Foto: Comstock Images

Methoden der Dublettenerkennung

Was sind Dubletten und wie werden sie identifiziert? Leicht verständlich erklärt Professor Dr. Felix Naumann wichtige theoretische Grundlagen und Begriffe, die auch Praktikern helfen.

ALS DUBLETTEN bzw. Duplikate werden verschiedene Datensätze bezeichnet, die *dasselbe Realweltobjekt* repräsentieren. Typische Beispiele sind mehrfach geführte Kunden in einem Kundenmanagementsystem, verschiedene Repräsentationen eines Produkts oder doppelt gebuchte Bestellungen. Duplikate treten in der Informationsintegration häufig auf, da Objekte oftmals in unterschiedlichen Systemen mehrfach geführt werden, die erst bei der Integration zusammenkommen. So ist ein Produkt in der Software der Produktionsplanung, der Lagerhaltung, dem Verkauf, im Marketing etc. vertreten. Werden diese Systeme, zum Beispiel im Rahmen eines Data-Warehouse-Projekts, zusammengeführt, müssen die verschiedenen Repräsentationen erkannt und zu einer homogenen Darstellung vereinigt werden. Aber auch in nicht integrierten Systemen entstehen Duplikate, etwa durch die Mehrfacheingabe eines Kundenkontaktes.

Neben dem vergleichsweise geringen Mehraufwand zur Speicherung der Duplikate sind die wirtschaftlichen Schäden durch das Nichterkennen der Duplikate groß: Kunden werden bei Marketingaktionen mehrfach angeschrieben; Duplikate verfälschen Statistiken über Lagerbestände; der Gesamtumsatz eines Abnehmers wird nicht erkannt. Umgekehrt wird eine potenziell starke Verhandlungsposition nicht erkannt, wenn zum Beispiel ein Unternehmen aufgrund doppelter Aufführung im Lagersystem bei einem Zulieferer mehrfach Produkte bestellt, ohne dies zu bemerken.

> Groß sind die wirtschaftlichen Schäden durch das Nichterkennen der Duplikate.

Ironischerweise ist die Duplikaterkennung selbst unter einer Vielzahl von Namen bekannt. Im Deutschen spricht man auch von Dublettenerkennung, im Englischen auch von *record linkage, object identification* oder von *entity resolution*.

Ziele der Duplikaterkennung

Das Grundprinzip der Duplikaterkennung ist der paarweise Vergleich aller Datensätze. Aus dem Vergleich wird eine Maßzahl für die Ähnlichkeit der Datensätze abgeleitet. Ist diese Zahl größer als ein Schwellwert, so gelten die beiden Datensätze als Duplikate. Das einfachste Ähnlichkeitsmaß für zwei Datensätze ist deren Identität. Für die meisten Anwendungen ist Identität aber kein geeignetes Maß, da zu viele tatsächliche Duplikate, die eben nicht genau identisch sind, übersehen werden. Stattdessen werden anwendungsabhängige Ähnlichkeitsmaße verwendet.

Methoden zur Duplikaterkennung müssen zwei Ziele in Einklang bringen: Die Methoden sollen *effektiv* sein, das heißt, dass das Ergebnis, also die Menge der erkannten Duplikate, von hoher Qualität sein soll. Die Effektivität hängt von dem gewählten Ähnlichkeitsmaß und dem Schwellwert ab. Die Methode soll zudem *effizient* sein, ihre Laufzeit soll also mit der Menge an Datensätzen skalieren. Die Laufzeit von Duplikaterkennungsläufen auf großen Datenmengen mit Hunderttausenden oder Millionen Datensätzen sollte in Stunden, nicht in Tagen gemessen werden. Effizienz gewinnt man durch geschicktes Auswählen der tatsächlich zu vergleichenden Datensatzpaare.

Effktivität

Duplikaterkennungsmethoden vergleichen Paare von Datensätzen und

erklären sie entweder zu Duplikaten oder zu Nicht-Duplikaten. Dabei passieren Fehler, wenn Duplikate nicht als solche erkannt oder Nicht-Duplikate fälschlicherweise als Duplikate identifiziert werden. Im Ergebnis einer Methode sind damit vier Arten von Paaren enthalten, wie die Abbildung rechts zeigt.

Um die Effektivität von Duplikaterkennungsmethoden zu bewerten, werden zwei Maße verwendet: *Precision* und *Recall*. Die Maße stammen aus dem Bereich des *Information Retrieval* und werden dort unter anderem verwendet, um Ergebnisse auf Suchanfragen zu bewerten. Precision misst den Anteil der *true-positives* an allen von der Methode als Duplikate bezeichneten Paaren. Eine hohe Precision besagt also, dass erkannte Duplikate getrost als solche angenommen werden können. Eine manuelle Überprüfung der Ergebnisse ist unnötig. Hohe Precision wird durch ein besonders gutes und insbesondere „strenges“ Ähnlichkeitsmaß oder einen besonders hohen Schwellwert erreicht.

