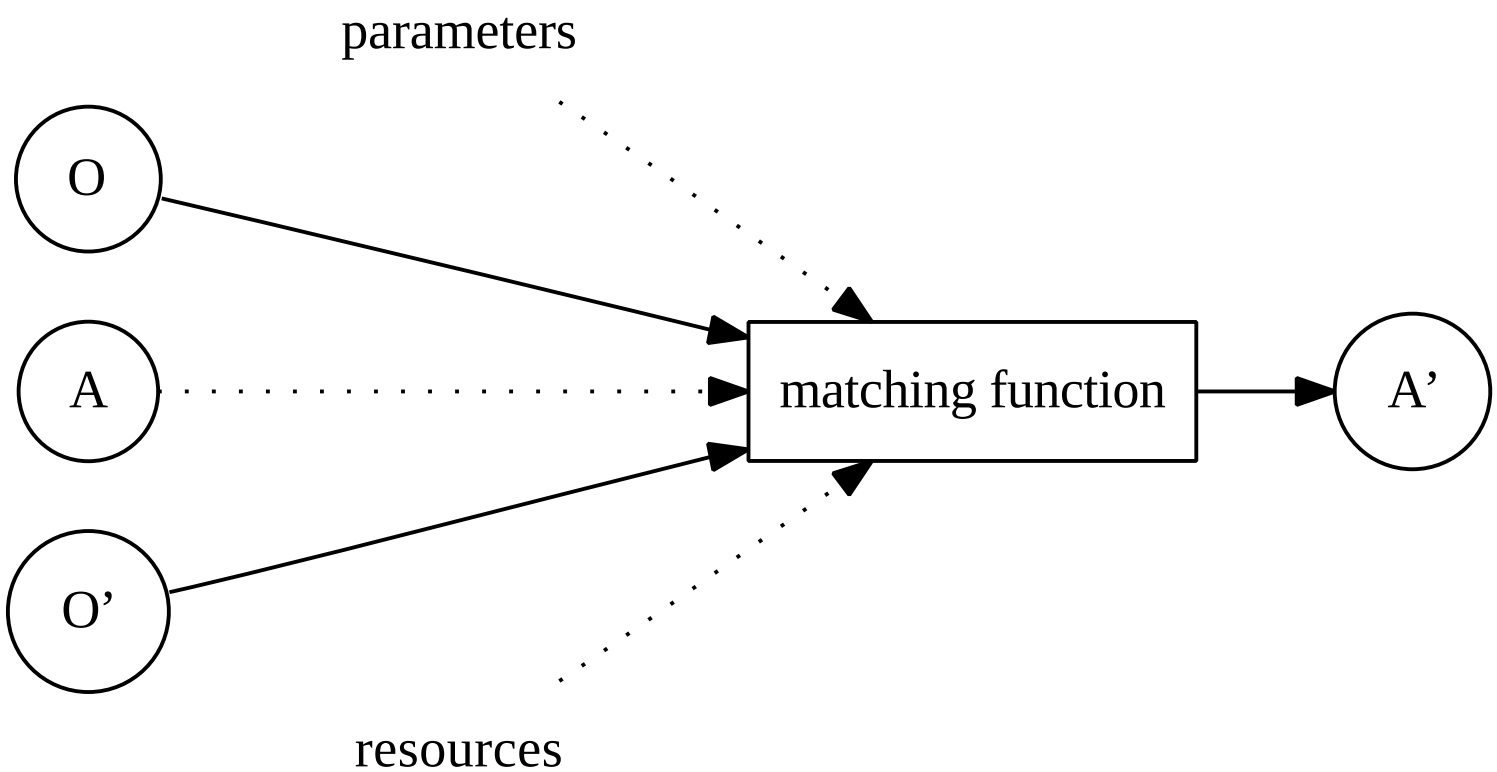


Abbildung: Der Matching Prozess Euzenat u. Shvaiko [2007]

