KOBRA: Praxisfähige lernbasierte Verfahren zur automatischen Konfiguration von Business-Regeln in Duplikaterkennungssystemen

Simone Braun [1,2] Georges Alkhouri[3] Eric Peukert[4]

Abstract: Duplikaterkennung, -suche und -konsolidierung für Kunden- und Geschäftspartnerdaten, sog. „Identity Resolution", ist die Voraussetzung für erfolgreiches Customer Relationship Management und Customer Experience Management, aber auch für das Risikomanagement zur Minimierung von Betrugsrisiken und Einhaltung regulatorischer Vorschriften und viele weitere Anwendungsfälle. Diese Systeme sind jedoch hochkomplex und müssen individuell an die kundenspezifischen Anforderungen angepasst werden. Der Einsatz lernbasierter Verfahren bietet großes Potenzial zur automatisierten Anpassung. In diesem Beitrag präsentieren wir für ein KMU praxisfähige, lernbasierte Verfahren zur automatischen Konfiguration von Business-Regeln in Duplikaterkennungssystemen. Dabei wurden für Fachanwender Möglichkeiten entwickelt, um beispielgetrieben das Match-System an individuelle Business-Regeln (u.a. Umzugserkennung, Sperrlistenabgleich) anzupassen und zu konfigurieren. Die entwickelten Verfahren wurden evaluiert und in einer prototypischen Lösung integriert. Wir konnten zeigen, dass unser Machine-Learning-Verfahren, die von einem Domainexperten erstellten Business-Regeln für das Duplikaterkennungssystem „identity" verbessern konnte. Zudem konnte der hierzu erforderliche Zeitaufwand verkürzt werden.

Keywords: Sequential Model-Based Optimization; Nonlinear Regression; Reinforcement Learning; Entity Resolution; Identity Resolution

1 Einleitung

Ein wesentlicher Anteil des Aufwands in IT-Großprojekten fließt in die Migration von Daten von Alt- auf Neusysteme und die damit verbundene Integration verschiedener Datenquellen sowie deren Bereinigung. Fehler in der Migration sorgen für kritische Fehlfunktionen und Dateninkonsistenzen in neu entwickelten Systemen, die häufig erst zu einem späten Projektzeitpunkt entdeckt werden. Beispielsweise können falsche oder fehlende Kundendaten zu irrtümlich versendeter Werbung, falschen Rechnungen und Fehlbuchungen führen, die das Kundenverhältnis nachhaltig schädigen. Die Korrektur solcher Migrationsfehler sprengt meist den zeitlichen und finanziellen Rahmen des gesamten Großprojekts und führt

[1] UNISERV GmbH, Business Development, Rastatter Str. 13, Pforzheim, 75179, simone.braun@uniserv.com, https://orcid.org/0000-0002-4825-1648
[2] Hochschule Offenburg, Fakultät B+W, Klosterstraße 14, Gengenbach, 77723, simone.braun@hs-offenburg.de
[3] Institut für Angewandte Informatik e.V., Datenbanken, Goerdelerring 9, Leipzig, 04109, georges.alkhouri@gmail.com
[4] Institut für Angewandte Informatik e.V., Datenbanken, Goerdelerring 9, Leipzig, 04109, peukert@infai.org

doi:10.18420/inf2020_37

letztendlich zum Scheitern. Selbst wenn die Migration und das Gesamtprojekt erfolgreich waren, können auch in modernen IT-Systemen Fehl- oder Doppeleingaben oder veraltete Kundenstammdaten nicht ausgeschlossen werden.

Abhilfe schaffen hier hochleistungsfähige Datenmigrations- und Integrationswerkzeuge wie sie z.B. vom mittelständischen Unternehmen UNISERV zur Duplikaterkennung und -suche für Geschäftspartnerdaten (sog. „Identity Resolution“) angeboten werden. Identity Resolution ist die Voraussetzung für erfolgreiches Customer-Relationship-Management und Customer-Experience-Management, aber auch für das Risiko-management zur Minimierung von Betrugsrisiken und Einhaltung regulatorischer Vorschriften (z.B. Embargoverordnungen). Heute verstehen wir darunter nicht mehr allein Dublettenbereinigung im Rahmen von Datenqualitätsmanagement. Die Möglichkeit alle Informationen aus den unterschiedlichsten Kanälen, Systemen und Devices über Interessenten und Kunden zur 360°-Sicht in den Gesamtkontext zu stellen, bringt großen Mehrgewinn, z.B. besseren Einblick in Kundenbedürfnisse.

Der Bereich Duplikaterkennung ist an sich intensiv erforscht. Existierende Systeme sind in der Lage, Duplikate in Datenbeständen weitgehend automatisch und effizient zu identifizieren. Nur für kleine Mengen von Datensätzen sollen Nutzer noch einen manuellen Abgleich vornehmen müssen. Leider zeigt sich, dass Kunden sehr unterschiedliche und spezifische Anforderungen an die Duplikatidentifikation haben, so dass für jeden Anwendungsfall das Matching-System manuell neu konfiguriert werden muss. Typische Szenarien sind Haushaltsabgleiche, Umzugserkennung, Sperrlistenabgleiche etc., welche wiederum abhängig vom Geschäftsszenario in ihren Anforderungen variieren (z.B. Konsolidierung aller Finanz- und Versicherungsverträge in einem Haushalt, Vermeidung von Mehrfachzustellung einer Informationsbroschüre pro Haushalt oder Datenanreicherung und Personalisierung für Marketing-Kampagnen) und die jeweils ein unterschiedliches Verhalten der Duplikaterkennung erfordern. Bisher wird dies durch wenige Experten in einem zeitlich aufwendigen, iterativen Prozess gemeinsam mit den Fachanwendern beim Kunden umgesetzt (s.a. Kapitel 2). Dieser Anpassungsaufwand ist sehr kostenintensiv und benötigt lange und intensive Testphasen. In sehr großen Projekten mit komplexen Anwendungsfällen, wie bspw. im Finanz- und Versicherungsbereich, kann dieser Prozess gerne bis zu 50 Personentage in Anspruch nehmen; zeigt die Erfahrung von UNISERV. Diese Zeit steht jedoch oft nicht zur Verfügung, so dass neue Lösungen zur Automatisierung der manuellen Konfiguration von Duplikaterkennungssystemen gesucht sind, (a) unter Berücksichtigung kunden-individueller Anforderungen an die Duplikatidentifikation sowie (b) der praxisfähigen Anwendung und Einsatzes für ein mittelständisches Unternehmen wie UNISERV.

