

Entwicklung einer Benutzeroberfläche zur interaktiven Regelgenerierung für die Suche in historischen Dokumenten

Andrea Ernst-Gerlach, Dennis Korbar, Ara Awakian

Universität Duisburg-Essen
Lotharstr. 65,
47048 Duisburg
{ernst, korbar, awakian}@is.inf.uni-due.de

Zusammenfassung

Für Retrieval in historischen Dokumenten wird eine Abbildung der Suchbegriffe auf ihre historischen Varianten in den Dokumenten benötigt. Hierfür wurde ein regelbasierter Ansatz entwickelt. Mit dem *RuleGenerator* wurde dafür eine Benutzeroberfläche entworfen, die es dem Anwender selbst ermöglicht, automatisch Belege zu erstellen und daraus Regeln zu generieren und zu bearbeiten. Im Rahmen einer eyetrackergestützten Benutzerstudie wurde die Benutzerfreundlichkeit des Werkzeugs evaluiert.

Abstract

Retrieval in historic documents with non-standard spelling requires a mapping from search terms onto the terms in the document. For describing this mapping, a rule-based approach has been developed. The *RuleGenerator* user interface enables the user to create evidences automatically and hence to generate and edit rules. With an eyetracker-based user evaluation the usability of this tool has been demonstrated.

Einleitung

Trotz verfügbarer Volltextsuche für die wachsende Zahl an digitalen, historischen Kollektionen werden viele Dokumente nicht gefunden, weil die

Schreibweise in vielen Ländern über Jahrhunderte hinweg nicht festgelegt war. Die nicht-standardisierte Schreibweise führt zu Fehlern, da die meisten Benutzer den Suchbegriff in moderner Sprache eingeben, und diese sich von der historischen Sprache in den Dokumenten unterscheidet. Z. B. ist *akzeptieren* die moderne Form der Schreibvariante *acceptieren*. Um dieses Problem zu lösen, wurde eine regelbasierte Suchmaschine entwickelt, bei der der Benutzer seine Anfragen in zeitgemäßer Schreibweise eingeben kann, wenn er in historischen Texten suchen möchte [Ernst-Gerlach/Fuhr 07].

Mit wörterbuchbasierten Methoden (z. B. [Hauser et al. 07]) können nur Wörter gefunden werden, die im Wörterbuch enthalten sind. Außerdem ist der zeitliche Aufwand für den Aufbau der Wörterbücher relativ hoch. Werden Ähnlichkeitsmaße (z. B. [Pilz 09]) verwendet, muss der Suchbegriff mit jedem Wort aus dem Index verglichen werden. Die entwickelte Suchmaschine überwindet diesen Nachteil mit einem regelbasierten Ansatz, um das gesamte Vokabular abzudecken und dadurch den Recall zu erhöhen. Dafür werden Transformationsregeln generiert, die aus einem Suchbegriff die historischen Varianten generieren [Ernst-Gerlach/Fuhr 06].

Die automatische Regelgenerierung benötigt als Trainingsdaten Belege aus modernem Lemma und zugehöriger historischer Variante. Daraus werden die sogenannten Regelkerne bestimmt. Diese beinhalten die notwendigen Transformationen und identifizieren den zugehörigen Kontext. Z. B. ergibt sich für die moderne Wortform *unnütz* und die historische Form *unnuts* die folgende Menge, die aus zwei Regelkernen besteht: $\{(unn(\ddot{u} \rightarrow u)t), (t(z \rightarrow s))\}$. Im zweiten Schritt werden für jeden Regelkern die zugehörigen Regelkandidaten (z. B. $\ddot{u} \rightarrow u, n\ddot{u} \rightarrow nu, \ddot{u}t \rightarrow ut, n\ddot{u}t \rightarrow nut, C\ddot{u} \rightarrow Cu, z\$ \rightarrow s\$$) bestimmt. Diese berücksichtigen auch die Kontextinformationen (z. B. Konsonant (C) oder Wortende (\$)) der modernen Schreibweise. Im letzten Schritt werden die nützlichen Regeln durch Pruning der Regelmenge bestimmt. Dazu wird eine modifizierte Version des PRISM-Algorithmus [Cenderowska 87] eingesetzt.