Ein kleines Beispiel zum Einstieg

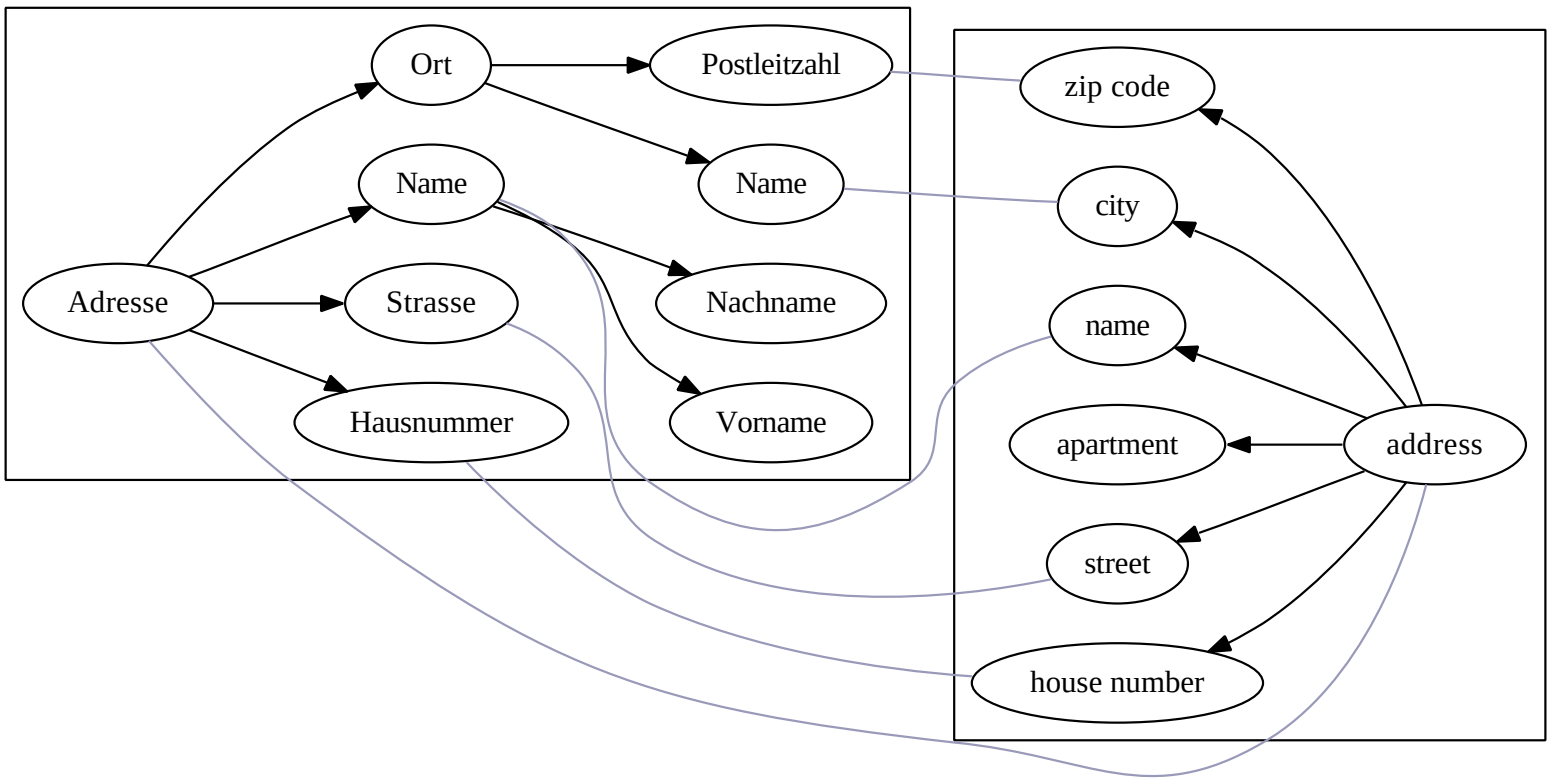


Abbildung: Ein Alignment zwischen zwei unterschiedlichen Adressformaten in verschiedenen Sprachen.

Warum Probleme beim Ontologie Matching auftreten

Ontologies are seen as the solution to data heterogeneity on the web. However, the available ontologies could themselves introduce heterogeneity: given two ontologies, the same entity can be given different names or simply be defined in different ways, whereas both ontologies may express the same knowledge but in different languages. Euzenat u. Valtchev [2004]

Mögliche Probleme beim (semi-)automatischen Ontologie-Matching

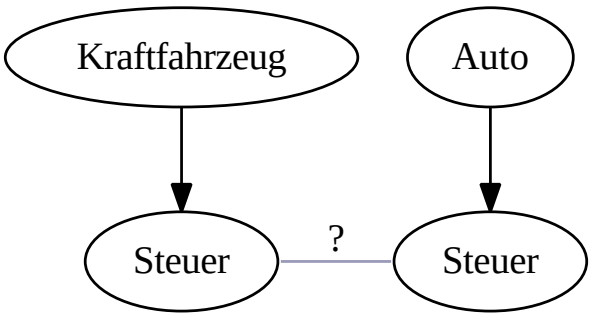


Abbildung: Teekesselchen (Homonyme)

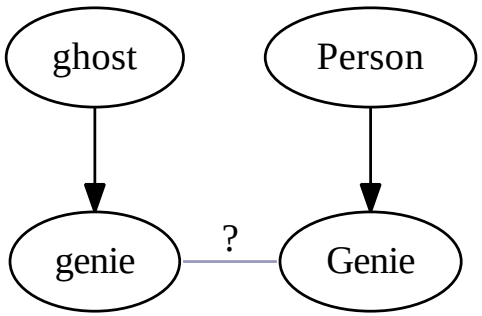


Abbildung: „false friends“

Außerdem können Abkürzungen („THW“ $\leftrightarrow$ „Technisches Hilfswerk“, „Turnverein Hassee-Winterbek“ und „Tidehochwasser“), Kunstwörter („Kripo“), Synonyme („Abendstern“, „Morgenstern“ und „Venus“ Frege [1892]) oder Übersetzungen zu nicht erkannten Correspondences führen.

