Exposé zur Diplomarbeit

Analyse und Optimierung eines Systems zur Dublettenbereinigung für Katalogmanagementsysteme

Humboldt Universität zu Berlin
Math.-Nat. Fakultät II
Institut für Informatik

Eingereicht von

Roman Keller:	Humboldt Universität zu Berlin Institut für Informatik Matr.Nr.: 177679

Betreuer

Prof. Dr. Ulf Leser:	Knowledge Management in Bioinformatics Humboldt Universität zu Berlin Institut für Informatik
Dr. Michael Eimermacher:	European IT Consultancy EITCO GmbH

17. Oktober 2008

Inhaltsverzeichnis

1 Einleitung

Viele Unternehmen in unterschiedlichen Branchen führen eine breite Produktpalette, die gepflegt, aktualisiert und durchsucht werden soll. In Bereichen, wie *eProcurement*, *Marktplätzen* und weiteren, kommen dazu komplexe Systeme zur Produkt- und Katalogverwaltung zum Einsatz.

Es ist wichtig, dass ein eProcurement-System einfach zu bedienende Funktionalitäten bietet, um die verschiedenen Organisationen des einkaufenden Unternehmens und auch der Lieferanten, Hersteller und Kunden in das System zu integrieren. Durch den Einsatz von solch modernen Beschaffungslösungen lassen sich materieller und zeitlicher Aufwand für den strategischen Einkauf, aber auch für seine Lieferanten und Bedarfsträger (Endkunden) deutlich senken, und die Kosten für die Gestaltung und Umsetzung der Geschäftsprozesse über gesamte Wertschöpfungskette innerhalb des Unternehmens können reduziert werden.

Im Geschäftsalltag werden Produkte zum Gebrauch oder Verbrauch angeboten. Als Produkt definieren wir ein zum Kauf angebotenes Objekt oder eine Dienstleistung. Alle Produkte verschiedener Hersteller und/oder Lieferanten werden vom eProcurement-System in einem firmen-internen Gesamtkatalog zusammengefasst. Durch die unterschiedlichen Interessen der Lieferanten kann es im Gesamtkatalog zu Redundanzen kommen: Einerseits können verschiedene Hersteller sehr ähnliche Produkte anbieten. Andererseits kann ein Produkt desselben Herstellers von mehreren Lieferanten vertrieben werden. Sofern es für den Endnutzer annähernd egal ist, welches Produkt bzw. aus welcher Fundstelle im Gesamtkatalog er das Produkt bekommt, sprechen wir in allen diesen Fällen von „Dubletten“. Zwar existieren verbreitete Klassifizierungsstandards, wie ***eCl@ss***[1] oder ***UNSPSC***®[2], die ein Produkt in einer mehrstufigen Klassifikationshierarchie einordnen, die Zuordnung ist jedoch nicht eindeutig. Denn einerseits erlaubt z.B. der *eCl@ss*-Standard keine Multi-Hierarchie (das Verwenden einer Klasse an mehreren Stellen in der Hierarchiestruktur). Andererseits wollen die Produzenten häufig bewusst ein Produkt unter mehreren Klassen ablegen, oder verschiedene Produzenten legen ein ähnliches Produkt unter verschiedenen Klassen ab. Daher ist ein klassenübergreifender Vergleich nötig. Außerdem trennt die Klassifikation selbst nicht zwischen ähnlichen und verschiedenen Produkten innerhalb der Klassen.

[1] `http://www.eclass.de` - Internationaler Standard zur Klassifizierung und Beschreibung von Produkten und Dienstleistungen

[2] `http://www.unspsc.org` - The United Nations Standard Products and Services Code®

Dubletten sind für eProcurement und Marktplätze aus verschiedenen Gründen interessant. Beim eProcurement behindern sie die Effizienz des strategischen Einkaufs: Je mehr Lieferanten ein Produkt in den Gesamtkatalog einstellen können, desto geringer ist die Markt-Macht für den Einkauf beim Aushandeln der Rahmenverträge. Außerdem erhöht sich durch Zersplitterung der Verwaltungsaufwand. Schließlich führen redundante Angebote auch bei den Bedarfsträgern häufig zu längeren Suchvorgängen beim Bestellen eines Produktes. So kann die Suche nach der vermeintlich besten Alternative viel teurer werden als das Produkt selbst. Aus diesen Gründen hat der strategische Einkauf ein hohes Interesse, Dubletten zu vermeiden. Umgekehrt sieht es häufig bei Marktplätzen aus: Hier möchte man gerade auf ähnliche Produkte hinweisen. Nur so kann der Endkunde gezielt zwischen Produkten vergleichen, und dies erhöht den Mehrwert der Marktplätze.