Durch die Zeit- und Ortsabhängigkeit der Regeln müssen die Regelsätze jeweils neu generiert werden, wenn ein neues Korpus verfügbar wird. Zur Vereinfachung dieses Prozesses ist eine Benutzeroberfläche erstellt worden, die dem Benutzer sowohl bei der Belegerstellung als auch bei der Regelerstellung interaktive Unterstützung bietet. Da in [Nguyen et al. 06] für den Bereich des Data Mining gezeigt wurde, dass eine Visualisierung den

Benutzer bei der Bildung von neuen Regeln unterstützt, werden insbesondere geeignete Übersichten über die Belege sowie die Regelmengen und -kandidaten visualisiert. Dabei soll der Benutzer keine Informatikkenntnisse zur Bedienung benötigen. Ausgehend von seinen Bedürfnissen wird sich der Benutzer mehr auf den Recall oder die Precision seiner Suche konzentrieren. Das Werkzeug bietet an dieser Stelle die notwendige Flexibilität. Dabei wird dem Benutzer automatische Unterstützung für den gesamten Prozess der Regelgenerierung angeboten.

Im folgenden Artikel wird zunächst ein kurzer Überblick über die verwandten Arbeiten im Bereich der Regelerstellung gegeben. Anschließend wird in Abschnitt 3 der *RuleGenerator* vorgestellt. Das Werkzeug wird in Abschnitt 4 evaluiert. Der letzte Abschnitt fasst den Artikel zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

Verwandte Arbeiten

[Gotscharek et. al. 09] haben mit dem LeXtractor ein Werkzeug zur Konstruktion von historischen Lexika entwickelt. Die Lexikoneinträge können auch als Belege aufgefasst werden. Da Lexikoneinträge eine hohe Präzision aufweisen sollen, muss ein Experte alle unbekannt Wörter der Kollektion überprüfen. Zur Unterstützung wird eine Liste mit Textstellen angeboten, wenn ein Wort für die Konstruktion eines Lexikoneintrags ausgewählt wird. Der LeXtractor verwendet manuell erstellte Regeln (sog. Patterns), um potenzielle moderne Formen in einem aktuellen Lexikon zu finden.

Pilz und Luther [Pilz/Luther 09] unterstützen die Sammlung von Belegen in ihrem Evidencer Werkzeug mit einem Bayes'schem Klassifizierer, der die Wahrscheinlichkeit schätzt, ob es sich um eine Schreibvariante handelt. Nach einer Trainingsphase mit bekannten Belegen wird eine Liste mit unbekannt Wörtern präsentiert, welche absteigend nach der Wahrscheinlichkeit für Schreibvarianten sortiert werden. Der Benutzer kann den Klassifizierer anpassen, indem er den Grenzwert für mögliche Varianten verändert.

VARD 2 [Baron/Rayson 08] ist ebenfalls in der Lage, moderne Formen für Schreibvarianten in historischen Dokumenten zu finden. Das Werkzeug markiert alle Varianten, die nicht in einem modernen Lexikon zu finden sind.

Für jedes markierte Wort wird dem Benutzer eine Liste mit potenziellen zugehörigen modernen Schreibungen angeboten. Der Benutzer kann aus der Liste die passende moderne Form auswählen. Ein zweiter Modus bietet zudem die Möglichkeit, automatisch die Vorschläge mit dem höchsten Ranking zu akzeptieren, wenn der Wert über einem vom Benutzer festgelegten Mindestwert liegt. Um diese Vorschläge zu generieren, werden eine manuelle Liste mit Belegen, eine modifizierte Version des SoundEx-Algorithmus und eine manuell erstellte Liste von Ersetzungsregeln verwendet.

Zusammenfassend lässt sich feststellen, dass alle betrachteten Ansätze einen hohen manuellen Aufwand benötigen, bevor sie zur Suche nach Schreibvarianten eingesetzt werden können. Deswegen würde ein automatisches Werkzeug den Zugang zu historischen Dokumenten für den Benutzer deutlich erleichtern.