Klassifikation

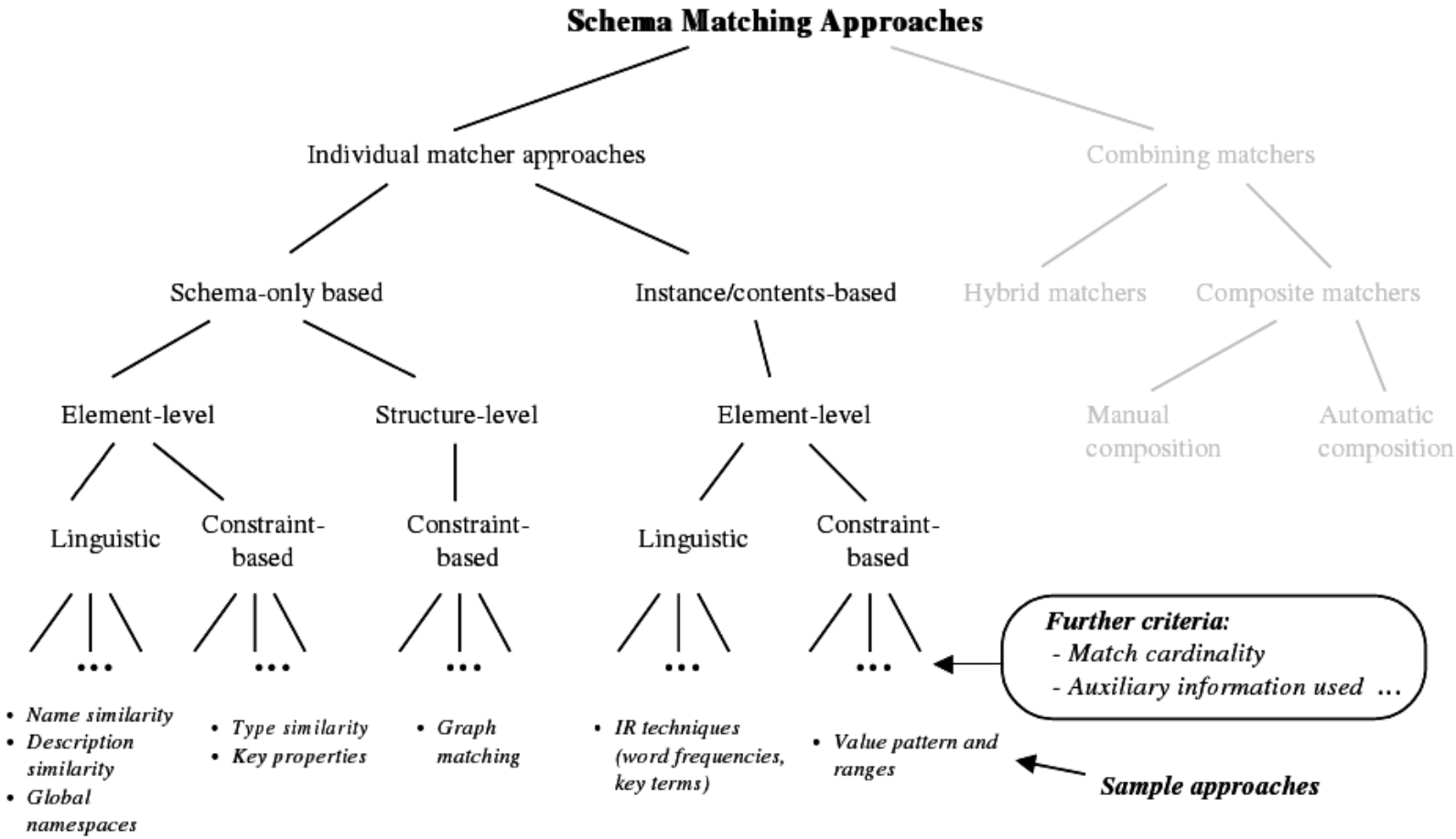


Abbildung: Schema Matching Approaches Rahm u. Bernstein [2001]

Levenshtein-Distanz

Es stehen nach Levenshtein [1966] folgende drei Operationen zur Verfügung:

1. „Einfügen“ $\Lambda \rightarrow 0; \Lambda \rightarrow 1$
2. „Löschen“ $0 \rightarrow \Lambda; 1 \rightarrow \Lambda$
3. „Ersetzen“ $0 \rightarrow 1; 1 \rightarrow 0$

- „Schaf“ und „Schafe“: $d = 1$
- „Wolf“ und „Schaf“: $d = 4$

(Λ ist das leere Wort)

Die Levenshtein-Distanz ist die minimale Anzahl der Editieroperationen, die nötig sind um einen String in den anderen umzuwandeln.