Die Identifikation von Dubletten muss bisher - wenn überhaupt machbar - während einer langweiligen manuellen Datenreinigung angegangen werden, was kostenaufwendig und fehleranfällig ist.

Bei großen Mengen werden die Produkte oft in Kataloge gepackt. Hier kommt ein *Katalogmanagementsystem* (kurz *KMS*) zum Einsatz. Ein *KMS* ist meistens mächtig genug, unterschiedlich standardisierten Kataloge importieren und verwalten zu können. Es existieren mehrere Standards für den elektronischen Austausch von Produktkatalogen, wie zum Beispiel ***BMEcat***[3], oder ***xCBL***[4]. Der Letztere stellt z.B. eine Sammlung vordefinierter XML Dokumente und deren Komponenten und XML-Schemas, die der Standardisierung des Austausches von Geschäftsdaten dienen und frei benutzt werden dürfen, dar. Als Beispiel für ein *Katalogmanagementsystem* kann ***jCatalog***[5] genannt werden, das das Importieren von Katalogen in mehreren Formaten (z.B. *BMEcat*, *xCBL*, *CUP*, *ASCII*, *CSV*) unterstützt. Solche Systeme bieten außerdem oft verschiedene Funktionalitäten für Lieferanten, Einkaufsorganisationen oder Marktplatzanbieter, wie z.B. eigenständige Einspielung und Pflege der Kataloge für Lieferanten, oder verschiedene Reporting- und Statistiktools für einkaufende Instanzen.

Ein *Katalogbereinigungssystem* (kurz *KBS*) kann bei der Datenaufbereitung, Klassifizierung und Qualitätssicherung unterstützen. In den Beschreibungen/Definitionen zu den einzelnen Produkten verstecken sich oft wichtige Informationen, die nicht nur einem potenziellen Käufer, sondern auch dem *Katalogmanagementsystem* nutzen können, wie z.B. Maßangaben eines mechanischen Ersatzteils. Diese Eigenschaften würden dem System erlauben, eine feinere Klassifizierung eines gegebenen Produkts zu vollziehen, die Strukturierung und Normalisierung des gesamten eigenen Datenbestandes zu verbessern und die Effizienz verschiedener Operationen auf der Datenmenge zu steigern oder gar erst zu ermöglichen. Doch die Produktbeschreibungen bringen auch Schwierigkeiten mit sich.

[3] `http://www.bmecat.org/deutsch/index.asp`
[4] Xml Common Business Library: `http://www.xcbl.org`
[5] `http://www.jcatalog.de`

Da diese frei definierbar sind, können (und das ist auch meistens der Fall) zwei unterschiedliche Lieferanten dasselbe Produkt mit vollkommen verschiedenen Beschreibungen versehen (knappe vs. ausführliche Beschreibung, Beschreibungsform und Wortwahl, Maßangaben in Zentimetern vs. Maßangaben in Millimetern u.s.w.). Dies führt dazu, dass ein „unvorbereitetes“ System diese Produkte als unterschiedlich wertet und bestimmte mögliche Vorgänge in dem gesamten Produktmanagement blockiert.

Es lässt sich behaupten, dass ein System, das die Erstellung/Importierung und Pflege der Produktkataloge unterstützt, ein zentrales Element einer modernen Beschaffungslösung für ein zukunftsorientiertes und fortschrittliches Unternehmen darstellt.

2 DublettenScout

DublettenScout (entwickelt von der Firma *EITCO*[1]) ist ein System zur Bereinigung von Dubletten in Multilieferantenkatalogen. Es kann als eine Standalone-Applikation zur Bereinigung der Kataloge, oder auch als ein Teil eines bereits bestehenden Katalogmanagementsystems betrieben werden.