RuleGenerator

Mit dem *RuleGenerator* [Awakian 2010] (s. Abb. 1) wurde ein Werkzeug entwickelt, das dem Benutzer die automatische Generierung von Belegen und Regeln ermöglicht. Vor der Erstellung des *RuleGenerators* wurde zunächst eine Anforderungsanalyse durchgeführt. Darauf aufbauend wurde ein Design-Konzept entwickelt. Hierfür wurden zunächst mit einer hierarchischen Aufgaben-Analyse die Teilaufgaben festgelegt. Anschließend wurde das konzeptionelle Modell der Benutzeroberfläche mit Hilfe von Mockups umgesetzt und mit kognitiven Durchgängen getestet, bevor das Konzept implementiert wurde. Bei der Analyse ergaben sich die Schwerpunkte Bildung und Bearbeitung von Belegen sowie die Generierung und Bearbeitung von Regeln. Deswegen wurde zunächst eine horizontale Zweiteilung in die Komponenten *SmartEvidencer* (oben) für die Belege und *Rule-Modification* (unten) [Korbar 2010] für die Regeln vorgenommen. Der *SmartEvidencer* gliedert sich wiederum in die Komponenten *Evidences* (rechts oben) zur Belegsammlung und -bearbeitung und *Historic Text* (links oben) mit den Textquellen. Die *Rule-Modification* gliedert sich in die *RuleSelector*- und die *Rule-Visualization*-Komponenten. Die *Rule-Selector*-Komponente (links unten) erlaubt es dem Benutzer, durch die Regelmenge zu browsen und bestimmte Regeln zu finden. Wird eine

konkrete Regel ausgewählt, wird sie rechts unten in der Visualisierungs-Komponente dargestellt. Sowohl beim *SmartEvidencer* als auch bei der *Rule-Modification* wird jeweils in der linken Teilkomponente der Überblick gegeben und entsprechend der Lese- und Interaktionsrichtung werden rechts weitere Details zur Verfügung gestellt. Wird ein neuer Beleg in der Evidencer-Komponente erstellt bzw. ausgewählt, werden die daraus erstellten Regeln in der darunter liegenden Komponente Rule-Selector angezeigt. Die einzelnen Komponenten werden im folgenden näher erläutert.