Recall hingegen misst den Anteil der *true-positives* an allen tatsächlichen Duplikaten. Ein hoher Recall bezeugt also, dass viele der tatsächlichen Duplikate gefunden wurden, allerdings möglicherweise auch viele Nicht-Duplikate als Duplikate ausgewiesen werden. Hoher Recall wird durch ein tolerantes Ähnlichkeitsmaß und einen niedrigen Schwellwert erreicht. Vor der Weiterverarbeitung ist eine manuelle Prüfung der Ergebnisse nötig; dafür werden aber nur wenige tatsächliche Duplikate übersehen.

Welche der beiden Aspekte das Hauptziel einer Methode zur Duplikaterkennung sein soll, ist eine anwendungsabhängige Entscheidung. In der Regel wird für eine zusammenfassende Bewertung der Güte von Duplikaterkennungsverfahren eine Kombination beider Maße benutzt, der so genannte *f-measure* – das harmonische Mittel aus Precision und Recall.

Mögliche Ergebnisse der Duplikaterkennung

Methode \ Realität	Duplikat	Nicht Duplikat
Duplikat	true-positive	false-positive
Nicht Duplikat	false-negative	true-negative

Effizienz

Um bei gegebenem Ähnlichkeitsmaß und Schwellwert alle Duplikate innerhalb einer Tabelle der Größe n zu finden, müssen theoretisch $(n^2-n)/2$ Vergleiche durchgeführt werden, da jeder Datensatz mit jedem anderen Datensatz verglichen werden muss. Für eine Tabelle mit nur 10 000 Datensätzen sind also bereits circa 50 Millionen Vergleiche notwendig. Die wichtigste Optimierung in Duplikaterkennungsmethoden ist deshalb die Vermeidung möglichst vieler solcher Vergleiche. Statt alle Paare zu betrachten, wird die Tabelle in (eventuell überlappende) Partitionen unterteilt und Paare nur innerhalb ihrer Partitionen verglichen. Diese auch als „blocking“ bekannte Strategie verschlechtert tendenziell den Recall, da Duplikate möglicherweise über mehrere Partitionen verteilt liegen und damit nicht mehr gefunden werden. Die Wahl der richtigen Partitionierung ist deshalb von entscheidender Bedeutung.

Die Abbildung auf Seite 42 unten veranschaulicht diesen Zielkonflikt zwischen Precision, Recall und Effizienz.

Ähnlichkeitsmaße

Ein Ähnlichkeitsmaß vergleicht zwei Datensätze. In der Regel wird ein Gesamtmaß für Datensätze aus Maßen für die einzelnen Attributwerte zusammengesetzt. Das Gesamtmaß sollte eine hohe Zahl zurückgeben, wenn die beiden Datensätze wahrscheinlich Duplikate sind. Ähnlichkeitsmaße sind in höchstem Maße anwendungsabhängig. Gerade für numerische Werte ist es extrem schwierig, allgemein gültige Regeln anzugeben – Unterschiede in Altersangaben, Lagerbeständen oder Geldbeträgen benötigen alle eigene Maße. Im Folgenden geht es daher nur um allgemeine Ähnlichkeitsmaße für Zeichenketten. Für viele Datentypen lassen sich in konkreten Anwendungen spezialisiertere Ähnlichkeitsmaße definieren.

Für viele Datentypen lassen sich spezialisierte Ähnlichkeitsmaße definieren.

Foto: HPI/K. Herschelmann

DER AUTOR

Prof. Dr. Felix Naumann leitet das Fachgebiet Informationssysteme am Hasso-Plattner-Institut für Softwaresystemtechnik an der Universität Potsdam. Er untersucht den effizienteren und effektiveren Umgang mit heterogenen Informationen in großen, autonomen Systemen. Dazu zählen unter anderem Methoden der Informationsintegration, der Informationsqualität und Datenreinigung, der Informationssuche und des Metadatenmanagements.

Edit-basierte Ähnlichkeitsmaße

Edit-basierte Ähnlichkeitsmaße sind Maße, die zwei Zeichenketten buchstabenweise vergleichen. Je ähnlicher die Buchstabenmenge und ihre Anordnung in den Zeichenketten sind, desto höher ist die Gesamtähnlichkeit.