Die Diplomarbeit sieht eine Evaluation des Systems, zum Teil auch durch ein methodisches und konsequentes Untersuchen und Anwenden der einzelnen (noch zu definierenden und zum Teil zu implementierenden) Funktionalitäten des Systems, vor. Die Auswirkungen der Teilschritte bei der Bereinigung eines gegebenen Produktkatalogs auf die Gesamtperformanz der Duplikaterkennung sollen untersucht werden. Es soll unter Anderem festgestellt werden, wie sich die Performance der Duplikaterkennung mit oder ohne vorangegangener Lemmatisierung der Texte unterscheidet.

Schwerpunkt der Bereinigung im *DublettenScout* ist zunächst die Erkennung von Dubletten. Innerhalb des *Scouts* werden dazu die Produktbeschreibungen nach verschiedenen Kriterien bereinigt, und Attribute werden aus den Texten extrahiert. Die Bereinigung der Kataloge wird in Teilschritten vollzogen, wie z.B. eine Abbildung von Flexionsformen auf gemeinsame Grundform. Einige dieser Schritte sollen im Rahmen dieser Diplomarbeit modifiziert bzw. implementiert und/oder beschrieben und evaluiert werden.

[1] `http://www.eitco.de`

3 Dublettenlokalisierung

Dublettenlokalisierung, also die Identifikation von Datensätzen aus unterschiedlichen Quellen, die auf dasselbe Objekt oder zwei sehr ähnliche Objekte aus der realen Welt referenzieren, ist einer der Schritte der Informationsintegration.

Zum Thema Dublettenlokalisierung allgemein existieren bereits verschiedene Ansätze. Die traditionellsten Verfahren, um die Ähnlichkeit zwischen Texten zu überprüfen, zielen dabei entweder auf die Methoden, die auf den zeichenbasierten Editierungsoperationen beruhen, wie Lösch-/Einfüge-/Austausch- und Vergleichsoperationen auf der Zeichenebene, oder auf *Vektorraum-Retrieval* (*Vector Space Model* - *VSM*, siehe hier: [4]). *VSM* ist im Gegensatz zu zeichenbasierten Methoden besser zur Anwendung für die Ähnlichkeitsprüfung auf längeren Texten geeignet. Es betrachtet einen gegebenen Text als eine Menge von Token (*a bag of words*) und abstrahiert dabei (zumindest teilweise) von der Reihenfolge des Auftretens dieser Token im Text. Ein gegebener Text von n Token wird als ein m-dimensionaler Vektor dargestellt, wobei m für die Größe von dem Token-Vorrat (z.B. ein Lexikon), aus dem dieser Text gebildet wurde, steht. Dieser Vektor besteht aus nichtnegativen ganzen Zahlen, die die Anzahl der Vorkommnisse eines jeden Token aus dem Korpus im gegebenen Text darstellen [2]. *TF-IDF* (siehe hier: [3]) ist eine verbreitete Methode, die *VSM*-basierten Vektoren zu erzeugen und dabei eine Gewichtung einer jeden Dimension im Kontext nicht nur des aktuell vorliegenden Textes (Dokumentes), sondern mehrerer Texte (Dokumentenfamilie, Document-Level Statistics) mit einer einfachen Gewichtungsfunktion, die auf Vorkommnissen in Dokumentenfamilien basiert, vorzunehmen. Im Gegensatz zu den vordefinierten Stoppworten (wie „und“, „oder“), die meistens bereits noch vor einer *VSM*-Berechnung aussortiert werden, ist es hierdurch möglich, auch kontextspezifische Stoppworte (wie „Auto“ in einem Dokument aus der Autoindustrie) mit Hilfe der Gewichtungsfunktion auszufiltern.

Diese beiden richtungweisenden Herangehensweisen sind gut dokumentiert und in weiteren Variationen/Optimierungen vorhanden. Außerdem existieren Lernverfahren auf diesen Einsatzgebieten, die die Methoden beschreiben, welche die Justierung der Parameter z.B. bei der Ermittlung der *Levenshtein-Distanz* (siehe hier: [5]), oder bei der Anwendung von *VSM*-Algorithmen erlernen lassen und durch diese die Verfahren zwecks Qualitätssteigerung automatisch steuern können (siehe hierzu: [6]) - diese wären wiederum zum Feststellen oder Anpassen an eine bestimmte Einsatzdomäne verwendbar (z.B. postalische Adressen).