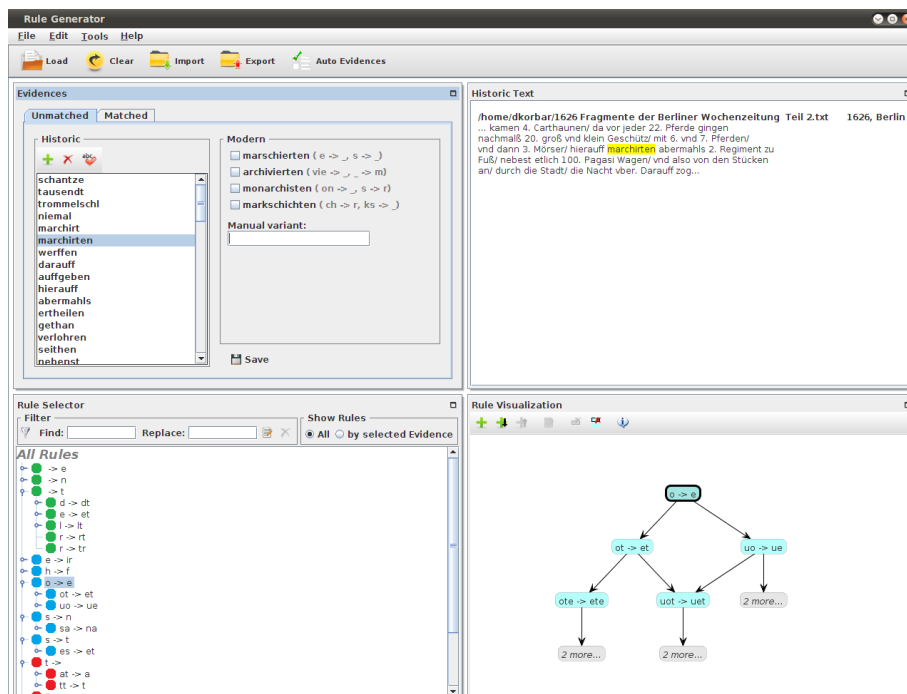


Abbildung 1: RuleGenerator Applikation

SmartEvidencer

Mit dem SmartEvidencer werden zunächst die Belege gesammelt, aus denen dann die Regeln generiert werden. Hierfür wurde die automatische Belegegenerierung [Ernst-Gerlach/Fuhr 2010a/b] integriert, die aus den unbekanntem Wörtern und Vorschlägen der Rechtschreibprüfung Belege bildet. Daraus werden wie gewohnt Regelkerne generiert und die häufigsten (für die Belegbildung) akzeptiert. Basierend auf diesen Regeln werden

schrittweise Belege vom System akzeptiert. Die Ergebnisse dieses Prozesses werden dem Benutzer in einer Liste präsentiert (s. Abb. 2). Diese zeigt die modernen und historischen Formen sowie den Regelkern an. Der Benutzer kann anschließend einzelne Belege oder alle Belege akzeptieren. Experimentelle Ergebnisse haben gezeigt, dass ein großer Teil der Belege und letztendlich auch der Regelkerne auf diese Weise generiert werden kann. Dadurch können die Trainingsdaten deutlich schneller und mit geringerem manuellen

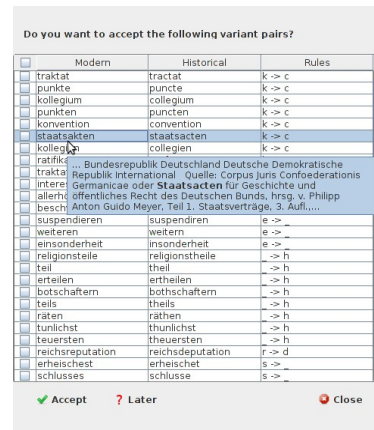


Abbildung 2: Automatische Belege

Aufwand erzeugt werden. Der Benutzer kann diesen Prozess beeinflussen, indem er die minimale Wortlänge, die minimale Anzahl an Regelvorkommen sowie die maximale Anzahl der Regelnwendungen pro Wort als Parameter einstellt.

Im Anschluss an die Bearbeitung der automatischen Belege kann der Benutzer in dem Unmatched-Reiter (s. Abb. 1 oben links) aus den noch nicht zugeordneten Wörtern weitere Belege bilden. Dazu werden dem Benutzer die unbekanntenen Wörter in einer Liste angezeigt. Zu einem ausgewählten Wort kann der Benutzer entweder einen Vorschlag der Rechtschreibprüfung akzeptieren oder selbst einen Vorschlag eingeben.

Im Matched-Reiter werden die akzeptierten Belege mit den Regeln angezeigt. Zudem besteht in dieser Ansicht auch die Möglichkeit, bereits akzeptierte Regeln noch weiter zu bearbeiten. Belege, die in dieser Ansicht mit einem Fragezeichen erscheinen, wurden automatisch generiert und mit dem Later-Button anstatt mit dem Accept-Button akzeptiert. Diese Unterscheidung ermöglicht es dem Benutzer, Belege vorläufig zu akzeptieren, wenn er schnell eine Suche ermöglichen will. Diese Belege kann er bei Bedarf zu einem späteren Zeitpunkt überprüfen.

In der *Historic-Text*-Komponente werden die möglichen Schreibvarianten in einem kurzen Textauszug hervorgehoben dargestellt. Auf diese Weise kann der Benutzer bei Bedarf aus den Kontextinformationen die genaue Bedeutung des Wortes erschließen. Falls der Textauszug nicht ausreichend ist, kann auch eine Anzeige des gesamten Textes erfolgen.

