Volltext via OCR

Möglichkeiten und Grenzen

Maria Federbusch & Christian Polzin

Staatsbibliothek zu Berlin
Preußischer Kulturbesitz

Maria Federbusch
Christian Polzin

Volltext via OCR – Möglichkeiten und Grenzen

Testszenarien zu den Funeralschriften der Staatsbibliothek zu Berlin – Preußischer Kulturbesitz

Mit einem Erfahrungsbericht von Thomas Stäcker aus dem Projekt „Helmstedter Drucke Online“ der Herzog August Bibliothek Wolfenbüttel

Staatsbibliothek zu Berlin – Preußischer Kulturbesitz

Impressum

Beiträge aus der Staatsbibliothek zu Berlin – Preußischer Kulturbesitz
Band 43

Herausgegeben von Barbara Schneider-Kempf
Generaldirektorin der Staatsbibliothek zu Berlin

Layout, Druck: Gallery Print, Berlin (Boris Brumnjak, Leo Lamprecht)
Buchbinderische Verarbeitung: Stein + Lehmann, Berlin
ISBN 978-3-88053-185-7

Inhalt

Zum Geleit

Texterkennung oder auch Optische Zeichenerkennung (OCR) ist ein zentrales Thema der Wissensgesellschaft am Beginn des 21. Jahrhunderts. Die Image-Digitalisierung macht heute schon Bücher jederzeit, überall und in kürzester Zeit verfügbar, doch erst über durchsuchbare Volltexte potenzieren sich die Möglichkeiten wissenschaftlichen Erkenntnisgewinns.

Bibliotheken mit bedeutenden historischen Beständen können hier entscheidend zum Entstehen entsprechender virtueller Forschungsumgebungen beitragen. Die Staatsbibliothek zu Berlin – Preußischer Kulturbesitz spielt bereits seit Jahren eine wesentliche Rolle beim Auf- und Ausbau der nationalbibliographischen Verzeichnisse der deutschen Drucke des 16. bis 18. Jahrhunderts (VD16, VD17 und VD18). Die Digitalisierung dieser Werke wird über zahlreiche, meist von der Abteilung Historische Drucke betreute Projekte entscheidend vorangetrieben. Die Staatsbibliothek setzt den Fokus in besonderem Maße auf die Erschließung der Images etwa durch eine tiefgehende Strukturdatenerfassung. Seit einigen Jahren engagiert sich die Staatsbibliothek für Einrichtung und Aufbau von Virtuellen Bibliotheken bzw. Portalen wie Europeana.eu, das Zentrale Verzeichnis Digitalisierter Drucke oder die World Digital Library. Der Präsident der Stiftung Preußischer Kulturbesitz ist Vorstandssprecher der Deutschen Digitalen Bibliothek, die 30.000 deutsche Kultur- und Wissenschaftseinrichtungen vernetzen und das nationale Kulturerbe über eine gemeinsame Plattform öffentlich zugänglich machen soll.

Die Staatsbibliothek sieht eine wesentliche Zukunftsaufgabe in der Weiterentwicklung des Angebots digitalisierter Sammlungen hin zu Volltextbibliotheken. Die vorliegende Studie präsentiert die Ergebnisse eines zwölfmonatigen Pilotprojekts zur Evaluierung unterschiedlicher Software-Lösungen für Fraktur-OCR. Ich freue mich sehr, dass die Deutsche Forschungsgemeinschaft mit ihrer Förderung dieses innovative Projekt ermöglicht hat.

So konnte eine für Software-Anbieter eher periphäre, für die deutschsprachigen Drucke der Frühen Neuzeit aber entscheidende Problemstellung in einem intensiven Testprozess analysiert und Möglichkeiten und Grenzen der OCR-Anwendung bei diesem Material ausgelotet werden. Als Testbestand wurde eine vom 16. bis 18. Jahrhundert im protestantischen Bereich massenhaft verbreitete Quellengattung ausgewählt, die von den unterschiedlichsten historisch orientierten Disziplinen intensiv beforscht wird und als serielle Quelle besonders hohes Potenzial für eine Volltexterschließung aufweist: Die Staatsbibliothek besitzt eine der umfangreichsten Sammlungen von Funeralschriften (Leichenpredigten und andere Trauerschriften), aus der im Projekt ein kleines Testsample mit OCR erschlossen wurde.