Die oben beschriebenen Verfahren eignen sich leider nur bedingt bei einer Dublettensuche in Produktkatalogen, da diese sich rein auf textuelle Eigenschaften bei dem Vorgang

verlassen. In unserem Fall wären weitere Schritte, wie z.B. Extraktion und eine, nach bestimmten Szenarien vorbereitete, Auswertung der extrahierten Merkmale eines Produktes und dessen Beschreibung notwendig. Man betrachte an dieser Stelle als Beispiel nur die Vielzahl von Schrauben, die sich nur durch Länge oder Durchmesser unterscheiden. Ihre Beschreibung ist sehr ähnlich, aber sie sind sicher keine Dubletten, was bei einer genügend langen Beschreibung mit hoher Wahrscheinlichkeit eine rein textbasierte Methode behaupten würde.

Im weitesten Sinne handelt es sich, wie bei den oben beschriebenen existierenden Methoden, so auch im unseren Fall, immer um eine Hash-Funktion, wenn sie auch sehr komplex aussehen mag. Doch es kommt eben darauf, relevante Eigenschaften für diese Funktion zu ermitteln und sie zu definieren.

Da es weder einen Standard-Multilieferantenkatalog noch eine allgemeingültige Definition von Dubletten gibt, ist es sogar für Menschen manchmal schwierig, zu zwei gegebenen Produktbeschreibungen eindeutig festzulegen, ob es sich dabei um Dubletten handelt. An dieser Stelle geben wir eine informelle Definition von Dubbleten aus der Sicht des Systems *DublettenScout*:

Definition 1 (Dubletten)**.** *Zwei Artikeln werden als* ***Dubletten*** *bezeichnet, wenn der Score-Wert für die Kombination ihrer Ähnlichkeitswerte (z.B. im Bereich von* 0 *bis* 100*) über einer einstellbaren Schranke (z.B.* 90*) liegt. Die Ähnlichkeitswerte umfassen u.a.:*

- *Übereinstimmung der für diese Warengruppe definierten Attributtypen (z.B. aus dem Text extrahierte Längenangaben)*
- *Anteil der übereinstimmenden Texttoken*

Ein Score-Wert wird nach dem in der Abbildung 3.1 dargestellten Schema ausgerechnet, wobei jeder der einzelnen Scoring-Mechanismen einen bestimmten Score-Anteil zu der Gesamtwertung einbringt. Dabei ist zu beachten, dass die Menge und die Auswahl der variablen Filter vom Durchlauf zum Durchlauf variieren darf, jedoch in jedem einzelnen Durchlauf für beide zu vergleichende Artikel immer übereinstimmt.

Entsprechend dieser Definition ist der Aufwand bei der Bereinigung eines Produktkataloges und bei der Erkennung von Dublikaten umfangreich. Diese Prozesse beinhalten viele Aspekte, vom Inhalt und Umfang des eigenen Lexikons bis zu Reihenfolge der Ausführung einzelner Module innerhalb des Systems. Die Herausforderung besteht darin, all diese Aspekte zu betrachten, einzelne Bereinigungsschritte zu formulieren, zu definieren und zu evaluieren und deren Zusammenspiel gewinnbringend zu organisieren.