Das am meisten verbreitete edit-basierte Ähnlichkeitsmaß ist die *Edit-Distanz* oder *Levenshtein-Distanz*. Sie ist definiert als die minimale Anzahl an Edit-Operationen (einfügen, löschen, ersetzen), um die eine Zeichenkette in die andere zu überführen. So haben die Wörter RASEN und HASE eine Edit-Distanz von 2: RASEN kann in HASE überführt werden, indem man das führende R mit einem H ersetzt und das schließende N löscht. Um die Edit-Distanz in ein Ähnlichkeitsmaß umzuwandeln, wird sie zunächst auf die Länge des längeren der beiden Zeichenketten normalisiert und anschließend von 1 abgezogen.

Weitere edit-basierte Ähnlichkeitsmaße sind das *SOUNDEX-Maß*, das die Aussprache von Wörtern bei der Ähnlichkeitsbewertung einbezieht, und die *Jaro-Winkler-Ähnlichkeit*, die Transpositionen einzelner Buchstaben speziell berücksichtigt und vor allem für kurze Wörter geeignet ist.

Die Wörter RASEN und HASE haben eine Edit-Distanz von 2.

Tokenbasierte Ähnlichkeitsmaße

Die Edit-Distanz ist ungeeignet für Zeichenketten, die aus mehreren Wörtern bestehen, da sie extrem sensibel auf Reihenfolgevertauschungen reagiert. Wörter können aber in vielen Reihenfolgen stehen und trotzdem eine sehr ähnliche Bedeutung haben. Beispielsweise haben „Peter Müller“ und „Müller, Peter“ die hohe Edit-Distanz 13.

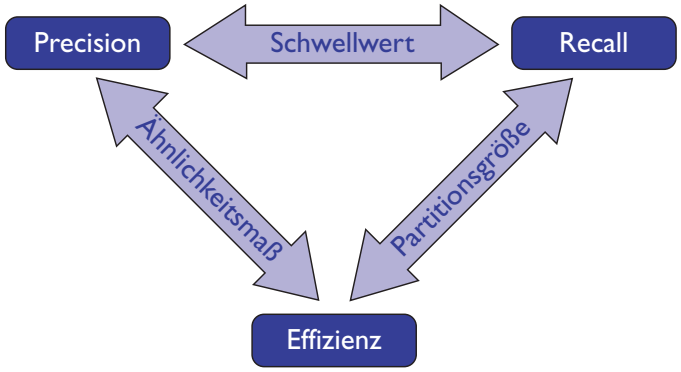
Um solche Effekte zu berücksichtigen, werden die Zeichenketten erst in Wörter bzw. Token zerlegt. Dann wird geprüft, welche und wie viele Token die beiden zu vergleichenden Zeichenketten gemeinsam haben. Tokenbasierte Ähnlichkeitsmaße sind geeignet für textuelle Daten, zum Beispiel Beschreibungen oder Zusammenfassungen, und für Fälle, in denen sich die Struktur der Datensätze nicht erkennen lässt.

Ein viel verwendetes tokenbasiertes Ähnlichkeitsmaß ist die *Jaccard-Ähnlichkeit*. Sie vergleicht die Anzahl der gemeinsamen Token beider Zeichenketten mit der Anzahl aller Token der beiden Zeichenketten. Ein weiteres tokenbasiertes Ähnlichkeitsmaß beruht auf der *term-frequency/inverse-document-frequency (tfidf)*. Sie gibt jedem Token in einer Zeichenkette ein Gewicht, das von seiner Häufigkeit in der Zeichenkette selbst (term-frequency) und seiner Häufigkeit in allen betrachteten Zeichenketten (document-frequency) abhängt. Intuitiv erhalten Token, die oft in der fraglichen Zeichenkette vorkommen, ein hohes Gewicht, und Token, die in fast jeder Zeichenkette vorkommen, ein geringes Gewicht. Darauf aufbauend lässt sich eine Zeichenkette als Vektor der Gewichte aller Token darstellen. Die Ähnlichkeit zweier Zeichenketten kann nun als der Kosinus der beiden Gewichtsvektoren berechnet werden.