Die Ergebnisse der von der Abteilung Historische Drucke in Zusammenarbeit mit zwei Software-Anbietern durchgeführten komplexen Testverfahren können nun präsentiert werden. Chancen, Herausforderungen und Aufwände einer gattungsspezifischen OCR-Konversion des anspruchsvollen Materials werden umrissen. Erfreulicherweise wird die Publikation durch einen Beitrag von Thomas Stäcker ergänzt, der die einschlägigen Projekterfahrungen aus der Herzog August Bibliothek Wolfenbüttel in die Studie einbringt. Ich hoffe, dass der so entstandene umfassende Praxisbericht eine wesentliche Hilfestellung bei der Etablierung und Weiterentwicklung automatischer Texterkennungsverfahren für Frakturschriften geben kann.

Karl Werner Finger
Ständiger Stellvertreter der Generaldirektorin der Staatsbibliothek zu Berlin PK

1 Einführung
1.1 Projekthintergrund

Seit mehreren Jahren ist die Staatsbibliothek zu Berlin – Preußischer Kulturbesitz (SBB PK) bestrebt, ihre herausragende und überregional bedeutende Funeralschriftensammlung[1] zu digitalisieren. Über eine Imagedigitalisierung hinausgehend sollen auch Volltexte entstehen. Daher wurde 2008 das *Pilotprojekt zum OCR-Einsatz bei der Digitalisierung der Funeralschriften der Staatsbibliothek zu Berlin*[2] konzipiert. Im folgenden Jahr sagte die Deutsche Forschungsgemeinschaft eine Förderung für die Dauer von 12 Monaten zu. Von Oktober 2010 bis September 2011 setzte eine Projektkraft in der Abteilung Historische Drucke dieses Vorhaben um. Ziel war es, die Volltexterfassung mittels optischer Zeichenerkennung (OCR) an bis zu 2.000 Drucken (maximal 50.000 Seiten) aus dem Bestand der Staatsbibliothek zu testen. Diese Anzahl Funeralschriften berücksichtigt einen Ausschnitt aus der mit ca. 15.000 Drucken weltweit zweitgrößten Sammlung ihrer Art. Projektinhalt bildeten Testszenarien für zwei grundlegend verschiedene Softwarelösungen. Die Produkte wurden auf OCR-Erkennungsqualität sowie Konfigurations- und Optimierungsmöglichkeiten hin untersucht und verglichen. Die Umsetzung einzelner Aufgaben erfolgte dabei arbeitsteilig mit den Dienstleistern (zugleich auch Software-Entwicklern). Zur Handhabung der Software bestand mit diesen ein kontinuierlicher Informationsaustausch.

Im Pilotprojekt wurden folgende zwei OCR-Produkte getestet:

- B.I.T. Bureau Ingénieur Tomasi SARL Toulouse[3] – Software: BIT-Alpha
- Herrmann & Kraemer GmbH und Co-KG Garmisch-Partenkirchen[4] – Software: HK-OCR auf Basis der ABBYY FineReader Engine 9.

Durch die inhaltliche Ausrichtung des Materials auf Funeralschriften sowie die daraus resultierenden regionalen und zeitlichen Schwerpunkte ergaben sich zudem gattungsspezifische Optimierungsmöglichkeiten, handelt es sich doch überwiegend um Drucke des 17. und beginnenden 18. Jahrhunderts aus dem mitteldeutschen Raum. Beispielhaft entstand eine Wortliste, die Begriffe versammelt, welche häufig in Funeralschriften benutzt werden. Darüber hinaus integriert diese Wortbibliothek auch die Thesauri der Forschungsstelle für Personalschriften Marburg (THELO[5] und THEPRO[6]) sowie eine Liste historischer Krankheitsbezeichnungen des Vereins für Computergenealogie[7].

Im Projekt wurden Möglichkeiten zur automatisierten Volltexterfassung größerer Funeralschriftenbestände getestet. Speziell für die weitere Digitalisierung der Funeralschriften der Staatsbibliothek konnten Erkenntnisse für ein Vorgehen in der Bibliothek selbst gewonnen werden.[8] Weitere Etappen auf dem Weg zu einer vollständigen digitalen Funeralschriftensammlung werden planbar. Eine Übertragbarkeit der Ergebnisse auf zukünftige Massendigitalisierungsprojekte aus dem Bereich der frühneuzeitlichen Drucke wird angestrebt.