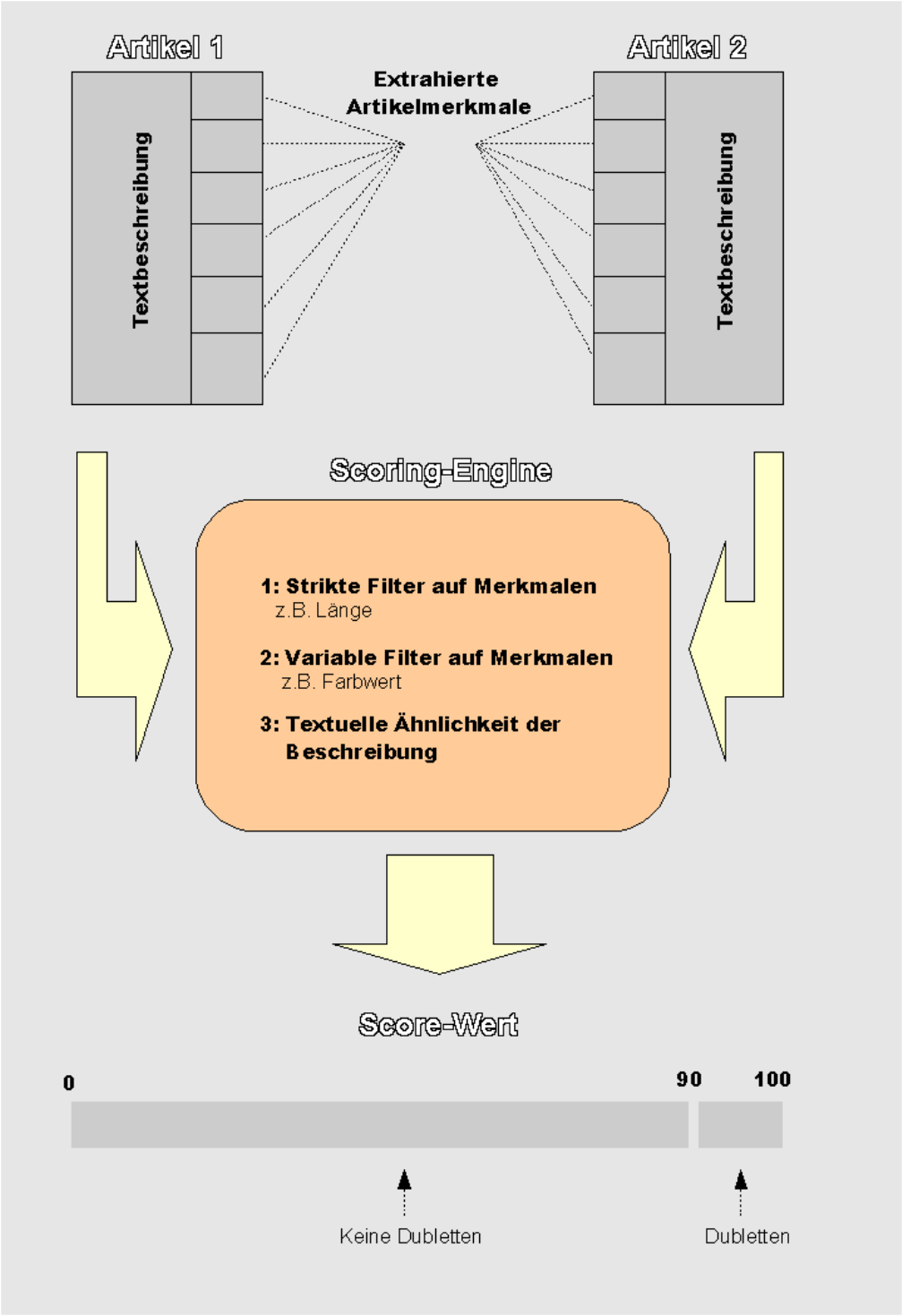

Abbildung 3.1: Scoring-Mechanismus bei Dublettenerkennung

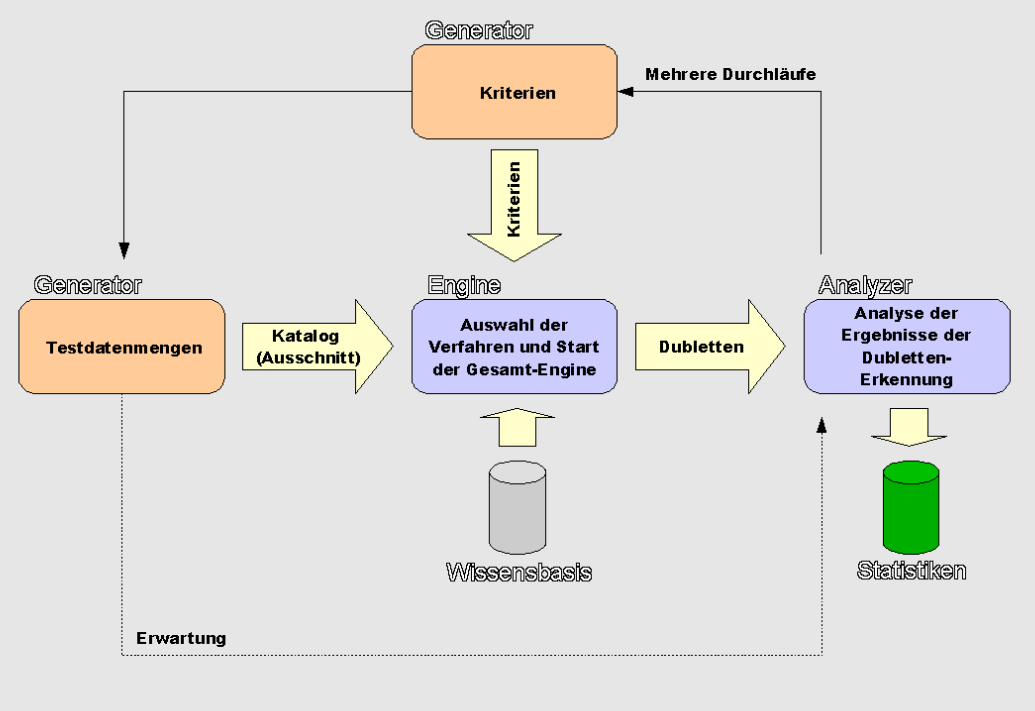

Abbildung 3.2: Workflow des Analyse-Tools

Es sollen tiefer-greifende Probleme sowie auf der Backend-Ebene des Systems, als auch die aus den oberen Schichten angesprochen werden. So soll z.B. untersucht werden, welche linguistische Regeln dem Auffinden von Dubletten überhaupt verhelfen können, indem sie zur Bereinigung einer gegebenen Beschreibung angewendet werden können, und welche allgemein geltenden bis spezifischen Heuristiken basierend auf diesen Regeln definiert werden können. Auch welche internen Datenstrukturen bei der Lösung einer speziellen Aufgabe zu wählen wären, ist manchmal eine interessante Frage.

Ein großer und wichtiger Teil der Diplomarbeit soll sich dem halbautomatischen Lernen für die Dubletten-Recherche widmen. Die Extraktion von Attributen aus einer Produktbeschreibung ist ein wesentlicher Arbeitsschritt im Ablauf der Bereinigung. Wir bezeichnen diesen Schritt mit *Wissensextraktion*. Mit Hilfe der extrahierten Attribute ist es möglich, eine Steuerung im System vorzunehmen, welche Informationen in welchen Kontexten als Kriterium für eine Dublettenmarkierung gelten sollen und welche nicht (Gewichtung auf den möglichen Steuerhebeln mit „voll“ gewichten bzw. aus der Auswertungsfunktionalität ausschließen - mit Null gewichten).

