Exposé

im Studiengang
Informatik (M.Sc.)

Analyse, Design, Entwicklung und Evaluation eines skalierbaren,
Nahe-Echtzeit Entity Resolution Framework für Streaming-Daten
von
Kevin Sapper

Referent:	Prof. Dr. Adrian Ulges
Korreferent:	Prof. Dr. Reinhold Kröger
Betreuer:	Thomas Strauß (Universum Group)
Unternehmen:	Detim Consulting GmbH / Universum Group

1 Einleitung

Die Masterarbeit soll in Zusammenarbeit mit der Firma Detim Consulting GmbH geschrieben werden. Dazu wird ein Problemfeld bei dem Kunden Universum Group in dessen Geschäftsfeld Inkasso-, Liquiditäts-, und Risikomanagement gewählt.

2 Problemfeld Universum Group

Die Universum Group bietet Lösungen für Onlineshops zur Bonitäts- und Adressprüfung, sowie zum Forderungsankauf der Onlineshop-Kunden. Dabei wird dem Händler bei entsprechender Bonität seines Kunden das Angebot gemacht, die Forderung nach Ablauf einer Zahlungsperiode zu 100 % zu übernehmen. Damit eine möglichst zuverlässige Aussage über die Bonität des Kunden getroffen werden kann, muss zunächst herausgefunden werden, ob der Kunde bereits bei der Universum Group bekannt ist. Das Problem an dieser Stelle ist, dass der Kunde seine Daten Online selbst erfasst und diese nicht anhand von Personalausweis oder ähnlichen Dokumenten überprüft werden können. Fehler bei der Datenerhebung sind beispielsweise unterschiedliche Schreibweisen, insbesondere bei Adressen, Tippfehler, welche bei Namen nicht offensichtlich sind, unterschiedliche Konventionen, etwa Str. für Straße, oder akademische Titel und Adelstitel, welche in Onlineformularen nicht standardisiert erfasst werden. Bei der Bonitätsprüfung dient die Personenidentifizierung dazu, Kunden mit positiver oder negativer Zahlungsmoral zu erkennen und anzunehmen bzw. abzulehnen. Je genauer die Personenidentifikation ist, desto aussagekräftiger sind die Bonitätsauskunfte von externen Dienstleistern, beispielsweise der Schufa. Beim Inkassomanagement gilt zudem das sog. Schadensminderungsprinzip. Das bedeutet, das alle angekauften Forderungen eines Kunden nur einmalig abgemahnt werden dürfen. Daher müssen hier Personendubletten gefunden und zusammengeführt werden.

Das aktuelle System zur Personenidentifizierung funktioniert nur bei der Bonitätsprüfung und ist durch einen externen Dienstleister realisiert. Dieser bereinigt und prüft Namen und Adressen. Allerdings skaliert das System dabei nur innerhalb eines vorgegebenen monatlichen Kontingents.

3 Duplikatserkennung

Die Methoden zur Duplikatserkennung stammen ursprünglich aus dem Gesundheitsbereich (Felegi & Sunter 1969). Je nach Fachgebiet gibt es unterschiedliche Fachbegriffe. Statistiker und Epidemiologen sprechen von *record* oder *data linkage*, während Informatiker das Problem unter *entity resolution*, *data* oder

field matching, *duplicate detection*, *object identification* oder *merge/purge* kennen. Dabei geht es nicht um die reine Personenidentifikation, sondern vielmehr um die Identifikation von Entitäten aller Art, beispielsweise Kunden, Patienten, Produkte oder Orte. Dabei können die Entitäten nicht durch ein einzigartiges Attribut identifiziert werden. Zudem sind die Datensätze oft fehlerhaft, beispielsweise durch Rechtschreibfehler oder unterschiedliche Konventionen. Die Methoden zur Entitätsauflösung arbeiten meist auf Datensatzpaaren und liefern als Ergebnis eine Menge von Übereinstimmungen. Eine Übereinstimmung verknüpft zwei Entitäten. Zusätzlich kann über einen optionalen Ähnlichkeitswert (engl. similarity score), normalerweise zwischen 0 und 1, die Intensität der Übereinstimmung angegeben [1].