Herkömmliche Regel- und lernbasierte Duplikaterkennungen vergleichen sich mit einem Goldstandard von bekannten Ergebnissen um einen Qualitätswert zu erhalten. In der Arbeit von [KR08] wurde vor allem am Problem der Erstellung guter Trainingsdaten gearbeitet. Durch die gezielte Auswahl aussagekräftiger Paare konnte der Aufwand des manuellen Matchings für den Nutzer signifikant reduziert werden. Das Fever-System [KTR09, KR10] vergleicht verschiedene etablierte Lernverfahren im Objekt-Matching-Umfeld hinsicht-

lich des Parametrierungsaufwands, der erforderlichen Menge an Trainingsdaten und der erreichbaren Qualität anhand eines Webdaten-Matching-Problems. Hier konnte gezeigt werden, dass manche Lernverfahren bereits mit kleineren Mengen an Trainingsdaten gute Matching-Ergebnisse erzielen können. Zudem konnte für lernbasierte Ansätze insbesondere für schwierige Anwendungsfälle wie das Matching von Produktangeboten aus Web-Shops [Köp12] eine qualitativ bessere Dublettenerkennung als mit herkömmlichen manuell einzustellenden Ansätzen nachgewiesen werden. Jedoch ist auch hier die Menge an manuell zu labelnden Daten für den realen Einsatz mit ca. 500 Labeln sehr groß.

Großes Potenzial zur Adressierung des geschilderten Konfigurationsproblems und damit zur Reduktion des manuellen Labelaufwands versprechen lernbasierte Optimierungsansätze. Das algorithmische Framework der Sequential Model Based Optimization (SMBO), bietet einen globalen Optimierungsansatz für unterschiedlichste Lernmodelle, der sich für teure Blackbox-Funktionen als effektiv und dateneffizient erwiesen hat [Sha16]. Das Feld selbst erlebt eine Renaissance in der Anwendung zur automatischen Konfiguration von Algorithmen und zur Hyperparameteroptimierung im Bereich des maschinellen Lernens [Yao18, Sha16, HHL11, Feu15, Men16, ZL16]. Vermehrt Interesse an lernbasierten Optimierern gibt es auch im Bereich des Meta-Learning auf Basis von Recurrent Neuronal Networks (RNN). [TV20] entwickelten mit RNN-Opt einen Ansatz zur Blackbox Optimierung, wenn kein Gradient gebildet werden kann. Des Weiteren wurde Reinforcement Learning für die automatische Konfiguration virtueller Maschinen von [Rao09] genutzt. Bis zum jetzigen Zeitpunkt wurden nach unserem Wissensstand allerdings die Verfahren nicht in der realen Anwendung von KMU oder mit großen Datenmengen in der Domäne der Kunden- und Geschäftspartnerdaten evaluiert.

Im Folgenden gehen wir detailliert auf die Ausgangssituation und daraus abgeleitete Anforderungen und Zielstellung ein. In Kapitel 3 präsentieren wir die KOBRA-Lösung zur automatischen Konfiguration von Business-Regeln in Duplikaterkennungssystemen mit lernbasierter Unterstützung sowie in Kapitel 4 die Lernkomponente und implementierten Lern- und Optimierungsverfahren, bevor wir im Anschluss (Kapitel 5) auf Evaluation und Test der Verfahren und insbesondere die Praxisfähigkeit für ein KMU wie UNISERV eingehen. In Kapitel 6 diskutieren wir die Ergebnisse der Evaluation der Gesamtlösung am praktischen Anwendungsfall, bevor wir unseren Beitrag mit einer Zusammenfassung und einem Ausblick abschließen.

2 Ausgangssituation

Wann im Bereich der Kunden- und Geschäftspartnerdaten eine Dublette als eine Dublette erkannt werden soll, hängt stark vom kundenindividuellen Geschäftsszenario und ihren variierenden Anforderungen ab. Zum Beispiel kann es sich bei den Datensätzen „G. Mayer, Pforzheimer Str. 320, 70499 Stuttgart“ und „Gerhard Meier, Pforzheimer Str. 320, 70499 Stuttgart“ um dieselbe Person mit Tippfehler bei der Namenseingabe handeln oder um zwei getrennte Identitäten. Mag eine Zusammenführung der Datensätze bei

einer Datenanreicherung für eine Marketingkampagne weniger kritisch sein, so wäre eine Konsolidierung für die Zusammenführung von Bankkonten ggf. fatal.

Bei der Konfiguration der Business-Regeln (auch Matching-Regeln) in Duplikaterkennungssystemen wie dem UNISERV „identity“ ist nun kundenindividuell einzustellen, ob und an welcher Stelle ein strenger Abgleich, (zeichengenaue Übereinstimmung „Gerhard Mayer-Vorfelder“ | „Gerhard Mayer-Vorfelder“) oder toleranter Abgleich („Gerhard Mayer-Vorfelder“ | „Gerhart Meier-Forvelter“ | „Erhart Nayer-Vorfeider“ | „Gero MV“ | „Герхард Маыер-Ворфелдер“) oder auch Vergleiche auf Haushaltsebene (Erkennen der Zusammengehörigkeit von „Gerhard Mayer-Vorfelder“ und „Margit Mayer-Vorfelder“) u.v.m. erfolgen soll; wobei der Anteil an „false-positive“ und „false-negative“ Treffern gering zu halten ist um manuelle Nachbearbeitungen zu vermeiden.