Der vorliegende Werkstattbericht schildert sowohl die Ausgangslage, den methodischen Ansatz und die Ergebnisse des Softwarevergleichs als auch die erreichte OCR-Optimierung anhand des Beispielmaterials. Im Einzelnen werden neben Aussagen zur Erkennungsqualität Einschätzungen zur Binarisierung (Bildvorbearbeitung und Wandlung in Schwarz/Weiß) und zur Bild-, Wort- und Zeichensegmentierung getroffen. Zusätzlich bewertet werden die Nutzung von Zeichen- und Wortbibliotheken sowie vorhandene Trainings- und Validierungsmöglichkeiten. Die Dokumentation der gewählten Parameter und Einstellungen eröffnet Bibliotheken die Möglichkeit der späteren Weiterverwendung von Projektergebnissen.

Die Beschreibung von Nutzungsszenarien der untersuchten Softwareprodukte zeigt potentiellen Anwendern sowohl die Möglichkeiten als auch die generellen Probleme der OCR-Prozesse und des Ma-

[1] http://staatsbibliothek-berlin.de/die-staatsbibliothek/abteilungen/historische-drucke/sammlungen/bestaende/personale-gelegenheitsschriften/ [Stand: 02/2013]
[2] http://staatsbibliothek-berlin.de/die-staatsbibliothek/abteilungen/historische-drucke/aufgaben-profil/projekte/funeralschriften/ [Stand: 02/2013]
[3] http://bit.dyndns.biz/ [Stand: 02/2013]
[4] http://www.hk-gap.de/ [Stand: 02/2013]
[5] http://www.personalschriften.de/datenbanken/thelo.html [Stand: 02/2013]
[6] http://www.personalschriften.de/datenbanken/thepro.html [Stand: 02/2013]
[7] http://wiki-de.genealogy.net/Kategorie:Krankheitsbezeichnung [Stand: 02/2013]
[8] http://www.slideshare.net/mdz-bsb/digitalisierungspraxis-federbusch-ocrpraxistest [Stand: 02/2013] und http://mdzblog.wordpress.com/ [Stand: 02/2013]

terials selbst. Als wesentliches Projektergebnis ist die Abhängigkeit der Softwarebeurteilung von den jeweiligen Anwendungsszenarien zu nennen. Je nach Zielstellung wird eines der beiden Softwareprodukte günstiger erscheinen. Die vorliegende vergleichende Darstellung einzelner Funktionen und Einsatzmöglichkeiten soll Interessenten dabei unterstützen, sachbezogen den Einsatz von OCR-Software zu planen. Auch wenn die Defizite in der OCR-Erkennungsqualität gegenüber Drucken des 19. Jahrhunderts unbestritten sind, ist es möglich, auch die OCR-Konversion ausgewählter frühneuzeitlicher Texte sinnvoll durchzuführen und zu optimieren.

Auch mittels OCR-Konversion entstandene Volltexte eröffnen zusätzliche Suchmöglichkeiten in digitalen Daten. Damit geht die Funktionalität von Digitalisaten über das bloße Lesen der Texte hinaus. Stäcker erläutert in seinem Beitrag in Kap. 8.6 ff die verschiedenen Vorstellungen über ausreichende Textgenauigkeiten und die daraus resultierenden Nutzungsmöglichkeiten. Neben der direkten Einbindung der entstandenen OCR-Daten bietet sich sowohl die manuelle nutzerseitige Korrektur als auch Verbesserung durch nachgelagerte halbautomatische Prozesse (z. B. hinsichtlich der Layouterkennung) sowie semantische Auszeichnung an.