Partitionierungsstrategien

Algorithmen zur Duplikaterkennung haben prinzipiell die Aufgabe, ein vorgegebenes Ähnlichkeitsmaß auf alle Paare von Datensätzen anzuwenden und gemäß dem vorgegebenen Schwellwert zu entscheiden, ob es sich um ein Duplikat handelt. Um den Aufwand für die quadratische Zahl von Datensatzpaaren zu vermeiden, werden jedoch nicht alle Paare miteinander verglichen. Stattdessen werden die Datensätze zu-

Zielkonflikte bei der Duplikaterkennung



nächst in Partitionen zerlegt und dann nur innerhalb einer Partition alle Paare verglichen. Eine einfache Methode dazu ist die Partitionierung nach einzelnen Attributen oder Attributteilen. Damit werden allerdings nicht alle Duplikate gefunden: Werden beispielsweise Kundendaten nach dem Wohnort oder dem ersten Buchstaben des Nachnamens partitioniert, werden Duplikate von Kunden, die den Wohnort gewechselt haben oder deren Nachname bereits im ersten Buchstaben einen Tippfehler enthält, nicht gefunden.

Nachfolgend wird beispielhaft die in der Wissenschaft weit verbreitete Sorted-Neighborhood-Methode vorgestellt, die dieses Problem zumindest teilweise vermeidet und dennoch nur eine kurze Laufzeit hat. Der Name „Sorted Neighborhood" bedeutet übersetzt sortierte Nachbarschaft und beschreibt das Vorgehen: Die Datensätze werden zunächst nach einem bestimmten Schlüssel sortiert. Dann werden Datensätze nur noch mit anderen Datensätzen in ihrer „Nachbarschaft" verglichen. Das Vorgehen ist in drei Phasen unterteilt:

1. Schlüsselbildung: In dieser Phase wird von einem Domänenexperten ein Schlüssel als Buchstabensequenz aus relevanten Attributwerten definiert. Ein Schlüssel auf Kundendaten könnte sich zum Beispiel aus den ersten drei Buchstaben des Nachnamens, den ersten zwei Konsonanten des Vornamens und dem Geburtsjahr zusammensetzen. Für jeden Datensatz wird der Schlüssel berechnet und mit einem Verweis auf den ursprünglichen Datensatz gespeichert.

2. Sortierung: In dieser Phase werden alle Schlüssel alphabetisch sortiert. Durch eine geeignete Schlüsselwahl wird erreicht, dass nach der Sortierung ähnliche Datensätze nahe beieinander liegen.

3. Vergleiche: In der letzten Phase wird ein „Fenster" mit einer festen Größe über die sortierten Datensätze geschoben. Dieses Fenster definiert die Nachbarschaft bzw. Partition. Innerhalb dieser Partition werden alle Datensätze paarweise miteinander verglichen und gegebenenfalls als Duplikate gekennzeichnet. Anschließend wird das Fenster um einen Datensatz nach unten verschoben, und wieder werden alle Vergleiche durchgeführt.

Oft wird anschließend die *transitive Hülle* der Duplikate gebildet: Wenn Datensätze A und B sowie B und C als Duplikate erkannt wurden, so bilden A, B und C eine Duplikatgruppe: Sie sind drei Repräsentationen des gleichen Realweltobjekts. So können auch Datensätze als Duplikate erkannt werden, die nicht gemeinsam innerhalb eines Fensters auftauchen.

Der theoretische Aufwand des Algorithmus ist vergleichsweise gering. Eine für die Performanz sehr wichtige Eigenschaft des Algorithmus ist außerdem, dass die erste und dritte Phase nur einen (sehr effizient ausführbaren) linearen Scan der Tabelle erfordern, während die zweite Phase sich auf schnelle Sortierverfahren stützen kann, die in jeder kommerziellen Datenbank zur Verfügung stehen.

Anmerkung

Der vorliegende Text ist adaptiert aus dem Buch „Informationsintegration" von Ulf Leser und Felix Naumann, erschienen 2006 im dpunkt Verlag, Heidelberg.

Verbesserungspotenzial und Datenfusion

Sowohl in der industriellen Entwicklung von Werkzeugen zur Duplikaterkennung als auch in der akademischen Forschung steckt noch großes Verbesserungspotential. Bei den Ähnlichkeitsmaßen sind beispielsweise automatisch lernende Verfahren noch wenig erprobt; bei den Partitionierungsstrategien sind verbesserte inkrementelle Methoden, die nicht den gesamten Datenbestand, sondern lediglich laufende Aktualisierungen berücksichtigen, denkbar. Neue Formen der Daten, insbesondere in XML formatierte Daten, bergen ganz neue Herausforderungen an etablierte Methoden.

Zuletzt soll das an die Duplikaterkennung anschließende Problem, nämlich die *Datenfusion*, genannt werden. Sind mehrere Datensätze als Duplikate erkannt, gilt es nun sie zusammenzuführen. Die einfache Auswahl eines der Datensätze als „survivor" greift hier offensichtlich zu kurz. Vielmehr ist es wünschenswert, aus den einzelnen Werten der Datensätze einen neuen, reicheren Datensatz zu erstellen. ■