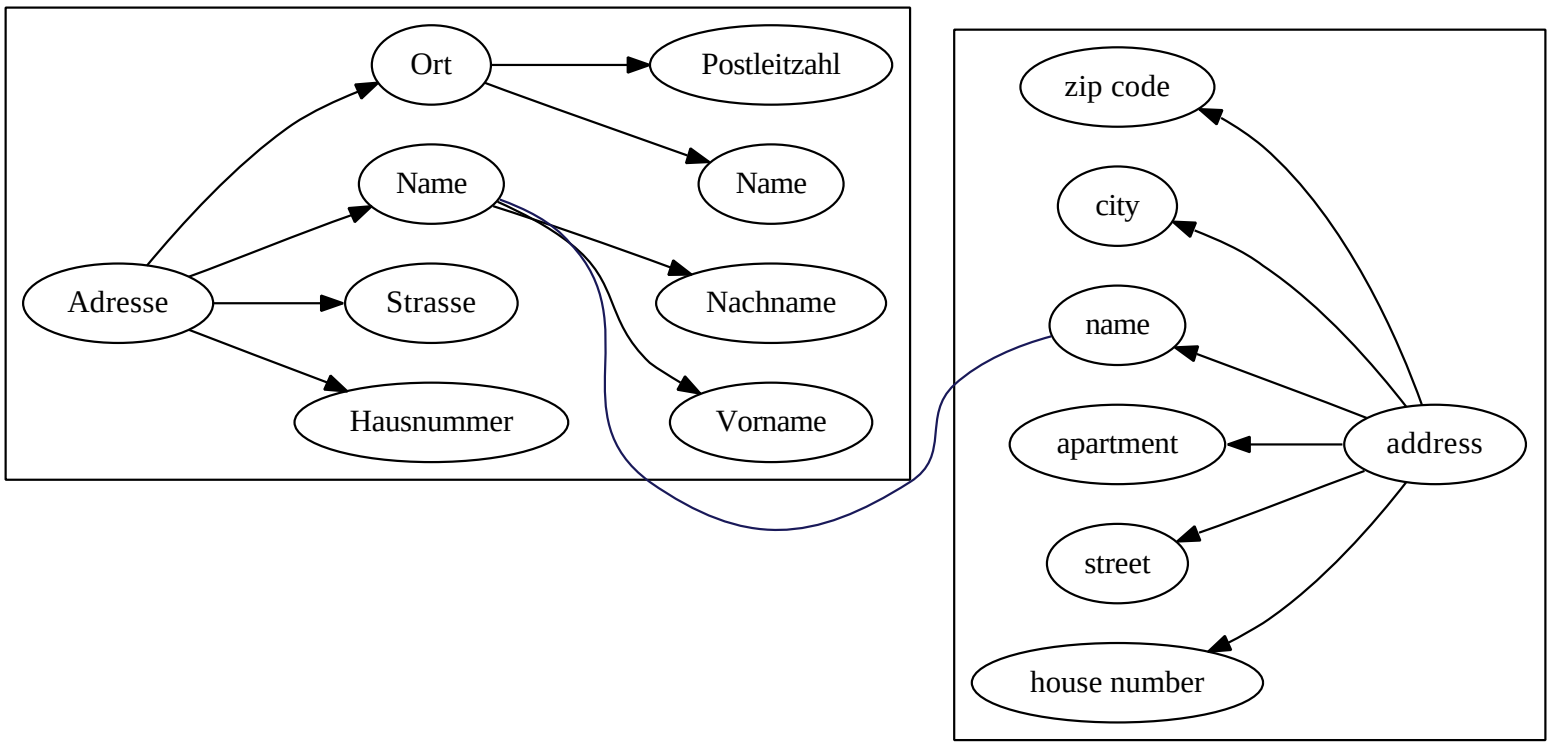


Abbildung: Der Ähnlichkeitswert von „Name“ und „name“ beträgt nach Berechnung der Levenshtein-Distanz $0,75$.

N-Gramm

Hier werden hingegen die Häufigkeiten des Auftretens verschiedener Buchstabengruppen betrachtet. Der Text wird dabei in Buchstabengruppen der Länge n (N-Gramme) zerlegt und die Häufigkeit der einzelnen Buchstabenfolgen gezählt.

$$\sigma(s,t) = |ngram(s,n) \cap ngram(t,n)|$$

$$\overline{\sigma}(s,t) = \frac{|ngram(s,n) \cap ngram(t,n)|}{\min(|s|,|t|) - n + 1}$$

$$ngram(\text{„Schaf“}, 3) = \{\text{„Sch“}, \text{„cha“}, \text{„haf“}\}$$

$$ngram(\text{„Schafe“}, 3) = \{\text{„Sch“}, \text{„cha“}, \text{„haf“}, \text{„afe“}\}$$

$$\overline{\sigma}(\text{„Schaf“}, \text{„Schafe“}) = \frac{3}{5-3+1} = 1$$

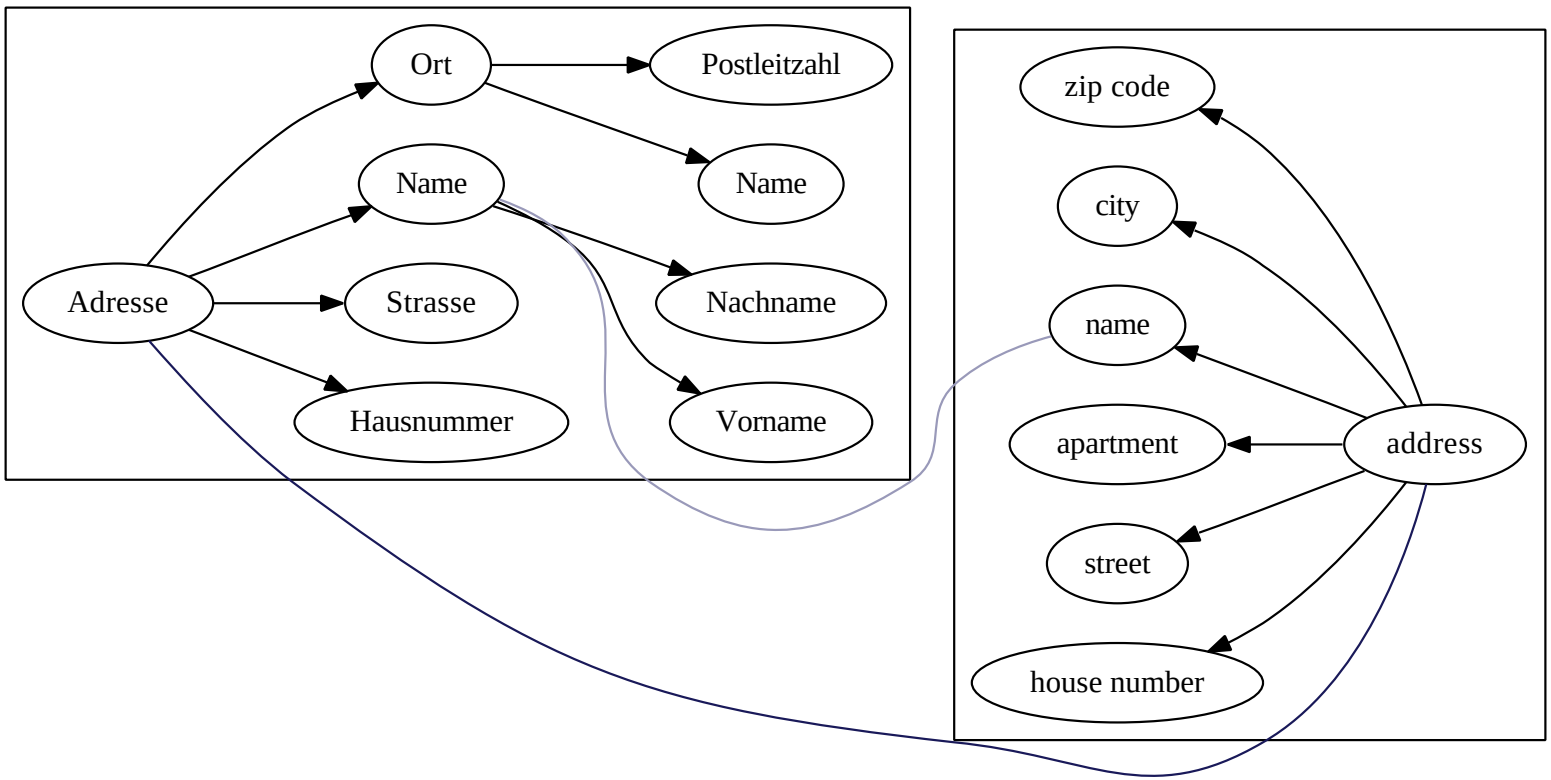


Abbildung: Der Ähnlichkeitswert von „Adresse“ und „address“ beträgt nach Vergleich 3-Gramme $0,6$.