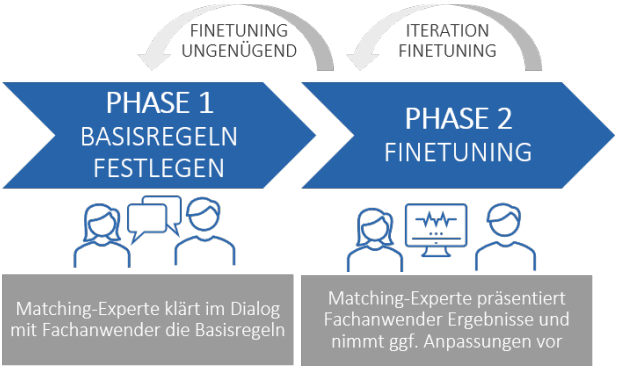

Abb. 1: Bisherige Vorgehensweise Erstellung der Matching-Regeln

Die aktuell übliche Vorgehensweise zur Regelkonfiguration im Matching-System „identity“ von UNISERV ist wie folgt (s. Abbildung 1): Der Kunde stellt einen repräsentativen Datenauszug bereit und erläutert UNISERV Consultants und Matching-Experten den Business Case, Ursprung und Inhalt der Daten (z.B. überwiegend natürliche oder juristische Personen) sowie weitere Kriterien (z.B. Existenz von Pflichteingabefeldern oder Toleranzen). Es erfolgt eine erste gemeinsame Analyse. Basierend auf Erkenntnissen aus dem Gespräch wählen die Matching-Experten aus einer Reihe von Standard-Konfigurationen Basisregeln für das Matching-System aus. Das Matching-System wird ausgeführt und das Ergebnis wird mit den Fachanwendern diskutiert und gemeinsam entschieden, ob das grundsätzliche Ziel erreicht wurde. Falls nicht, werden neue Basisregeln gewählt und der Vorgang wiederholt. Ansonsten erfolgt im nächsten Schritt die Feinjustierung anhand ausgewählter Beispiele (Markierung von false-positive und false-negative) bzw. neuer Beispiele seitens der Fachanwender. So können Fachanwender anhand ihres Anwendungsfalls z.B. folgende Situationen entscheiden: (1) Input: „G. Mayer, Pforzheimer Str. 320, 70499 Stuttgart“ und Possible Match: „Gerhard Meier, Pforzheimer Straße 320, 70499 Stuttgart“ → diese Art von Datensätze sollen als Dublette identifiziert werden, d.h. wären true-positive; (2) Input: „G. Mayer, Pforzheimer Str. 320, 70499 Stuttgart“ und Possible Match: „IT-Mayer GmbH, Rastatterstr. 320 70499 Stuttgart“ → diese Art von Datensätze sollen nicht als Dublette identifiziert werden, d.h.

wären false-positive. In diesem Prozess versuchen Consultants bzw. Matching-Experten insbesondere Widersprüche in den Entscheidungen der Fachanwender aufzulösen. Die Matching-Regeln werden auf Basis des Feedbacks der Fachanwender angepasst, das System ausgeführt und das Ergebnis erneut mit den Fachanwendern diskutiert. Dies wird so lange wiederholt bis ein zufriedenstellendes Ergebnis erzielt ist, was zeitlich sehr aufwendig sein kann.

Bisher ist die Erstellung der Matching-Regeln komplex und erfordert tiefes technisches Wissen und Erfahrung. Abbildung 7 zeigt einen Ausschnitt der Konfigurationsmatrix, die das Matching-Verhalten und die Bewertung sämtlicher Elemente steuert[5]. Über die Parameterwerte kann ausgedrückt werden, dass der Name tolerant sein kann, wenn die Adresse identisch ist und umgekehrt. Oder die Hausnummer darf verschieden sein, wenn Name und Adresse identisch sind. Aber auch wie eine phonetische Ähnlichkeit, zum selben Ort gehörige PLZ oder Hausnummernnachbarschaft zu bewerten sind sowie erforderliche Mindestübereinstimmung, Elementreihenfolge beim Abgleich, weitere Ähnlichkeitsgrade wie Größe von Wortabständen, Synonyme, Initialen oder Umgang mit leeren Feldern, Wortreihenfolge („Meyer-Vorfelder“ | „Vorfelder-Meyer“) oder Überhängen („Meyer“ | „Meyer-Vorfelder“), Berücksichtigung des Geschlechts beim Vornamen, Vertauschen von Tag und Monat im Geburtsdatum u.v.m.

So wird z.B. beim Abgleich der Adresse für jeden Adresselementtyp (wie PLZ, Ortsname, Straßenname, Hausnummer, Name, Vorname) und für jedes Vergleichsergebnis (identisch, phonetisch ähnlich, ähnlich, verschieden usw.) ein Parameterwert zwischen -100 und 100 festgelegt. Für jedes Wort der Eingabe wird der Grad der Übereinstimmung mit dem entsprechenden Begriff der Referenz ermittelt. Aus diesem Ergebnis ergibt sich im Zusammenhang mit dem Parameterwert der Wert, mit dem das Wort in den Gesamtwert eingeht. Die Gesamtbewertung der Adresse ist z.B. der Durchschnitt der für die Begriffe ermittelten Werte unter Einbeziehung der Gewichtung je Adresselementtyp. Nur Adressreferenzen, deren Gesamtbewertung über einem in der Abfrage angegeben Grenzwert liegen, werden als potenzielle Dubletten zurückgeliefert.

Die notwendige Genauigkeit einer Konfiguration ist hierbei immer vom konkreten Anwendungsfall abhängig. Werden bspw. bei Embargo-Prüfungen Datensätze (Personen) nicht erkannt, kann dies empfindliche Strafen zur Folge haben. Auf der anderen Seite verursacht jeder Verdachtsfall Kosten, da diese manuell überprüft werden müssen. Die Kosten nicht-erkannter Datensätze überwiegen hier jedoch, so dass false-negatives soweit wie möglich vermieden werden sollten, während die Kosten von false-positives weniger kritisch zu betrachten sind und im Zweifel lieber ein Treffer zu viel als zu wenig angezeigt werden sollte. Damit soll z.B. bei Prüfung gegen eine PEP-Liste („Politically Exposed Persons“) ein Treffer vorliegen, wenn eine der Personen im Namensfeld mit einem Eintrag in der PEP-Liste übereinstimmt, andere Attribute können dabei deutlich weniger Gewicht erhalten.

[5] Auf genaue Details der Matrix kann an dieser Stelle aus Gründen der KnowHow-Wahrung nicht weiter eingegangen werden.