Rule-Modification

Der *Rule-Selector* soll dem Benutzer einen Überblick über die gesamte Regelmenge verschaffen und ihn bei der Suche nach einer bestimmten Regel unterstützen. Zu diesem Zweck wurde ein Filter implementiert, der es ermöglicht, die vorhandene Regelmenge nach verschiedenen Regelkriterien (z. B. Regeltyp oder *find*- und *replace*-Teil) zu filtern. Die *Rule-Visualization*-Komponente ermöglicht es dem Benutzer, die im *Rule-Selector* ausgewählte Regel im Detail zu betrachten. Außerdem erlaubt sie es, die selektierte Regel zu modifizieren, zu löschen oder neue Regeln in Abhängigkeit der ausgewählten Regel zu erstellen. Für die Visualisierung wurden zunächst die im JUNG-Framework¹ vorhandenen Layout-Algorithmen für Graphen evaluiert. Da diese Algorithmen nicht direkt für die gegebenen Daten geeignet waren, wurde ein Konzept für ein eigenes Layout erstellt (s. Abb. 1 unten rechts), welches Ansätze aus den bestehenden Verfahren übernimmt. Zur optischen Platzierung der Knoten des Graphen wurde die hierarchische Einordnung von Knoten des Sugiyama-Layouts mit der rekursiven Zeichenweise von üblichen Tree-Layouts kombiniert. Dabei wird der Wurzelknoten auf Ebene 0 angeordnet, während alle anderen Knoten in Abhängigkeit zur maximalen Anzahl passierter Kanten bis zum Wurzelknoten angeordnet werden. Daraus ergibt sich ein Layout, bei dem alle Kanten nach unten gerichtet sind. Um die Übersichtlichkeit der Visualisierung zusätzlich zu verbessern, werden nur die Regeln angezeigt, die für die ausgewählte Regel relevant sind. Weitere Regeln werden in expandierbaren Knoten zusammengefasst und können bei Bedarf zusätzlich angezeigt werden.

Innerhalb der *Rule-Visualization* Komponente stellt eine Toolbar die verschiedenen Möglichkeiten zur Modifikation der Regelmenge zur Verfügung. Der Benutzer kann vorhandene Regeln editieren und löschen oder neue Regeln erstellen. Zur Erstellung von neuen Regeln gibt es Dialoge mit unterschiedlichen Unterstützungsmöglichkeiten. Der Benutzer kann z. B. eine Regel frei eingeben oder einen Dialog verwenden, der ihn dabei unterstützt, Regeln zu erstellen, die die selektierte Regel spezialisieren oder diese generalisieren.

Bei der Durchführung von Modifikationen wird der Benutzer von einem Preview-Modus unterstützt. Dieser wird aktiviert, sobald der Benutzer einen

¹<http://jung.sourceforge.net> – Stand 13.01.2011

Modifizierungs-Dialog öffnet. Der Preview-Modus visualisiert den Graph in dem Zustand, in dem er wäre, wenn die aktuell vorhandene Änderung durchgeführt würde. Jegliche Eingabe innerhalb eines Modifizierungs-Dialogs löst eine Aktualisierung des Preview-Modus aus. So sieht der