Nutzung externer Quellen

Zum Vergleich können außerdem externe Quellen genutzt werden um Beziehungen zwischen Wörtern herzustellen:

- Wortliste (Synonymlisten, einfache Übersetzungslisten)
- Thesaurus (Sinn- und sachverwandte Wörter)
- WordNet, Wortschatz (Universität Leipzig)

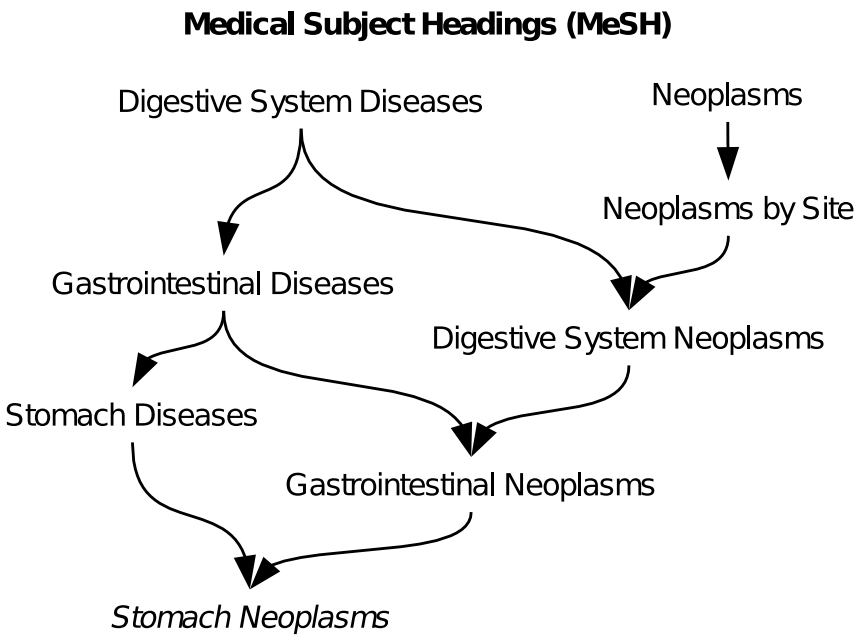


Abbildung: Beispiel eines Thesaurus

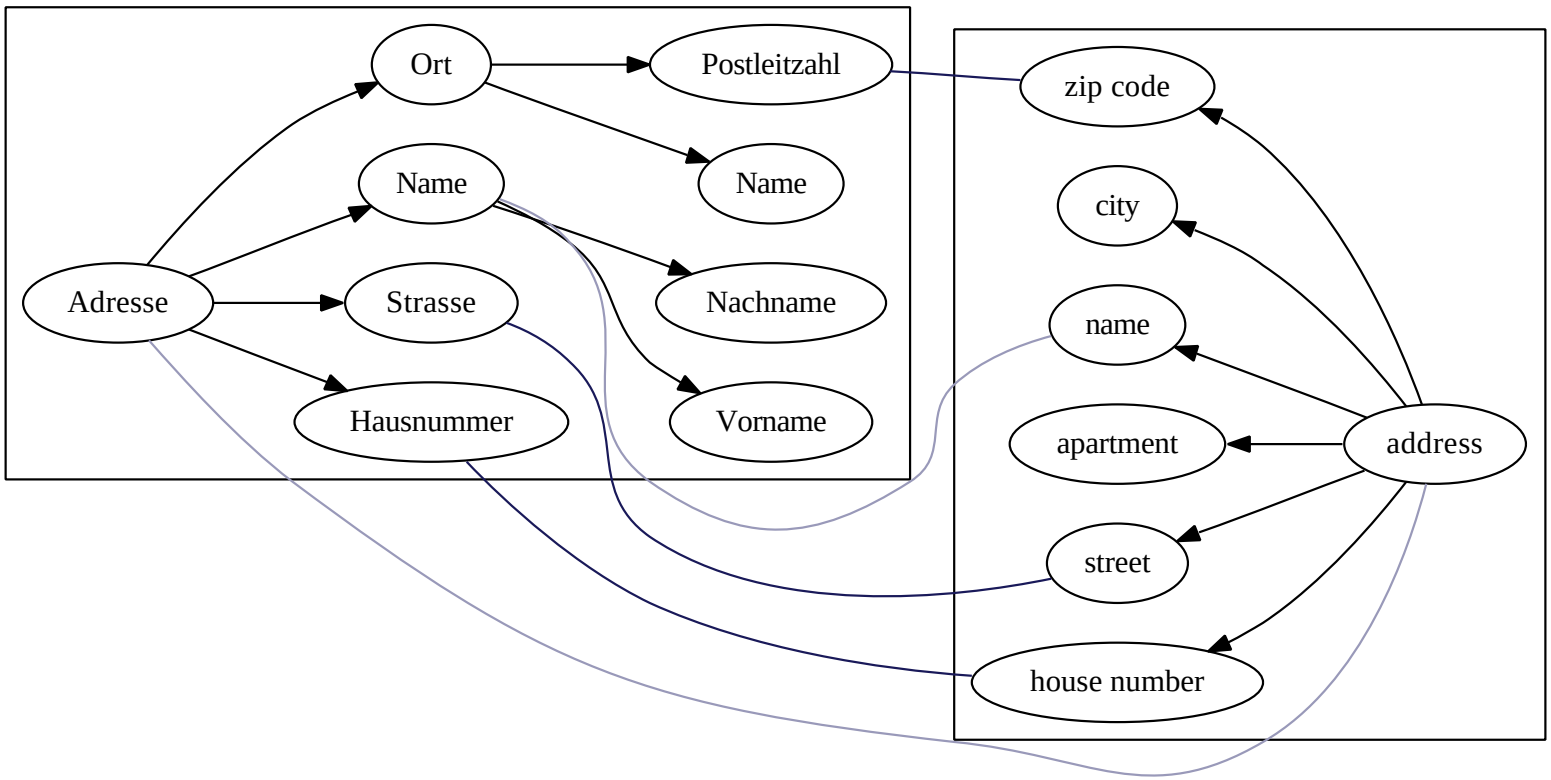


Abbildung: Die Correspondences der Wörter „Postleitzahl“ ↔ „zip code“, „Strasse“ ↔ „street“, „Hausnummer“ ↔ „house number“ können über Wörterbücher gefunden werden.