Ein anderes Beispiel stellt die Pflege einer Adressdatenbank dar: Soll eine Datenbank von Dubletten bereinigt werden, um bspw. alle Verträge einer Kundin in einem Online-Portal zusammenführen zu können, wird man die gegenteilige Strategie wählen. Falls im Fehlerfall eine Endkundin die Daten einer dritten Person einsehen könnte, wäre der Schaden ungleich höher, als wenn diese fälschlicherweise zwei Logins bekäme. False-positives sind hier kritischer zu sehen und die Konfiguration sollte sich in Zweifelsfällen gegen eine automatische Dubletten-Zusammenführung entscheiden.

Bei der Erstellung der Business-Regeln für eine Matching-Verarbeitung geht es in einigen Fällen nicht nur um doppelte oder mehrfach vorhandene Stammdaten, sondern oft auch um zugehörige Transaktions- oder Stammdaten aus anderer Quellen, die über "weiche" Identifikationskriterien wie Namen + Adressen, Namen + Geburtsdatum, Telefon-Nr., E-Mailadresse oder Social Login zugeordnet werden müssen. Somit kann es sein, dass die Abweichungstoleranz für einige Attribute sehr klein sein muss und für andere wiederum größer sein darf. Das heißt, das System ist so zu parametrieren, dass Datensätze auch dann als Treffer ausgewiesen werden, wenn diese sich in bestimmten Feldern unterscheiden.

Zusammenfassend lassen sich folgende Kernanforderungen ableiten:

- Die Entwicklung optimaler Regeln soll beschleunigt werden.
- Fachanwender sollen den Konfigurationsprozess selbstständig und an ihren individuellen Business Case anpassbar durchführen können.
- Fachanwender sollen keine zusätzlichen Kompetenzen im Bereich Konfiguration von Matching-Regeln aufbauen müssen.
- Anhand von Beispielen sollen Fachanwender eine passende Konfiguration automatisch ableiten lassen können.

3 Lösung

Ziel ist es, die Entwicklung von optimalen Matching-Regeln gegenüber dem bisherigen Vorgehen deutlich zu beschleunigen und zu vereinfachen. Fachendanwender des UNISERV Matchingsystems „identity“ sollen in die Lage versetzt werden möglichst optimale Matching-Regeln ohne die Unterstützung von Machting-Experten mit möglichst geringem Zeitaufwand und Know-How-Aufbau zu erstellen (vgl. Abbildung 2).

Die Grundidee ist, dass Fachanwender das System für ihren individuellen Business Case trainieren, indem sie manuell gelabelte Trainingsdaten zur Anreicherung eines vortrainierten Modells zur Verfügung stellen. Ein dadurch verbessertes Vorhersagemodell tritt an die Stelle des Consultants bzw. Matching-Experten und optimiert mithilfe der gelabelten Trainingsdaten die Konfiguration des Matching-Systems.

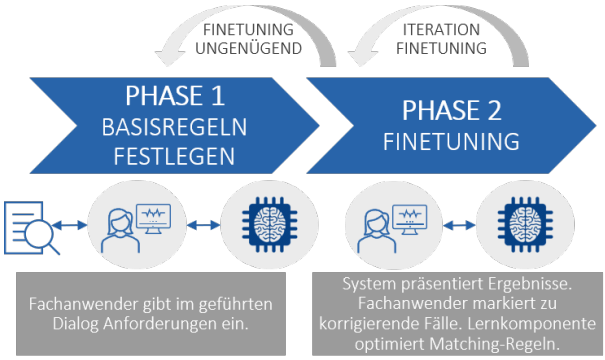

Abb. 2: Erstellung optimaler Matching-Regeln mit lernbasierter Unterstützung

Damit ein Projekt wirtschaftlich ist, muss die Zeit für das manuelle Labeling der Daten deutlich geringer sein als eine direkte Konfiguration zu erstellen. Hierzu ist die Extraktion einer repräsentativen Trainingsmenge notwendig. In der Regel unterscheidet sich eine gute Konfiguration von einer schlechten nicht im Bereich von sicheren Dubletten und auch nicht im Bereich von sicheren Nicht-Dubletten, sondern anhand der unsicheren Dubletten.

Um die Anforderungen und das gesteckte Ziel zu erfüllen, wurde die KOBRA-Lösung entwickelt. Sie besteht im Wesentlichen aus vier Komponenten: (1) Webapplikation für den Anwenderdialog, (2) Backend für die Datenverwaltung, (3) Matching-System auf Basis des UNISERV-Produkts „identity“ zur Durchführung der Duplikaterkennung und (4) Lernbasierte Komponente, die mittels sequentieller Optimierung und auf Grundlage der manuell gelabelten Daten ein Vorhersagemodell trainiert, um eine optimale Konfiguration der Matching-Regeln zu erstellen.

In einem geführten Prozess werden zunächst die Anforderungen ermittelt. Über die Webapplikation erstellen Fachanwender ein neues Projekt und beantworten im Dialog Fragen bzgl. Anwendungsfall oder spezieller Kriterien (z.B. phonetische Toleranz, Verwerfen eines Matches bei Differenz etc.). Parallel dazu erfolgt die Analyse der bereitgestellten Daten bzgl. Herkunftsland, Sprachraum, Befüllungsgrad, Qualität etc. Anhand der Eingaben und Datenanalyse werden die Basiskonfigurationen generiert.

Darauf erfolgt auf Grundlage der ermittelten Basiskonfigurationen die Bildung und Selektion von Dublettengruppen (es werden die Teilergebnisse selektiert, welche nicht identisch zu den Teilergebnissen anderer Konfigurationen sind). Die Gruppen werden den Fachanwendern zur Qualifizierung in (a) true-positive – gewünschter Treffer und (b) false-positive – unerwünschter Treffer angezeigt (s. Abbildung 3). Die Fachanwender bewerten die Dublettengruppen, indem sie eine Kopfdublette[6] auswählen und jeden weiteren Datensatz

[6]Bei einer Konsolidierung werden verschiedene Datensätze, die sich auf dieselbe Real-Welt-Entität beziehen, in einer Dublettengruppe mit einem Kopfdatensatz (der mit den meisten Informationen) zusammengefasst. Dieser kann mit weiteren Informationen aus den gefundenen Dubletten-Datensätzen ergänzt werden.

entsprechend qualifizieren, ob dieser richtig oder falsch als Mitglied dieser Dublettengruppe erkannt wurde. Das Ergebnis ist eine qualifiziert bewertete Trefferliste mit gewünschten und unerwünschten Dubletten.