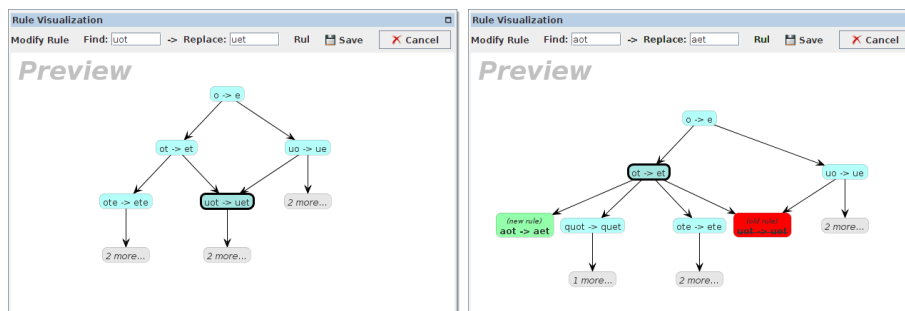


Abbildung 3: Preview-Modus

Benutzer sofort die Konsequenzen seiner Änderung und wird bei der Entscheidungsfindung unterstützt. Dies ist in Abbildung 3 zu sehen. Links wurde die Regel $uot \rightarrow uet$ ausgewählt und der Modifizierungs-Button betätigt. Daraufhin wurde sowohl im *find*- als auch im *replace*-Teil der Regel das *u* durch ein *a* ersetzt. Nach dieser Änderung wurde die Visualisierung des Preview-Modus aktualisiert. Dieser zeigt nun die ursprüngliche Form der Regel an (*old rule*) sowie die Form der Regel, die vorliegt, sofern diese Änderung bestätigt wird (*new rule*). Zusätzlich werden die Änderungen auch farblich markiert.

Evaluation

Die Benutzerfreundlichkeit der vorgestellten RuleGenerator-Applikation wurde innerhalb einer eyetrackergestützten Benutzerstudie evaluiert. An der Evaluation nahmen zehn Studenten der Universität Duisburg-Essen teil. Unter den Testpersonen befanden sich sieben Informatiker, zwei Sozialwissenschaftler und ein Erziehungswissenschaftler. Mit einem Computer arbeiteten die Testpersonen seit acht bis zwanzig Jahren und alle Testpersonen gaben an, den Computer täglich zu verwenden. Ziel der im Folgenden dargestellten Evaluation war der generelle Test der Abläufe.

Vor der Evaluation bekamen die Benutzer zunächst eine kurze Anleitung sowie eine mündliche Einführung in die Applikation. Es war den Testpersonen gestattet, während der Evaluation Fragen zu stellen. Die Fragen sowie die Maus- und Augenbewegung der Personen wurden beobachtet und protokolliert bzw. aufgezeichnet. Die Evaluation bestand aus einer Reihe von Aufgaben, die die Testpersonen innerhalb der *RuleGenerator* Applikation durchführen sollten. Im Anschluss daran gab es ein kurzes Interview und die Probanden füllten einen Fragebogen aus. Auf diesem gab es verschiedene Aussagen zur *RuleGenerator* Applikation, die die Probanden mit 1 (= trifft nicht zu) bis 5 (= trifft voll zu) bewerten sollten. Die Ergebnisse der wichtigsten Aussagen sind in Abbildung 4 zu sehen. Die ersten vier Aussagen in der Abbildung beziehen sich auf die Applikation insgesamt. Wie dort zu sehen ist, wurde die Applikation im allgemeinen positiv bewertet. Der Median von „Tool übersichtlich“, „Bedienung einfach“ und „Bedienung flüssig“ liegt jeweils bei vier und auch die weiteren Werte liegen bis auf die Ausreißer im positiven Bereich. Lediglich die Icons der Applikation wurden in der Aussage „Icons eindeutig“ mittelmäßig bewertet. Im Interview mit den Probanden bestätigten sich diese Ergebnisse und es stellte sich heraus, dass die Probanden vor allem Probleme mit den Icons in der Rule-Modification hatten.

Die nächsten vier Aussagen in der Abbildung 4 beziehen sich auf den Evidencer-Teil der Applikation. Dort wurde von den Benutzern „Belegfilter hilfreich“, „Erzeugung Belege einfach“ sowie „Manuelle Belegerstellung einfach“ im Median ebenfalls mit 4 bewertet. Die Aussage „Beleg editieren einfach“ wurde etwas schlechter bewertet. Im Interview zeigte sich, dass einige Personen Probleme hatten, den Zugang zu dieser Funktion zu finden. Sie erwarteten, dass die Operation ebenfalls über die Toolbar erreichbar ist, diese konnte jedoch lediglich über einen Doppelklick aktiviert werden.