Strukturvergleich

Bei einen Strukturvergleich werden, im Gegensatz zu den bisherigen Methoden, die verschiedenen Namen oder Bezeichner der Elemente nicht betrachtet.

Graph-based Annahme: „Der Apfel fällt nicht weit vom Stamm“ (Wenn die Eltern matchen, dann ist es wahrscheinlich, dass auch die Kinder matchen. Sharma [2006])

Weitere Ansätze sind Taxonomiebasierte und Modellbasierte Ansätze.

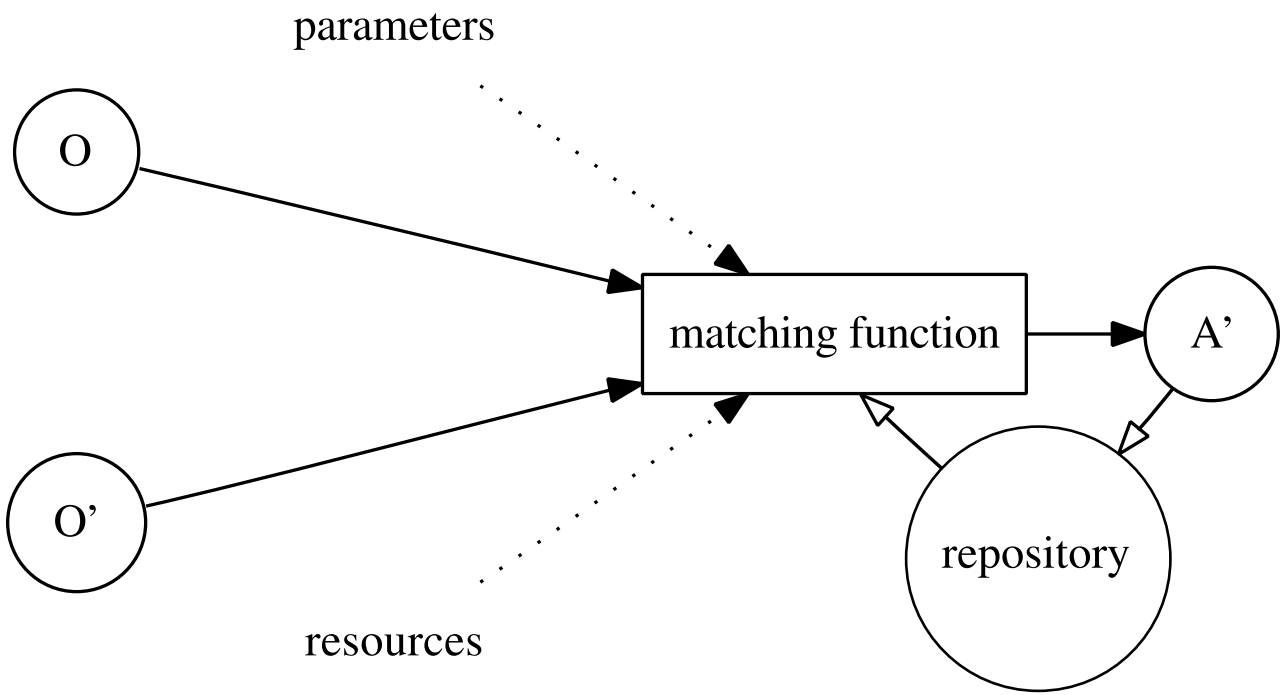


Abbildung: The Reuse Matcher

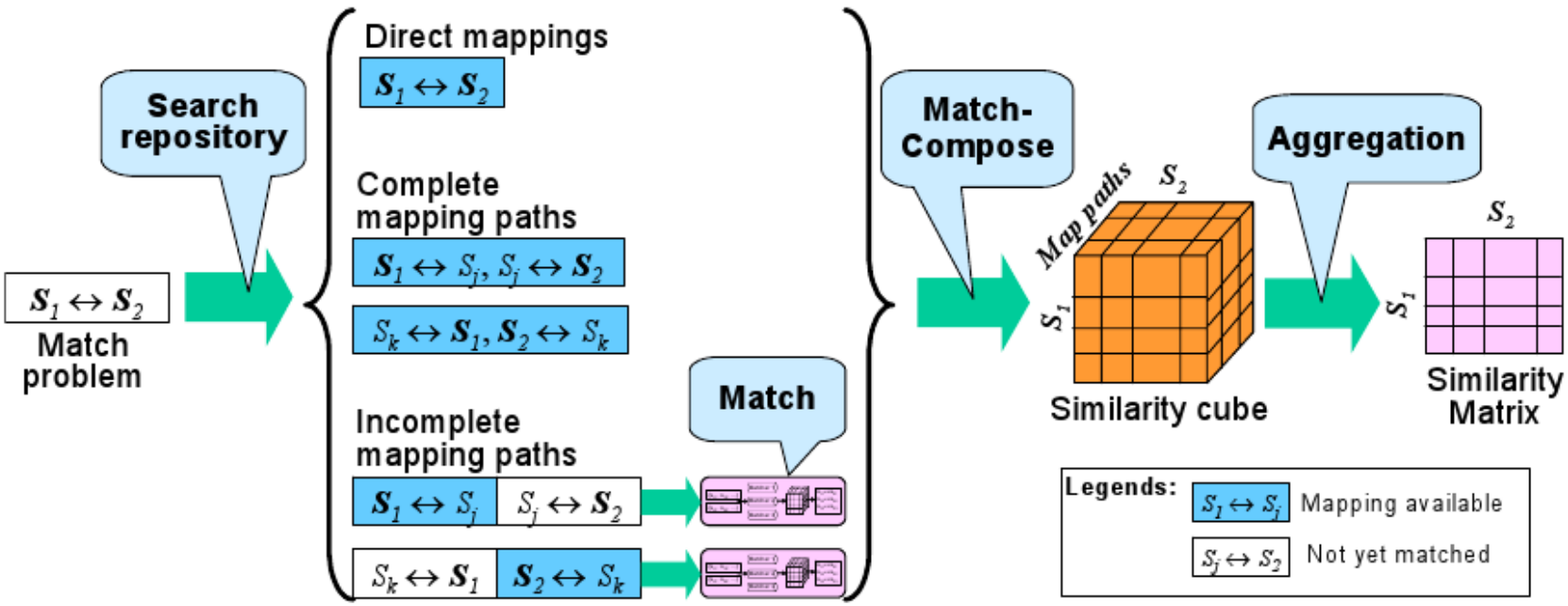


Abbildung: „Schema-level reuse in the Reuse matcher“ Do [2005]

Search repository

In diesem Schritt wird das Repository nach einem passenden Mappingpath durchsucht. Es gibt drei Arten von Mappingpathes, „Direct Mapping“, „Complete Mappingpath“ und „Incomplete Mappingpath“.

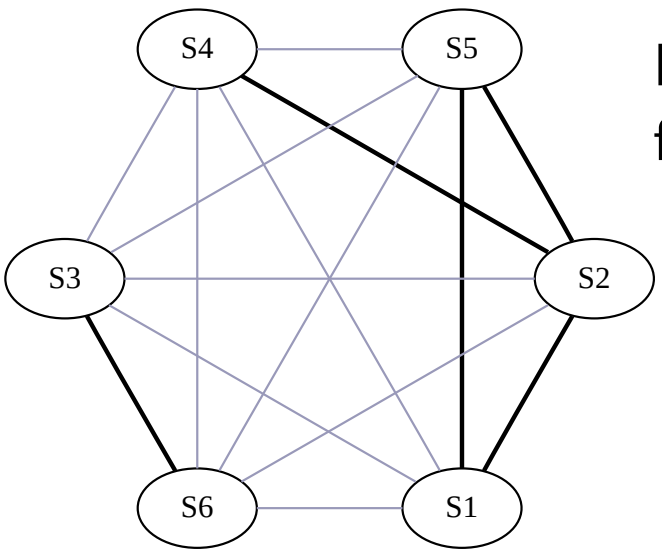
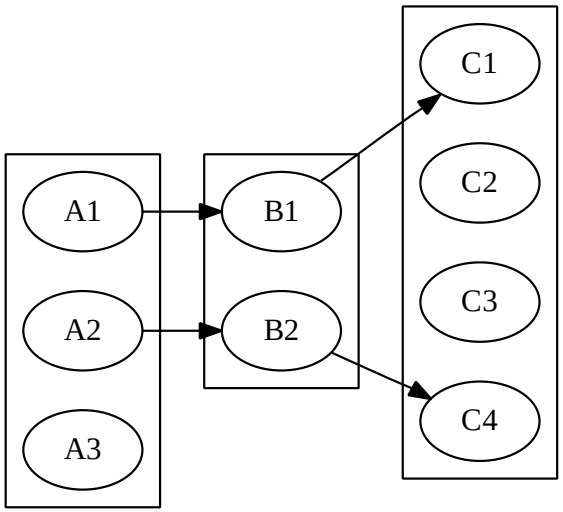


Abbildung: Das Alignmentrepository

Für ein Matching $S_1 \leftrightarrow S_4$ ergeben sich folgende Mappingpfade:

- Länge 1 $(S_1 \leftrightarrow S_4)$ es ist kein „Direct Mappingpath“ vorhanden.
- Länge 2 $S_1 \leftrightarrow S_2 \leftrightarrow S_4$ „Complete Mappingpath“
  $S_1 \leftrightarrow (S_5 \leftrightarrow S_4)$
  „Incomplete Mappingpath“

Match-Compose



$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Abbildung: Durch den Reuse Matcher wurden mehrere Mappingpathes erzeugt. Aufgrund der Transitivität ($A \leftrightarrow B; B \leftrightarrow C \Rightarrow A \leftrightarrow C$) des Mappings erzeugen wir aus jedem Pfad eine Matrix. Diese Matrizen werden zu einem „Similarity cube“ zusammengestellt.

Aggregation

Der „Similarity Cube“ wird in diesem Schritt zu einer „Similarity Matrix“ vereinfacht, welche das Matching zwischen S_1 und S_4 darstellt. Je nach Einstellung kann man hier verschiedene Funktionen zur Zusammenstellung verwenden, etwa $\max(), \min()$, einen Mittelwert oder gar eine Funktion die unterschiedliche Pfade unterschiedlich gewichtet.