Zur Bestimmung der Ähnlichkeit eines Datensatzpaares unterscheiden Elmagarmid et al. [2] zwischen Attributsvergleichs- (engl. field matching) und Datensatzvergleichsmethoden (engl. record matching). Methoden zum Attributsvergleich sind zeichenbasierend (edit distance, affine gap distance, Jaro distance metric oder Q-gram distance), tokenbasierend (atomic strings, Q-grams mit tf.idf [3]), phonetisch (soundex) oder nummerisch. Die Datensatzvergleichsmethoden sind probabilistisch (Naive Bayes), überwachtes bzw. semi-überwachtes Lernen (SVMLight [4], Markov Chain Monte Carlo [5]), aktives Lernen (ALIAS [6]), distanzebasierend (siehe Attributsvergleich - Datensatz als konkatenierter String) oder regelbasierend (AJAX [7]). Die Ausführung der Vergleichsmethoden ist enorm teuer, da diese das Kreuzprodukt zweier Mengen bilden müssen. Um die Ausführungszeit zu reduzieren, wird versucht, den Suchraum auf die wahrscheinlichsten Duplikatsvorkommen zu begrenzen. Diese Vorgehen werden als Blocking oder Indexing bezeichnet. Elmagarmid et al. nennen Standard Blocking, Sorted Neighboorhood Approach, Clustering und Canopies, sowie Set Joins als Vorgehensweisen.

Da es keine Methode zur Entity Resolution gibt, welche allen anderen überlegen ist, wurden Ende der 90er Jahre begonnen, Frameworks zu entwickeln, welche verschiedene Methoden miteinander kombinieren. Einen Vergleich dieser Frameworks wurde durch Köpcke & Rahm 2010 [1] durchgeführt. Ein Framework besteht aus verschiedenen Matchern. Ein Matcher ist dabei ein Algorithmus, welcher die Ähnlichkeit zweier Datensätze ermittelt. Ähnlich wie Elmagarmid et al. unterscheiden Köpcke & Rahm zwischen attributs- und kontextbasierenden Matchern. Als Kontext bezeichnen Sie die semantische Beziehung bzw. Hierarchie zwischen den Attributen. Um die Matcher miteinander zu kombinieren nutzen die Frameworks min. eine Matching Strategie. Eine Strategie ist, die Ähnlichkeitswerte verschiedener Matcher nummerisch zu kombinieren, beispielsweise durch eine gewichtete Summe oder einen gewichteten Durchschnitt. Ein anderer Ansatz ist regelbasierend. Eine einfache Regel besteht aus einer logischen Verbindung und einer Match-Kondition, beispielsweise einem Schwellenwert. Die dritte und komplexeste Strategie ist Workflow-basierend. Hierbei kann beispielsweise eine Sequenz von Matchern die Ergebnisse iterativ einschränken. Grundsätzlich können Workflows beliebig komplex werden. Einen passenden Workflow zu finden, kann selbst Domainexperten vor eine große Herausforderung stellen. Daher

gibt es trainingbasierende Ansätze, passende Parameter für Matcher oder Kombinationsfunktionen (z.B. Gewicht für Matcher) zu bestimmen. Solche Ansätze sind etwa Naive Bayes, Logistic Regression, Support Vector Maschine oder Decision Trees.

Ein Großteil der Forschung in Entity Resolution konzentriert sich auf die Qualität der Vergleichsergebnisse. Die von Köpcke & Rahm verglichenen Frameworks konzentrieren sich allesamt darauf, zwei statische Mengen miteinander zu vergleichen. Bei großen Datenmengen kann dies durchaus mehrere Stunden dauern. Daher gibt es in den letzten Jahren einige Ansätze und Frameworks, welche MapReduce zum Skalieren nutzen [8], [9]. Zudem gibt es immer mehr Bedarf, Vergleichsergebnisse in Nahe-Echtzeit zu liefern. Erste Ergebnisse, Entity Resolution skalierbar und in Nahe-Echtzeit zu erreichen, präsentieren Christen & Gayler in [10] 2008 unter Verwendung von Inverted Indexing Techniken, welche normalerweise bei der Websuche Anwendung finden. Dabei betrachten Sie vor allem die Anforderungen eines Anfragestroms (engl. query stream). Ihre Anforderungen sind, einen Strom von Anfragedatesätzen gegen potentiell riesige Datenmengen im Subsekundenbereich pro Anfrage abzuarbeiten. Dabei sollen die Treffer der Anfrage mit einem Ähnlichkeitswert versehen sein. Zudem muss es möglich sein, die Menge an Anfragen zu skalieren. Das Hauptproblem ist hierbei die Skalierung. Um skalieren zu können, wird versucht, die Abarbeitung des Suchraums zu parallelisieren. Eine Studie von Kwon, Balazinska, Howe, & Rolia [11] in MapReduce Anwendungen zeigt, das selbst geringe Ungleichgewichte bei der Verteilung des Suchraum auf Mapper bzw. Reducer, aufgrund der Komplexität der Matching Algorithmen zu deutlich längeren Laufzeiten und damit Gesamtlaufzeiten führt. In einem ihrer Beispiele sind bei einer Gesamtzeit von 5 Minuten die meisten Mapper innerhalb von 30 Sekunden fertig. Auch beim Streaming kann diese sog. Datenschiefe (engl. data skew) den Durchsatz eines Clusters signifikant mindern. Einen weiteren Ansatz, die Laufzeit für Nahe-Echtzeit Anwendungen zu optimieren, präsentieren Whang et al. [12]. Anstatt eine Ergebnismenge nach Abschluss eines Algorithmus zu liefern, zeigen Sie Möglichkeiten, partielle Ergebnisse während der Laufzeit des Algorithmus zu erhalten.