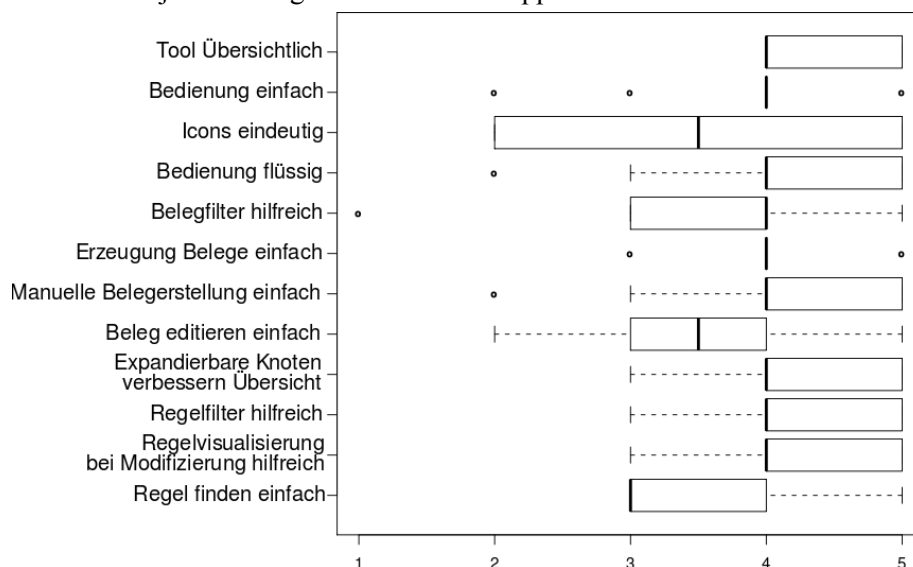


Abbildung 4: Aussagenbewertung der Probanden

Die letzten vier Aussagen beziehen sich auf den Teil *Rule-Modification*. Dort bewerteten die Probanden die Aussagen „Expandierbare Knoten verbessern Übersicht“, „Regelfilter hilfreich“ sowie „Regelvisualisierung bei Modifizierung hilfreich“ ebenfalls mit einem Median von vier. Die Aussage „Regel finden einfach“ wurde von den Benutzern mit einem Median von drei bewertet. Im Interview und bei der Auswertung der Eyetracker-Daten ergab sich, dass die Benutzer beim Finden von Regeln vor allem Probleme damit hatten, dass der Regelbaum bei der Benutzung des Filters nicht automatisch expandiert wurde.

Durch die Verwendung des Eyetrackers wurde deutlich, dass viele Teilnehmer den Bestätigungsdialog bei Modifikationen erst sehr spät bemerkten. Nachdem der Benutzer die ausgewählte Regel gelöscht hatte, schaute er auf den Belegteil oben, ohne den Dialog, der noch bestätigt werden musste, zu bemerken. Weitere Mängel, wie der fehlende Edit-Button im Belegteil, konnten anhand der gesammelten Eyetracker-Daten untersucht und bestätigt werden. Dadurch war z. B. ersichtlich, dass Probanden bei der Bearbeitung von Aufgaben, in denen Belege zu editieren waren, den Belegteil sequentiell nach einer Möglichkeit zum Editieren absuchten, anstatt einen Doppelklick auf dem Beleg durchzuführen.

Die bei der Benutzerstudie gefundenen Probleme wurden in einer Anpassung der Implementierung behoben. So wird für den Bestätigungsdialog der Regelmodifizierung der Hintergrund zunächst rot eingefärbt, um die Aufmerksamkeit des Benutzers zu erregen. Des Weiteren wurden eindeutigere Icons für einige Funktionen gewählt und die mittelmäßig bewerteten Filterfunktionen überarbeitet. Bei der Auswertung der aufgezeichneten Eyetracker-Daten wurden weitere Probleme gefunden, wie etwa fehlende alternative Möglichkeiten, um bestimmte Funktionen auszuführen. Diese wurden behoben, indem zusätzliche Möglichkeiten – wie etwa Kontextmenüs – implementiert wurden.