Anwendung auf Ontologien

- Welche Objekte werden miteinander verglichen? (Namen, Datentyp, Pfad, Instanzen)
- Welchen Algorithmus verwende ich am besten? (Stringvergleich, Strukturvergleich)
- Zusatzeingaben: Parameter, Ressourcen
- Output dieses Vorgangs Ähnlichkeitswerte

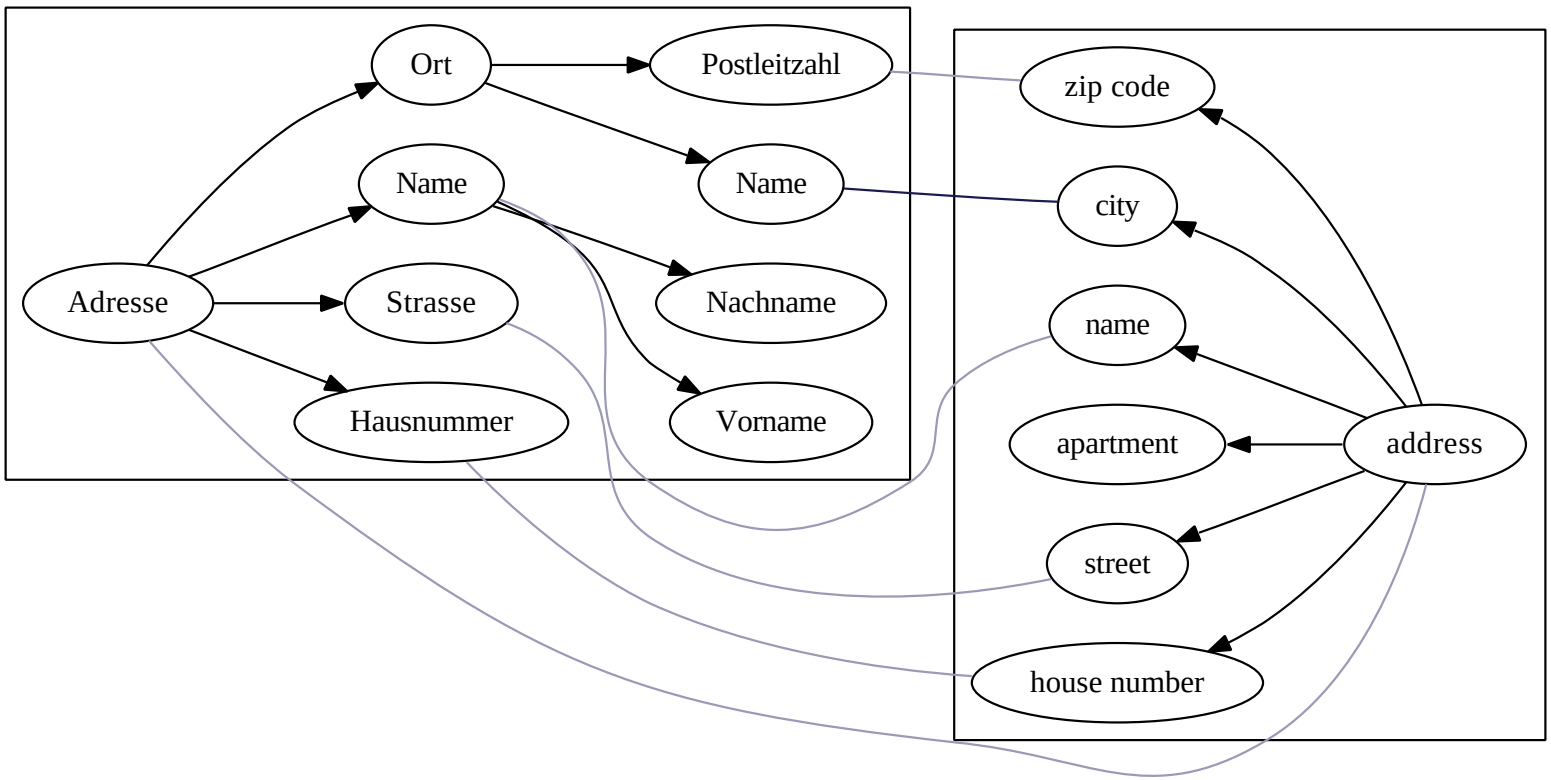


Abbildung: Durch Vergleich der Instanzen konnte eine Correspondence zwischen „Ort→Name“ und „city“ festgestellt werden.

Zusammenfassung

- Problemstellung des Ontologie Matchings
- Hürden auf dem Weg zum fertigen Alignment
- Verschiedene Matchansätze
  - Stringvergleich (Levenshtein, N-Gramm)
  - Externe Quellen (Wortlisten, Thesauri, WordNet, Wortschatz)
  - Strukturvergleich (Graphenbasiert)
  - Nutzung vorheriger Erkenntnisse (Reuse Matcher)
- Anwendung der Matchansätze auf Ontologien

Herzlichen Dank für Ihre Aufmerksamkeit.

Fortsetzung folgt ;-)

[Choi u. a. 2006] Choi, Namyoun ; Song, Il-Yeol ; Han, Hyoil: A Survey on Ontology Mapping. In: *SIGMOD Record* 35 (2006), S. 34–41

[Do 2005] Do, Hong H.: *SCHEMA MATCHING AND MAPPING-BASED DATA INTEGRATION*, University of Leipzig, Germany, Diss., 2005

[Euzenat u. Shvaiko 2007] Euzenat, Jérôme ; Shvaiko, Pavel: *Ontology Matching*. Springer, 2007

[Euzenat u. Valtchev 2004] Euzenat, Jérôme ; Valtchev, Petko: Similarity-based ontology alignment in OWL-Lite / INRIA Rhône-Alpes, Université de Montréal. 2004. – Forschungsbericht

[Frege 1892] Frege, Gottlob: Über Sinn und Bedeutung. In: *Zeitschrift für Philosophie und philosophische Kritik* 100. (1892), 25–50. `http://www.gavagai.de/HHP31.htm`

[Levenshtein 1966] Levenshtein, V. I.: binary codes capable of correcting deletions, insertions, and reversals. In: *soviet physics-doklady* 10 (1966), S. 707–710

[Rahm u. Bernstein 2001] Rahm, Erhard ; Bernstein, Philip A.: A survey of approaches to automatic schema matching. In: *The VLDB Journal* 10 (2001), S. 334–350

[Sharma 2006] Sharma, Asankhaya: Ontology Matching Using Weighted Graphs / National Institute of Technology, India. 2006. – Forschungsbericht