4 Zielsetzung

Im Rahmen der Thesis soll ein Entity Resolution Framework für Datensatzströme entstehen. Als Basis soll ein Stream Processing Framework, beispielsweise aus der Apache Familie (Flink, Samza, Spark, Storm, etc.) oder ein (Complex) Event Processing Framework, beispielsweise Esper, genutzt werden. Das Framework soll eine Reihe von Matchern sowie Kombinationsfunktionen der Matcher unterstützen. Die Implementierung der Matcher soll größtenteils aus Standard-Bibliotheken erfolgen. Auch bei den Maschine Learning Kombinationsfunktionen soll weitestgehend existierende Lösungen, etwa WEKA [13], dass auch im dedoop Framework [8] Anwendung findet, genutzt werden. Das Hauptaugen-

merk der Thesis hingegen soll die Skalierbarkeit sein. Dabei muss ein existierender Datenbestand zunächst so geclustert werden, dass die Entity Resolution in jedem Cluster möglichst gleich lange dauert. Bei MapReduce Anwendungen kann auf Basis der bekannten Datenmengen die Verteilung der Entitäten, sowie die Größe und Anzahl der Cluster berechnet werden. Dadurch wird das Data Skew Problem minimiert, sodass alle Mapper bzw. Reducer eines MapReduce-Jobs eine ähnliche Laufzeit haben. Beim Streaming hingegen kommen neue unbekannte Datensätze in das System. Ein Cluster kann daher zunächst nur auf Basis der existierenden Datensätze gebildet werden und ist abhängig von der gewählten Blocking Strategie. Damit die Latenzen möglichst gering bleiben, sollten die einzelnen Cluster eine bestimmte Größe nicht überschreiten. Um den Durchsatz innerhalb des Systems zu optimieren muss ein Load-Balancing durchgeführt werden. Dadurch soll verhindert werden das Backpressure auftritt und schlimmstenfalls Datensätze verloren gehen. Eine weitere Schwierigkeit ist, dass die Datenmenge nicht statisch ist, sondern neue Datensätze jederzeit hinzukommen können. Dabei muss betrachtet werden, ob die Blocking Strategie bzw. das Framework es zulässt das Cluster während der Laufzeit zu verändern. Zudem kann das Clustering bzw. das Load-Balancing auch statistische Daten heranziehen, beispielsweise das bei einer weltweiten Personenidentifikation tageszeitabhängig mehr Anfragen aus Europa, Asien oder Amerika kommen. Des Weiteren muss das Framework mindestens zwei Blocking Strategien implementieren, damit möglichst viele Duplikate gefunden werden können.

Während der Analyse sollen typische Streamingszenarien betrachtet, etwa ein kontinuierlicher Datenstrom kleiner Datensätze mit zwischenzeitlichen Peaks unterschiedlicher Länge. Unbekannte Entitäten, d. h. Entitäten für welche die Ergebnismenge der Entity Resolution leer ist, sollen als neue Entitäten in den Datenbestand aufgenommen werden. Ein interessantes Szenario diesbezüglich kann sein, kleiner Datenbestand mit vielen neuen Entitäten, da hier das System potentiell die schwierigste Phase durchlaufen muss.

Idealerweise soll der Durchsatz sowie die Qualität der Suchergebnisse mit bereits bekannten Veröffentlichungen verglichen werden. Da diese meist auf statischen Daten arbeiten, müssen die zu prüfenden Datenmengen, für einen Vergleich, künstlich in das System gestreamt werden.

5 Methoden

Zur Umsetzung der in Abschnitt 4 beschriebenen Ziele muss zunächst eine Wissensbasis durch Literaturarbeit in folgenden Grundlagen geschaffen werden:

- Matcher-Algorithmen zur Entity Resolution
- Kombinationsfunktionen zur Entity Resolution
- Blocking und Indexing Strategien für Entity Resolution
- Clusteringmethoden für Blocking und Indexing

- Data Skew bei verteilten und parallelen Anwendungen
- Load-Balancing für Streaminganwendungen
- Entity Resolution Frameworks - traditionell, MapReduce, Streaming
- Streaming Processing Frameworks/(Complex) Event Processing Frameworks

Weitere Methoden sind:

- UML-Entwurf
- Proof of Concept
- Funktionelle Leistungsbewertung anhand von Datensätzen in wissenschaftlichen Publikationen