1.2 Hilfsthesen

Um die Softwareprodukte nicht nur aufeinander zu beziehen, sondern sie auch ins Umfeld weitergehender Anwender-Erwartungen einzuordnen, wurde während der Projektarbeit von folgenden Annahmen ausgegangen:

- Wesentliche Verarbeitungsschritte der optischen Zeichenerkennung sind auch im bestmöglichen Fall regelmäßig von Informationsverlust betroffen. Ein praxisorientierter Ansatz wird sich zweckmäßigerweise auf die Betrachtung derjenigen Fehlerquellen konzentrieren, die gegenwärtig praktikabel behandelbar oder wenigstens aussichtsreich problematisierbar sind. Dennoch kann es während des Softwarevergleichs sinnvoll sein, auch auf Selbstverständlichkeiten wie vorläufig hinzunehmende Informationsverluste hinzuweisen und die jeweiligen Lösungsangebote hiernach einzuordnen.

- Erkennungsfehlern (systematischen wie unsystematischen) wird auf zwei prinzipiell unterschiedliche Arten begegnet: (a) durch Vorkehrungen zu ihrer Vermeidung und (b) durch nachträgliche Korrektur, d. h. Akzeptanz des Fehlers bei Aussicht auf Kompensation. Unabhängig davon, wann dieser Unterschied praxisrelevant ist oder nicht, ist es sinnvoll, die betrachteten Softwaremerkmale und Lösungsstrategien jeweils einem dieser Typen zuzuordnen.

- In manchen Anwendungsfällen sind Informationsverluste einiger Verarbeitungsstufen unschädlich. Auch wenn sich Anwender in der Praxis dessen bewusst sind und gerade hierauf setzen, kann eine systematische Nennung der Bedingungen, unter denen Informationsverluste harmlos werden können, die Planung von OCR-Vorhaben unterstützen.

- Auch wenn beide Hersteller ihre Software als integrierte Gesamt-OCR-Lösung konzipiert haben und sie primär sogar in eigener Regie als Dienstleister nutzen, könnten Teilschritte identifiziert werden, deren Ein- oder Ausgaben eine modularisierte Verwendung nahelegen, z. B. um ein Produkt speziell nur im Bereich seiner Stärken einzusetzen. Hierauf hinzuweisen lohnt sich auch dann, wenn die Import- bzw. Exportschnittstellen in der betrachteten Software noch nicht realisiert sind.

– Wenn aus Ressourcengründen für Massen-OCR meist allein eine vollautomatische Volltextgewinnung ohne manuelle Eingriffe als akzeptabel gesehen wird, wird dabei ein Verlust an menschlicher, bei manueller Erfassung intuitiv stets eingehender Objektwahrnehmung in Kauf genommen bzw. im besten Fall durch Korrekturverfahren kompensiert. Wenn es gelänge, durch Echtzeit-Evaluation von Teilschritt-Ergebnissen wenige effektive Eingriffspunkte für menschliche Justierungen zu erkennen und dafür eine Bedienoberfläche komfortabel auf sehr schnelle Eingriffe zu optimieren, dann könnte der bisher vorherrschende Einwand eines ökonomisch nicht tragbaren Zeitaufwands hierfür zumindest relativiert werden. Neben der Erkennungsgüte könnte dies die Baustelle sein, an der OCR-Arbeitsumgebungen sich in der nächsten Zeit vorrangig bewähren müssen.

1.3 Kurzfassung der Ergebnisse

Beide Programme bedienen einen ungeteilten Gesamt-OCR-Prozess von Bild-Binarisierung bis zur Wortkorrektur und sind im Prinzip für eine auf alte deutschsprachige Drucke abgestimmte Stapelverarbeitung durch Anwender geeignet. Unabhängig davon bieten beide Hersteller die OCR auch als Dienstleistung an.

Markant unterscheiden sich beide Softwareprodukte dadurch, dass BIT-Alpha eine Vielfalt detaillierter Konfigurationsmöglichkeiten bereitstellt, HK-OCR sich dagegen wegen der eingekapselten FineReader-Engine, die viele feste Voreinstellungen mitbringt, auf wenige Konfigurationsmöglichkeiten beschränkt. Es wurde gezeigt, dass folgende Möglichkeiten zur anwenderseitigen Optimierung der OCR-Erkennungsgüte bestehen:

– **Effekt von Binarisierungsoptionen:** Eine vorlagenspezifische Optimierung der Binarisierungs- und Segmentierungsparameter kann die OCR-Ergebnisse deutlich verbessern (nur BIT-Alpha; HK-OCR ist hier nicht konfigurierbar).