Vor dem eigentlichen Bereinigen und Importieren eines Kataloges, soll eine Teilmenge seines Datenbestandes von einer konstanten relativen Größe (z.B. 5%) zufällig gewählt und probeweise bereinigt werden. Es sollen mehrere Durchläufe einer solchen Probebereinigung mit jeweils unterschiedlich eingestellten Kriterien durchgeführt werden, wobei eine Reihe von vordefinierten Szenarien/Strategien formuliert und definiert werden soll, die festlegen, welche Kriterien angewendet werden sollen. Unter Kriterien verstehen wir hier verschiedene Filter- und Schrankeneinstellungen und sogar ganze Verarbeitungsprozesse aus den Bereinigungs-, Wissensextraktions- und Clusteringsschichten des Systems. Dabei sollen zu jedem Durchlauf statistische Informationen gesammelt werden, wie viele Dubletten (basierend auf den Ergebnissen eines Durchlaufes des Systems, sprich *Score-Wert*, siehe oben die Definition von *Dubletten* und Abbildung 3.1) und mit welchen Einstellungen lokalisiert wurden, wie viele verschiedenartige Attribute und wie verteilt in dem Katalog vorkommen, wie sehr die Menge der Produkte über die *eCl@ss*-Hierarchie verstreut gewesen ist, oder auch wie stark das System dabei ausgelastet wurde und welcher zeitlicher Aufwand bei der Wahl einer bestimmten Strategie entstehen kann. Basierend auf einer anschließenden Analyse kann man schlussfolgern, welche Strategie sich wie effizient und wie qualitativ gezeigt hat und die „beste“ davon auszeichnen.