6 Erwartete Ergebnisse

Die erwarteten Ergebnisse der Masterarbeit sind:

- Analyse von Entity Resolution Matchern
- Analyse von Entity Resolution Kombinationsfunktionen
- Analyse von Clusteringmethoden für Blocking und Indexing
- Analyse von Entity Resolution Frameworks
- Analyse von Stream Processing Frameworks und ggf. (Complex) Event Processing Frameworks
- Analyse der Data Skew und Load-Balancing Problematik für Entity Resolution
- Design eines Entity Resolution Streaming Framework
- Prototyp der wesentlichen Funktionen
- Evaluation des Prototypen gegen öffentliche Datensätze existierender Veröffentlichungen

7 Vorbedingungen

- Datensätze zum Evaluieren und Trainieren des Frameworks bzw. der Algorithmen [14–16]

Literatur

[1] Köpcke, Hanna ; Rahm, Erhard: Frameworks for Entity Matching: A Comparison. In: *Data & Knowledge Engineering* Bd. 69 (2010), Nr. 2, S. 197–210

[2] Elmagarmid, A. K. ; Ipeirotis, P. G. ; Verykios, V. S.: Duplicate Record Detection: A Survey. In: *IEEE Transactions on Knowledge and Data*

Engineering Bd. 19 (2007), Nr. 1, S. 1–16

[3] Gravano, Luis ; Ipeirotis, Panagiotis G. ; Koudas, Nick ; Srivastava, Divesh: Text Joins in an RDBMS for Web Data Integration. In: *Proceedings of the 12th International Conference on World Wide Web* : ACM, 2003, S. 90–101

[4] Joachims, Thorsten: Svmlight: Support Vector Machine. In: *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund* Bd. 19 (1999), Nr. 4

[5] Gilks, Walter R. ; Richardson, Sylvia ; Spiegelhalter, David J.: Introducing Markov Chain Monte Carlo. In: *Markov chain Monte Carlo in practice* Bd. 1 (1996), S. 19

[6] Sarawagi, Sunita ; Bhamidipaty, Anuradha: Interactive Deduplication Using Active Learning. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* : ACM, 2002, S. 269–278

[7] Galhardas, Helena ; Florescu, Daniela ; Shasha, Dennis ; Simon, Eric ; Saita, Cristian: *Declarative Data Cleaning : Language, Model, and Algorithms* (report) : INRIA, 2001 – Maßstab

[8] Kolb, Lars ; Rahm, Erhard: Parallel Entity Resolution with Dedoop. In: *Datenbank-Spektrum* Bd. 13 (2013), Nr. 1, S. 23–32

[9] Malhotra, Pankaj ; Agarwal, Puneet ; Shroff, Gautam: Graph-Parallel Entity Resolution Using LSH & IMM. In: *EDBT/ICDT Workshops*, 2014, S. 41–49

[10] Christen, Peter ; Gayler, Ross: Towards Scalable Real-Time Entity Resolution Using a Similarity-Aware Inverted Index Approach. In: *Proceedings of the 7th Australasian Data Mining Conference - Volume 87, AusDM '08*. Darlinghurst, Australia, Australia : Australian Computer Society, Inc., 2008 — ISBN 978-1-920682-68-2, S. 51–60

[11] Kwon, YongChul ; Balazinska, Magdalena ; Howe, Bill ; Rolia, Jerome: A Study of Skew in Mapreduce Applications. In: *Open Cirrus Summit* (2011)

[12] Whang, S. E. ; Marmaros, D. ; Garcia-Molina, H.: Pay-As-You-Go Entity Resolution. In: *IEEE Transactions on Knowledge and Data Engineering* Bd. 25 (2013), Nr. 5, S. 1111–1124

[13] Hall, Mark ; Frank, Eibe ; Holmes, Geoffrey ; Pfahringer, Bernhard ; Reutemann, Peter ; Witten, Ian H.: The WEKA Data Mining Software: An Update. In: *ACM SIGKDD explorations newsletter* Bd. 11 (2009), Nr. 1, S. 10–18

[14] Hernández, Mauricio ; Cohen, William ; Tejada, Sheila ; Lawrence, Steve: Duplicate Detection, Record Linkage, and Identity Uncer-

tainty: Datasets.

[15] Köpcke, Hanna ; Thor, Andreas ; Rahm, Erhard: Evaluation of Entity Resolution Approaches on Real-World Match Problems. In: *Proceedings of the VLDB Endowment* Bd. 3 (2010), Nr. 1-2, S. 484–493

[16] Draisbach, Uwe ; Naumann, Felix: DuDe: The Duplicate Detection Toolkit. In: *ResearchGate* (2010)