– **Effekt von Vorsortierung des Materials und Training eigener Muster:** Bei nach Schriftähnlichkeit vorsortierten Teilbeständen von Leichenpredigten weniger Druckorte und Jahrzehnte des 17. Jahrhunderts war ein Training speziell hierzu passender Muster sinnvoll und erbrachte deutlich bessere Ergebnisse als eine OCR mit allgemeinen Fraktur-Mustern.

– **Effekt geeigneter Wörterbücher:** Auf die Vorlage abgestimmte Lexika (gattungs- und zeitspezifische Wortformen, angereichert durch passende Titel- und Personendaten aus dem Verbundkatalog) erhöhen die Erkennungsgüte deutlich, wenn auch aufgrund verschiedener Mechanismen (in BIT-Alpha durch Nachkorrektur, in HK-OCR integriert in die Zeichenerkennung).

Für auf Seitenebene mehr oder weniger homogene Schriftbilder sind diese deutlichen Steigerungseffekte unmittelbar nutzbar (beide Programme). Für den häufigen Fall von Seiten ohne einheitliche Schriftart (Wechsel von Fraktur zu Antiqua, Wechsel von Sprachen, starke Größenunterschiede) ist gegenwärtig keines der beiden Produkte in der Lage, die Stärken einer selektiven Musterzuweisung auch auf Teilregionen von Seiten anzuwenden. Eine Kombination separat trainierter Musterbestände ist in BIT-Alpha (bzw. dem Mustereditor BIT-Knowledge) begrenzt, in HK-OCR/FineReader bisher nicht möglich; HK-OCR bietet stattdessen die Wahl, eine Anwendermustermenge mit den vorgegebenen FineReader-Mustern zu kombinieren. Analog muss zur lexikalischen Korrektur für jeden Batchlauf ein Wörterbuch bzw. eine Sprachengruppe (so der Begriff in der HK-OCR-Oberfläche) festgelegt werden; sprachlich gemischte Seiten oder Seitenstapel zwingen daher zur Verwendung eines entsprechend kombinierten Wörterbuchs mit den damit verbundenen Kompromissen in der Korrektursicherheit.

Der Export erfolgt jeweils mindestens in mit Textkoordinaten angereicherten XML-Formaten (BIT-Alpha: ALTO, HK-OCR: FineReader-XML) sowie einigen anderen Formaten (BIT-Alpha: PDF, HTML, Text; HK-OCR: PDF, RTF und ein gegenüber dem einzelzeichenorientierten FineReader-XML deutlich verschlanktes, wortorientiertes XML). Bisher von keinem der beiden Programme geleistet wird eine echte Strukturerkennung (angefragt waren formal gliedernde Elemente z. B. Kolumnentitel, Kustoden, Bogensignaturen u. a.) oder die Auszeichnung von *Named Entities* (Personen, Orte, Berufe, Krankheiten) aufgrund vorgegebener Listen oder von Datumsangaben, Bibelstellen etc.[9] Entsprechend ist eine Ausgabe in semantisch orientierten Formaten wie TEI allenfalls oberflächlich ohne die genannten textspezifischen Zusatzinformationen aus den Exporten ableitbar.

Jede Software hat Stärken und Schwächen bei der Unterstützung eines zügigen und effektiven Vorgehens besonders bei Training und Korrektur/Validierung. Diese Befunde wurden den Herstellern jeweils mitgeteilt, zumal manche Schwächen leicht behebbar erscheinen und möglicherweise einfach auf bisher unzureichenden eigenen Tests der Hersteller beruhen. Beide Firmen haben zugesagt, die Anmerkungen der SBB PK bei der Entwicklung der nächsten Versionen zu berücksichtigen.

Innerhalb dieser Grenzen können beide Programme von selbst trainierenden oder vorhandene Trainingsergebnisse nachnutzenden Anwendern sinnvoll eingesetzt werden. Im Kapitel 6.5 wird versucht, die Vor- und Nachteile jeder Software bezogen auf verschiedene Einsatzszenarien zu nennen. Die dabei beschriebenen Kriterien können zugleich als Hilfe zur Prüfung anderer OCR-Lösungen mit herangezogen werden. Als Nebenergebnis der Software-Tests konnten somit auch Anforderungen an Softwarekomponenten zur OCR Alter Drucke formuliert werden, die hiermit auch potentiellen Anwendern mitgeteilt werden, die nicht zuletzt selbst mit entsprechender Nachfrage die Entwicklungsrichtung der nächsten Versionen mitbestimmen werden.