Zusammenfassung

In diesem Artikel wurde eine Benutzeroberfläche zur automatischen Konstruktion von Belegen und Regeln vorgestellt. Die Benutzeroberfläche versetzt Benutzer in die Lage, ohne Programmierkenntnisse eigenständig

Regeln für historische Korpora zu erzeugen und ermöglicht somit Retrieval auf Texten mit nicht-standardisierter Rechtschreibung. Dem Benutzer wird dabei die Gelegenheit gegeben, die automatischen Vorschläge für Belege und Regeln zu löschen, zu editieren sowie neue zu erstellen. Dadurch ist der Ansatz sehr flexibel, weil der Benutzer den Prozess entsprechend seinen Erwartungen an Recall und Precision der Suchmaschine beeinflussen kann.

Die Evaluation hat gezeigt, dass neuen Benutzern die Bedienung der Applikation leicht fällt und die grundlegenden Konzepte der Applikation verstanden werden. Die bei der Evaluation gefundenen Probleme, wurden in einer Anpassung der Implementierung behoben. Dies soll im Rahmen einer weiteren Evaluation mit späteren Benutzern wie z. B. Historikern und Linguisten überprüft werden.

Literaturverzeichnis

Awakian, A. (2010). Development of a user-interface for an interactive rule development. Masterarbeit, Universität Duisburg-Essen

Baron, A.; Rayson, P. (2008). VARD 2: A tool for dealing with spelling variation in historical corpora. Proc. of the „Postgraduate Conference in Corpus Linguistics“. Aston University, Birmingham

Cendrowska, J. (1987). An Algorithm for Inducing Modular Rules. „International Journal on Man-Machine Studies“. Volume 27, Nummer 4, Seiten 349-370

Ernst-Gerlach, A.; Fuhr, N. (2006). Generating Search Term Variants for Text Collections with Historic Spellings. Proc. of the "Advances in Information Retrieval - 28th European Conference on IR Research, ECIR 2006, London, UK, April 10-12, 2006"

Ernst-Gerlach, A.; Fuhr, N. (2010a). Advanced Training Set Construction for Retrieval in Historic Documents. Proc. of the "Sixth Asia Information Retrieval Societies Conf. (AIRS 2010)"

Ernst-Gerlach, A.; Fuhr, N. (2010b). Semiautomatische Konstruktion von Trainingsdaten für historische Dokumente. Proc. of the "Information Retrieval 2010 Workshop LWA 2010, Kassel, Germany

Gotscharek, A.; Neumann, A.; Reffle, U.; Ringlstetter, Ch.; Schulz, K. U. (2009). Enabling information retrieval on historical document collections: the role of matching procedures and special lexica. Proc. of „The Third Workshop on Analytics for Noisy Unstructured Text Data“ Barcelona, Spain

Hauser, A.; Heller, M.; Leiss, E.; Schulz, K. U.; Wanzeck, C. (2007). Information Access to Historical Documents from the Early New High German Period. Proc. of the „International Joint Conference on Artificial Intelligence (IJCAI-2007) Workshop on Analytics for Noisy Unstructured Text Data“ Hyderabad, India

Korbar, D. (2010). Visualisierung von Regelstrukturen und -Modifikationsmöglichkeiten für die Suche in Texten mit nicht-standardisierter Rechtschreibung. Diplomarbeit, Universität Duisburg-Essen

Nguyen, DucDung, T. Ho, Kawasaki (2006). Knowledge visualization in hepatitis study. In: APVis '06: Proc. of the 2006 „Asia-Pacific Symposium on Information Visualization“, S. 59-62, Darlinghurst, Australia. Australian Computer Society, Inc.

Pilz, T. (2009). Nichtstandardisierte Rechtschreibung -Variationsmodellierung und rechnergestützte Variationsverarbeitung. Dissertation Universität Duisburg-Essen

Pilz, T. and Luther, W. (2009). Automated support for evidence retrieval in documents with nonstandard orthography. In „The Fruits of Empirical Linguistics Process“ Sam Featherston, S; Winkler, S.; Volume 1, S. 211-228, Mouton de Gruyter Berlin