Ein Tool, welches die besprochene vorbereitende Analyse durchführt und eine passende Dublettenlokalisierungsstrategie evaluiert, kann in der anfänglichen Phase des Systembetriebs einen Vorschlag über eine solche Strategie liefern. Der Benutzer seinerseits teilt seine Erfahrungen über die Qualität der Dublettenmarkierungsergebnisse mit, welche das Tool im Speziellen und das System im Gesamten feiner justieren lassen, und einem vollautomatischen Betrieb der Dublettenlokalisierung näher bringen können. Die spezifischen statistischen Daten und Merkmale, die nach dem beschriebenen Durchlauf entstanden sind, zusammen mit den dazugehörigen Entscheidungen des Nutzers sowie des Systems selbst werden zwecks halbautomatischen Lernens persistent gesichert.

Die Diplomarbeit sieht eine prototypische Implementierung eines solchen Tools (siehe Abbildung 3.2) und seine Integration in den *DublettenScout* vor.

4 Evaluation

Die Diplomarbeit sieht eine Evaluierung des Systems, also *DublettenScout*, die Beurteilung der Effizienz und des Gewichtes einzelner zuschaltbarer Funktionalitäten vor. Dies geschieht, indem man eine prototypische Version des oben beschriebenen Analyse-Tools implementiert, bestimmte Ausführungsszenarien für das Tool formuliert und dieses Tool und seine Ergebnisse während der Anwendung auswertet. Es sind folgende Fragen zu den einzelnen ausgewählten Komponenten zu stellen:

- *Effektivität*: verbessert die jeweilige Funktion das Ergebnis, indem sie hilft, sonst nicht erkannte Dubletten zu entdecken, oder verhindert sie dies bzw. erzeugt sie unerwünschte Pseudo-Ähnlichkeiten?
- *Effizienz*: wie hoch ist der Rechenaufwand?
- *Handhabbarkeit*: wie hoch ist der Bedienaufwand?
- *Vollständigkeit*: sollte die Funktion erweitert werden bzw. fehlt eine relevante Funktion?
- Falls Effektivität negativ ist oder in keinem sinnvollen Verhältnis zu Effizienz und Handhabbarkeit steht, ist zu klären, ob die Funktion ausgeschaltet oder ersetzt werden sollte.

Somit soll gezeigt werden, welche ausgewählten Teile und Vorgehensweisen das Katalogbereinigungssystem bei der Dublettenbereinigung mit welcher Effizienz und Qualität unterstützen, welche davon das System überlasten und als inakzeptabel ausgezeichnet werden können und welche die Ergebnisse sogar durch viele False-Matches negativ beeinflussen können.

Zum Zweck der Evaluierung des Analyse-Tools und des Systems im Gesamten wird von der Firma *EITCO* ein Korpus zur Verfügung gestellt, welcher mindestens 1000 Produkte mit mindestens 50-60 Dubletten enthält. Die Dubletten sind in dem Korpus als solche gekennzeichnet und der Korpus ist bei der Evaluierung als Goldstandard zu verstehen.

Literaturverzeichnis

[1] Thor Center for Neuroinformatics, Technical University of Denmark: „Information Retrieval“ - http://isp.imm.dtu.dk/thor/projects/multimedia/textmining/node1.html

[2] Mikhail Bilenko and Raymond J. Mooney, Department of Computer Sciences, University of Texas at Austin: „Adaptive Duplicate Detection Using Learnable String Similarity Measures“ - http://research.microsoft.com/~mbilenko/papers/03-marlin-kdd.pdf

[3] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze: „Introduction to Informationretrieval“ - http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html

[4] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze: „Introduction to Informationretrieval“ - Vector Space Model - http://nlp.stanford.edu/IR-book/html/htmledition/scoring-term-weighting-and-the-vector-space-model-1.html

[5] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze: „Introduction to Informationretrieval“ - Edit Distance - http://nlp.stanford.edu/IR-book/html/htmledition/edit-distance-1.html

[6] William W. Cohen and Jacob Richman: „Learning to Match and Cluster Large High-Dimensional Data Sets for Data Integration“ - http://www.cs.cmu.edu/~wcohen/postscript/kdd-2002.pdf

[7] Needleman, S.B., Wunsch, C.D., J. Mol. Biol. Vol.48, pp.443-453 (1970): „A general method applicable to the search for similarities in the amino acid sequence of two proteins“