Nicht eingehend untersucht wurden Open-Source-Lösungen[10]. Ein Vergleich mit diesen könnte weitere Optionen erschließen.

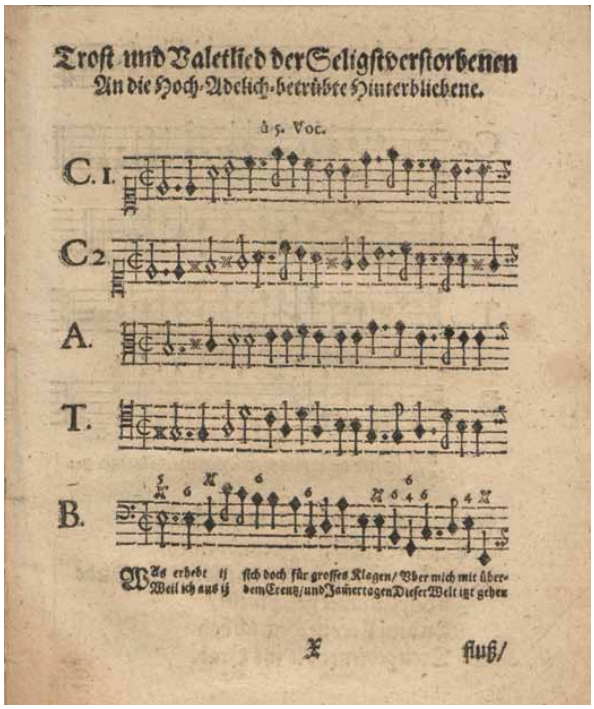
[9] Eine kleine Teilmenge aus dem „Metadatenmodell Funeralschriften (2008)“: http://staatsbibliothek-berlin.de/fileadmin/user_upload/zentrale_Seiten/historische_drucke/pdf/Metadaten_Funeralschriften.pdf [Stand: 02/2013]

[10] Ohne weitere Konfigurationen durchgeführte Tests an acht Seiten einer im Projektmaterial befindlichen Funeralschrift mit der Software OCRopus wurden im Mai 2012 durchgeführt. Das Ergebnis ist durchwachsen, fällt jedoch gegenüber den im Projekt mit optimierten Einstellungen der getesteten Programme erkannten Seiten klar ab.

2 Materialspezifik der Funeralschriften
2.1 Gattungsspezifische Aspekte

Funeralschriften sind personale Gelegenheitsschriften, deren Anfänge in die Mitte des 16. Jahrhunderts hineinreichen. Starke Verbreitung fand der Brauch, Leichenpredigten zu drucken, vor allem im protestantischen Deutschland des 17. und beginnenden 18. Jahrhunderts. In dieser Zeit, die stark durch die Theologie des Pietismus geprägt wurde, war es üblich, besonders Adelige und Akademiker, aber auch Vertreter des wohlhabenden Bürgertums nach ihrem Tode mit diesen Schriften ausführlich zu würdigen. Funeralschriften bestehen in der Regel neben der Leichenpredigt aus dem Lebenslauf der Verstorbenen. Darüber hinaus enthalten sie oft eine Abdankungsrede (Leichabdankung), ein Programma Academicum, ein Epitaph und verschiedene Epicedien (Trostbriefe, Gedichte, Lieder, Musikbeigaben Verwandter und Bekannter). Als weitere druckgraphische Ausgestaltung findet man Porträts, Wappen, Ahnentafeln, Embleme, aber auch Darstellungen des Castrum doloris (hölzerner geschmückter Aufbau zur Ehre des Toten) und des Leichenzugs.[11]