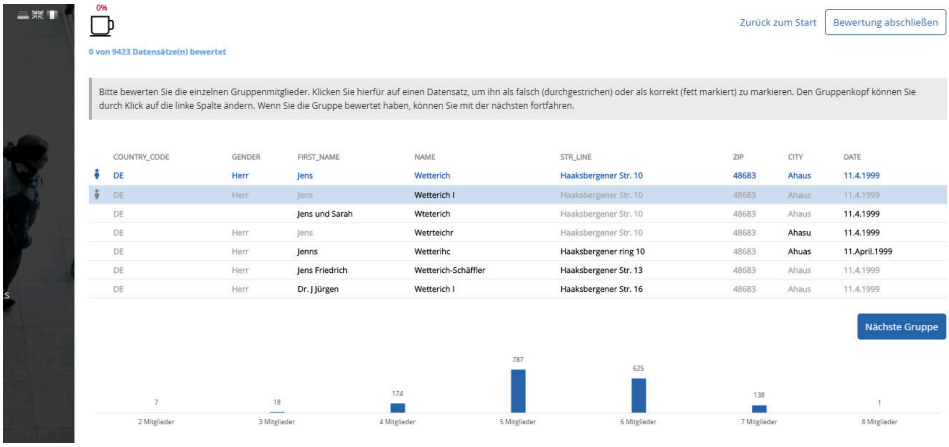

Abb. 3: Bewertung der Dublettengruppen

Mit dem entwickelten Goldstandard erfolgt zuletzt die Ermittlung einer optimalen Konfiguration mit Hilfe der Lernkomponente. Ziel ist es eine möglichst gute Konfiguration mit den vorhandenen Ressourcen zu ermitteln. Dabei werden wiederholt verschiedene Konfigurationen getestet und durch das Vorhersagemodell ausgewählt und mögliche Verbesserungen anhand des Goldstandards bewertet.

Im Allgemeinen können nicht alle Anforderungen des Goldstandards erfüllt werden und somit ist dieses Problem nicht fehlerfrei lösbar. Um eine Bewertung zu ermöglichen, wird ein Abstandsmaß zwischen dem Ergebnis der automatisch erstellten Konfiguration und dem Goldstandard des Fachanwenders berechnet. Die verwendete Metrik liefert verschiedene Kenngrößen zur Qualitätsbewertung sowie einen einzelnen Qualitäts-Score, welcher im Idealfall den Wert 100 oder den normierten Wert 1.0 annimmt.

4 Lernkomponente

Das Finden einer geeigneten Konfiguration ist gleich zu setzen mit einem hohen Qualitäts-Score, wie dem F-Score. Somit lässt sich ein sinnvolles Lernproblem aus der obigen Problemstellung als ein Regressionsproblem formulieren. Hierbei fungieren die Konfigurationsparameter als unabhängige Variablen mit Hilfe derer das Regressionsmodell den Qualitäts-Score ableitet. Intuitiv soll das Modell Wertebereiche im Konfigurationsraum finden, welche eine hohe Qualität besitzen und somit eine gute Duplikaterkennung ermöglichen sollen. Anschließend dienen die gefundenen Werte als Konfigurationsparameter für das Matching-System. Um erfolgreich ein Regressionsmodell zu trainieren, werden gelabelte

Daten in Form von Konfigurationen $\boldsymbol{x}$, mit resultierendem Qualitätswert $\boldsymbol{f(x)}$ benötigt, da es sich um einen Supervised-Lernalgorithmus handelt. Wie in Kapitel 1 erwähnt, sind solche Datenpaare in einem nicht ausreichenden Maße vorhanden. Wir lösen dieses Problem, indem wir Trainingspaare durch die Umgebung selbst generieren und als algorithmisches Framework die Sequential Model Based Optimization [HHL11] nutzen. Im Folgenden gehen wir näher auf das SMBO Interface ein.

Abbildung 4 zeigt vereinfacht die entwickelte Lernkomponente als SMBO Interface. Zu Beginn der Iteration wählt ein Modell $\boldsymbol{M}$ eine zu testende Konfiguration $\boldsymbol{x}$ aus (3). Dies geschieht mithilfe einer Acquisition-Funktion, welche bestimmt, wie der zugrunde liegende Konfigurationsraum exploriert werden soll und welche Parameter der Konfiguration geeignete Kandidaten sind. Anschließend evaluiert das Matching-System $\boldsymbol{f}$ die Konfiguration $\boldsymbol{x}$ (1). Dabei ist die Evaluation der Konfiguration ein kritischer Punkt, da diese die Laufzeit und somit auch die Performanz des Optimierungsprozesses beeinflusst. Daher sollten nur gut geeignete Konfigurationen für diesen Schritt ausgewählt werden. Im letzten Schritt der Iteration wird der Qualitäts-Score $\boldsymbol{f(x)}$ und die Konfiguration $\boldsymbol{x}$ genutzt, um das Regressionsmodell $\boldsymbol{M}$ neu zu trainieren (2). Die Hoffnung ist, dass somit wichtige Korrelationen von Konfigurations- und Qualitätswerten erlernt werden. Idealerweise erkennt das Modell die Eigenheiten der Konfigurationsparameter und kann bei neuen Optimierungsläufen schneller eine gute Konfiguration finden.

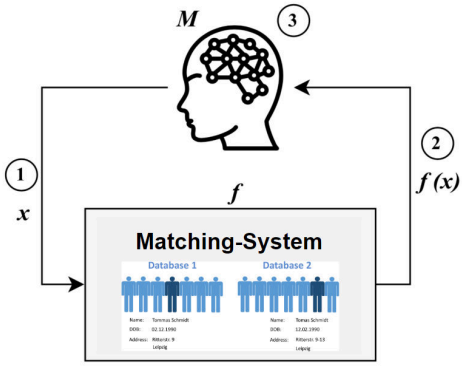

Abb. 4: SMBO Interface für die automatische Konfiguration eines Matching-Systems

Für die Vorhersage der Qualitätswerte haben wir zwei Regressionsmodelle ausgewählt, um diese im SMBO Interface zu nutzen: zum einen Extremely Randomized Trees und zum anderen Gaußsche Prozesse. Um zu zeigen, dass solch ein Konfigurationsproblem nicht trivial durch einen Direct Search Algorithmus gelöst werden kann, haben wir Random Sampling implementiert. Weiterhin haben wir auch einen, zum SMBO Interface q^1artverwandten Reinforcement Learning DQN-Ansatz [Osb16] für die Auswahl und Evaluierung von Konfigurationen verwendet.

5 Evaluation der Lern- und Optimierungsverfahren

Zu Evaluations- und Testzwecken wurden mittels eines Datengenerators synthetische Datensätze mit Namens- und Adressdaten erzeugt, die reale Datensätze mit einer Datenqualität repräsentieren, wie sie in Unternehmensanwendungen (z.B. CRM- oder ERP-Systeme)

vorzufinden sind. Der Rückgriff auf synthetische Daten erfolgt aus Rücksicht auf Datenschutzgesetze. Bei der Generierung werden gezielt Dubletten und False-Positives erzeugt. Als weiterer synthetischer Datensatz wurde ein realer Zensus-Datensatz der Mainzelliste[7] mit ≈50k Einträgen genutzt, der deutsche Adressdaten (Nachname, Vorname, Adresse, Telefon, PLZ, etc.) mit ca. 1200 bekannten Matches enthält.

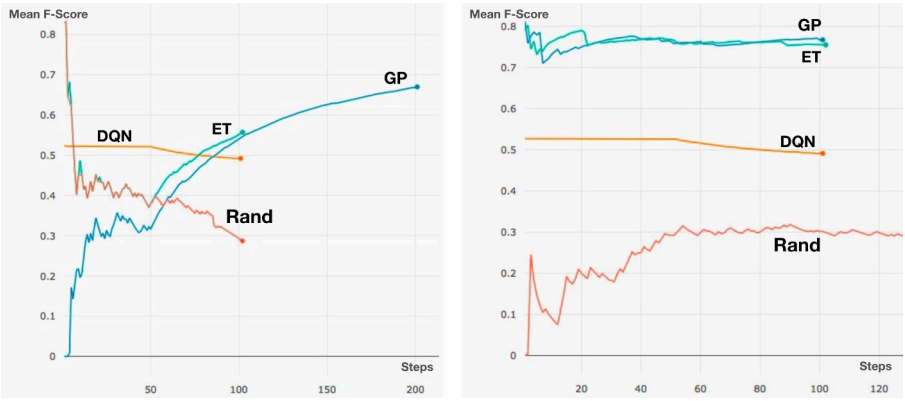

Abb. 5: Vergleich Regressions- und Optimierungsmodelle im Training- (li) und Test (re).

Auf Basis dieser erfolgte eine Einteilung in Trainings- und Evaluierungsdatensätze für einen ersten Vergleich von Random Sampling (Rand), Extremely Randomised Trees (ET), Gaussian Process (GP) und Deep Q-Learning (DQN) (s. Abbildung 5) zur Autokonfiguration des Matching-Systems von UNISERV. Die Modelle wurden mit den Trainingsdatensätzen vortrainiert und anschließend mit dem Evaluierungsdatensatz erneut trainiert (im 2. Training sollen Verbesserungen beobachtet werden). Idealerweise sollte der Agent mit zunehmender Anzahl von Versuchen lernen, das Matching-System optimal zu konfigurieren.

Im Laufe des Trainings nimmt der durchschnittliche F_1-Wert von ET und GP zu und wächst für GP logarithmisch und stabil. Rand und DQN zeigen bereits im Training kein gutes Verhalten. Die unterschiedlichen Startpunkte in der Qualität auf der Y-Achse sind zufällig gut gewählte Konfigurationen. Auch in der Evaluation zeigten Rand und DQN keinerlei Fähigkeit, eine Konfiguration zu verbessern. Dies ist für Random Sampling (Rand) nachvollziehbar, da hier nur zufällige Konfigurationswerte ausgewählt werden. DQN ist durch die Auswahl und Manipulation der Konfigurationswerte sehr begrenzt, da nur ein Wert pro Iteration verändert werden kann. Zusätzlich ist Q-Learning ein Off-Policy RL-Algorithmus, welcher erheblich mehr Beobachtungen benötigt, um sinnvolle Schlussfolgerungen für das jeweilige Optimierungsproblem zu ziehen. Somit kann DQN mit lediglich 50 bis 150 Iterationen bzw. Beobachtungen nicht erfolgreich trainiert werden. Durch die lange Laufzeit einer Deduplizierung ist der Ansatz in der Praxis nicht anwendbar. GP produziert hingegen im 2. Training mit Evaluationsdaten sofort hohe und stabile F_1-Werte, was die zu erwartende Dateneffizienz dieses Algorithmus belegt.

[7]https://www.unimedizin-mainz.de/imbei/informatik/ag-verbundforschung/mainzelliste.html

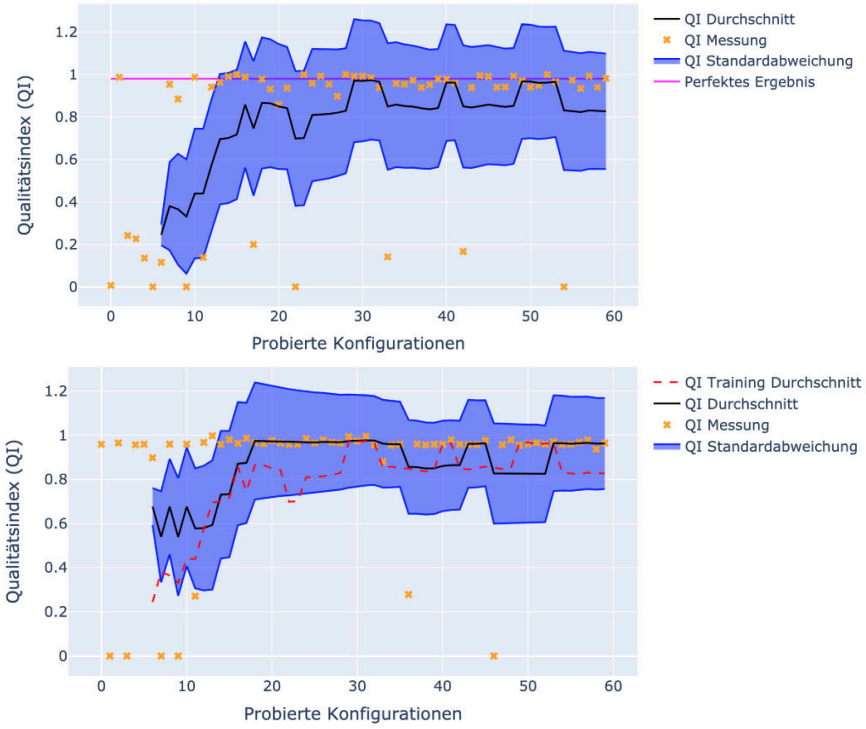

Abb. 6: Training (oben) und Test (unten) mit GP

Anhand der o.g. Datensätze wurden Gaußsche Prozesse als Regressionsmodell ausgewählt und die Evaluation intensiviert. Abbildung 6 zeigt Ergebnisse aus der Trainings- und Testphase. Visualisiert werden einzelne Messungen, Standardabweichung sowie die durchschnittliche Qualität. In der Trainingsphase benötigt GP ca. 30 probierte Konfigurationen, um ein gutes Ergebnis zu erreichen. Der Vergleich erfolgt zu einer manuell definierten bekannten Konfiguration. In der Testphase wird mit einem neuen Datensatz evaluiert und es zeigt sich, dass GP sehr viel früher gute Konfigurationen findet und schon nach 20 Schritten eine gute Konfiguration erreicht. D.h. GP hat ein Modell des Parameterraums des Matchsystems gelernt.

6 Evaluation der Gesamtlösung

Zur abschließenden Bewertung wurde die Gesamtlösung im praktischen Anwendungsfall evaluiert und automatisch erstellte Konfigurationen manuell von erfahrenen UNISERV-Matching-Experten erstellten Konfigurationen gegenübergestellt. Im Folgenden dargestellt ein spezifischer Use Case über die Bereinigung eines Datenbestands aus natürlichen

Personen für den Import in ein CRM-System, bestehend aus 1.262.769 Anschriften aus Deutschland. Der Ursprung der Daten (z.B. Call Center, Werbekampagne o.ä.) ist nicht bekannt. Pflichtfelder sind Ortsname und Nachname. Normale Fehlertoleranz ist gewünscht (Zeichenfehler, Synonyme, Initiale etc.) und mindestens gefordert bei Ortsname, Nachname und PLZ.

Abbildung 7 zeigt die durch einen Matching-Experten manuell erstellte Konfiguration zur gegebenen Problemstellung, mit einer Ergebnisqualität von 92. Die Parameter von Ortsnamen und Nachname sind identisch. Ebenso ähnlich ist die Konfiguration von Straßennamen. Die Parameterwerte werden bei manueller Einstellung (meist) auf volle 10er Schritte gerundet, da eine Mikro-Optimierung enorm aufwendig ist und Kunden meist nicht bereit sind diesen Mehraufwand zu bezahlen.

Abbildung 8 zeigt die durch die Lernkomponente automatisch erstellte Konfiguration zur gegebenen Problemstellung. Für die Gruppenbewertung nach true-positive / false-positive wurden von Fachanwender 250 Dublettengruppen qualifiziert. Nach ca. 35 Minuten konnte die Lernkomponente bereits eine Konfiguration ermitteln, die eine Ergebnisqualität von 98 erzielte. Im Vergleich dazu wurde für die manuell erstellte Konfiguration ein Wert von 92 erreicht.

```
[customer]
elements = zip,street_name,hno_num,last_name,first_name,city_name,[…],phone
#========================+======+======+======+======+======+======+======+======+======+
element = last_name      , 4   , 100 , 90  ,  0  ,  !  , 70  ,  0  ,  !  , 90 ,[…]
element = first_name     , 3   , 100 , 90  ,  0  , -100, 70  ,  0  , 90  , 90 ,[…]
element = name_rest      , 1   , 100 , 90  ,  0  , 0   , 70  , 90  , 90  ,  * ,[…]
element = zip            , 4   , 100 , 90  ,  0  ,  !  , 70  ,  0  , 90  , 90 ,[…]
element = city_name      , 4   , 100 , 90  ,  0  ,  !  , 70  ,  0  , !   , 90 ,[…]
[⋮]
element = phone          , 4   , 100 , 70  ,  0  ,  0  , 75  ,  *  ,  *  ,  * ,[…]
#========================+======+======+======+======+======+======+======+======+======+
min_mval = 70
```

Abb. 7: Konfiguration erstellt von Matching-Experte

Die Ergebnisse der automatisch erstellten Konfigurationen wurden umfassend ausgewertet. Zu bemerken ist:

- Die automatisch ermittelten Parameterwerte sind feingranularer mit 71, 62, 64, etc. Punkten und im Vergleich zur manuellen Konfiguration nicht auf 10er-Schritte gerundet. In der Praxis würde eine Rundung auf „volle 5er-Schritte“ eine ähnliche Qualität liefern bei zeitlich deutlich höherem Aufwand und entsprechender Erfahrung des Experten.

```
[optim_AI]
elements = zip,street_name,hno_num,last_name,first_name,city_name,[…],phone
#======================+======+======+======+======+======+======+======+======+======+
element = last_name      , 4   , 100 , 64  , 62  , -100, 55  , 35  , !   , 99 ,[…]
element = first_name     , 4   , 100 , 71  , 62  , 50  , 70  , 80  , 60  , 85 ,[…]
element = name_rest      , 2   , 100 , 73  , 55  , -50 , 60  , 5   , 65  , 15 ,[…]
element = zip            , 1   , 100 , 76  , 10  , !   , 99  , 30  , 90  , -50,[…]
element = city_name      , 3   , 100 , 67  , 42  , !   , 55  , 0   , !   , 10 ,[…]
[:]
element = phone          , 4   , 100 , 54  , 71  , 50  , 75  , 10  , 25  , 99 ,[…]
#========================+======+======+======+======+======+======+======+======+======+
min_mval = 55
```

Abb. 8: Mittels Lernkomponente automatisch erstellte Konfiguration

- Die Lernkomponente ermittelt einen deutlich tieferen Gesamtschwellenwert min_mval mit 55 Punkten (Mindestübereinstimmung, die erreicht werden muss, damit eine Dublette erkannt wird). Der Matching-Experte hat einen UNISERV-Default von 70 gewählt. Damit nutzt die Konfiguration der Lernkomponente einen breiten Punktebereich aus.

Bei den verschiedenen Tests wurde deutlich, dass die Qualität der automatisch erstellten Konfiguration stark abhängig von der Qualität der Bewertung in true-positive / false-positive (Gold-Standard) durch die Anwender ist. Für eine gute Qualität müssen Anwender etwa 100 Dublettengruppen bewerten. Bei dieser Menge ist in der Praxis immer mit widersprüchlichen Angaben (im Sinne der Matching-Software) durch die Anwender zu rechnen. Hier ist auf ein sorgfältiges Vorgehen zu achten.

7 Zusammenfassung und Ausblick

Deduplizierungssysteme, wie die von UNISERV angebotene „identity“-Lösung für Kunden- und Geschäftspartnerdaten, sind hochkomplex und müssen individuell an die kundenspezifischen Anforderungen angepasst werden. Die spezifische Konfiguration und Erstellung der Matching-Regeln erfordert tiefes technisches und großes Erfahrungswissen. Daher war Ziel eine Lösung mittels lernbasierter Verfahren zu entwickeln, welche die Erstellung von optimalen Matching-Regeln gegenüber dem bisherigen Vorgehen deutlich beschleunigt. Diese Regeln sollen nicht nur von Matching-Experten erstellbar sein, sondern es sollen auch Fachanwender des UNISERV Matching-Systems „identity“ in die Lage versetzt werden, diese Matching-Regeln zu erstellen.

Hierzu wurden verschiedene für ein KMU praxisfähige, lernbasierte Verfahren zur automatischen Konfiguration implementiert und evaluiert. Besonders geeignet erschienen hier Reinforcement Learning Ansätze. Es zeigte sich jedoch, dass aktuelle Reinforcement Learning Techniken nicht den gewünschten Effekt erreichen. Das Problem liegt dabei in der Übersetzung der vielseitigen Aktionen in der Konfiguration eines Matching-Systems auf die möglichen Aktionen eines Agenten. Deutlich bessere Ergebnisse wurden mit Sequential Model-based Optimization (SMBO)-Techniken erreicht. Tests in praktischen Anwendungsfällen mit der lernbasierten Gesamtlösung im Vergleich zu menschlichen Experten zeigen, dass optimale Matching-Regeln von sehr guter Ergebnisqualität auf Basis von Anwenderfeedback in relativ kurzer Zeit erstellt werden können. Mit dieser Lösung können bereits jetzt Junior Consultants von UNISERV in die Lage versetzt werden, in einem Kundenprojekt in kurzer Zeit eine produktionsfähige Konfiguration zu finden. Darüber hinaus kann auch eine Mikrooptimierung erzielt werden, die ansonsten zu teuer und zeitaufwendig ist. Die lernbasierte Komponente wird zukünftig auch Fachanwender (UNISERV-Kunden) befähigen, eine maßgeschneiderte Konfiguration in kurzer Zeit zu erstellen.

Danksagung: Dank an das ZIM-FuE Projekt „KOBRA – Konfiguration von Business-Regeln für Anwender von Duplikaterkennungssystemen“ (Ref. Nr. 16KN061125, https://infai.org/kobra/)

Literaturverzeichnis

[Feu15] Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; Hutter, F.: Efficient and robust automated machine learning, In: Adv Neur In, 2962-2970, 2015

[HHL11] Hutter, F.; Hoos, H.H.; Leyton-Brown, K.: Sequential Model-Based Optimization for General Algorithm Configuration. In: Proc. of Learning and Intelligent Optimization, LNCS 6683:507–23. Springer Berlin Heidelberg, 2011

[KR08] Köpcke, H.; Rahm, E.: Training Selection for Tuning Entity Matching. In: 6th Int. Workshop on Quality in Databases and Management of Uncertain Data (QDB/MUD), 2008

[KTR09] Köpcke, H.; Thor, A.; Rahm, E.: Comparative evaluation of entity resolution approaches with FEVER. In: Proc. 35th Int. Conf. on Very Large Databases (VLDB), 2009

[KR10] Köpcke, H.; Rahm, E.: Frameworks for entity matching: A comparison; Data & Knowledge Engineering, 69, 2, Elsevier Science Publishers, 197-210, 2010

[Köp12] Köpcke, H.; Thor, A.; Thomas, S.; Rahm, E.: Tailoring entity resolution for matching product offers. In: Proc. 15th Intl. Conf. on Extending Database Technology (EDBT), 545-550, 2012

[Men16] Mendoza, H.; Klein, A.; Feurer, M.; Springenberg, J.T.; Hutter, F.: Towards Automatically-Tuned Neural Networks. In: Proc. of the Workshop on Automatic Machine Learning, in PMLR 64:58-65, 2016

[Osb16] Osband, I.; Blundell, C.; Pritzel, A.; Van Roy, B.: Deep exploration via bootstrapped DQN. In: Proc. of the 30th Int. Conf. on Neural Information Processing Systems (NIPS'16). Curran Associates Inc., Red Hook, NY, USA, 4033–4041, 2016

[Rao09] Rao, J.; Bu, X.; Xu, C.-Z.; Wang, L.; Yin, G: VCONF: a Reinforcement Learning Approach to Virtual Machines Auto-Configuration. In: Proc. of the 6th Int. Conf. on Autonomic Computing (ICAC '09). ACM, New York, NY, USA, 137–146, 2009

[Sha16] Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; Freitas, N.D.; Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proc. of the IEEE, 104, 148-175, 2016

[TV20] TV, V.; Malhotra, P.; Narwariya, J.; Vig, L.; Shroff, G.: Meta-Learning for Black-Box Optimization. In: Proc. Of Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2019), LNCS, vol 11907, Springer, Cham, 2020

[Yao18] Yao, Q.; Wang, M.; Escalante, H.J.; Guyon, I.; Hu, Y.-Q.; Li, Y.-F.; Tu, W.-W.; Yang, Q.; Yu, Y.: Taking Human out of Learning Applications: A Survey on Automated Machine Learning. CoRR abs/1810.13306, 2018, https://arxiv.org/abs/1810.13306

[ZL16] Zoph, B.; Le, Q.V.: Neural Architecture Search with Reinforcement Learning. CoRR abs/1611.01578, 2017, https://arxiv.org/abs/1611.01578