Diese Gebrauchsschriften, deren Ziel es war, neben der Würdigung der Verstorbenen vor allem den Hinterbliebenen Trost zu spenden, sind seit mehreren Jahrzehnten Gegenstand multidisziplinärer Forschung. Mit ihren vielfältigen Informationen bieten sie Einblicke in die damalige Alltagskultur und stellen wichtige Quellen für weitergehende Forschungen dar. Scheibe benennt Forschungsinteressen: „Neben ihrer Bedeutung für Literaturgeschichte und Predigtforschung bieten die Leichenpredigten dem Sozial-, Kultur- und Universitätshistoriker eine Fülle statistisch auswertbarer Daten, dem Kunsthistoriker und Heraldiker reiches Material zur Ikonographie und Emblematik, dem Musikwissenschaftler eine ganze Reihe von nur hier überlieferten Trauerkompositionen; die teilweise sogar mit Abbildungen versehenen Krankheitsberichte schließlich liefern dem Medizinhistoriker wertvolle Informationen. Darüber hinaus aber stellen die Personalschriften ein einzigartiges prosopographisches Nachweisinst-


Trost und Valetlied der Seligstverstorbenen
An die Hoch-Adelich-betrübte Hinterbliebene.
à 5. Voc.
C. I.
C 2
A.
T.
B.
Was erhebt ij sich doch für grosses Klagen/ Vber mich mit über-
Weil ich aus ij dem Creutz/ und Jammertagen Dieser Welt itzt gehen
ij fluß/

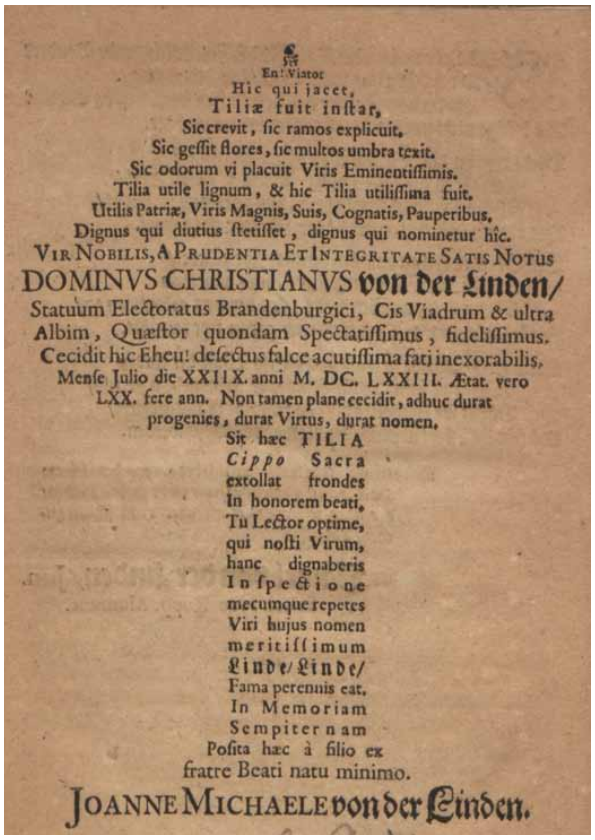
1 Komposition als Epicedium. Ee 705-1592 S. [85]


(2)
Obdieweil schon etliche Ihr Fürstl. Gn. Hochseliger Christmildter Andenckens Contrafacturen in Kupffer gestochen und sonsten gemacht/ so habe zwar ich mir auch solches für die hand genommen in Holtz zu schneiden/ aber weil ich (geliebts Gott) den gantzen Fürstl. Leich-Proceß ehender massen auff die zukünfftige Fürstl. Leichbegräbnüß/ meinem in Gott ruhenden Gn. Fürsten und Herrn/ und dem gantzen Fürstlichen Hauß und Stam Braunschweig und Lüneburg/ zu sonderer ängster Ehren und ewiger Gedächtnüß/ als dahin offenen Druck zu verfertigen stunde mal ich mich ißt daher an dem Fürstl. Hoffe in Ih. Fürstl. Gn. Dienste underthänigst auffgehalten.

2 Wappen. Ee 708-85 S. 2

[11] Eine ausführliche Beschreibung der Geschichte und des Aufbaus von Leichenpredigten ist zu finden auf der WWW-Seite der Forschungsstelle für Personalschriften Marburg: http://www.personalschriften.de/leichenpredigten/aufbau.html [Stand: 02/2013]

3 Frontispiz. Ee 705-28 S. [4]


En! Viator
Hic qui jacet,
Tiliæ fuit instar,
Sic crevit, sic ramos explicuit,
Sic gessit flores, sic multos umbra texit.
Sic odorum vi placuit Viris Eminentissimis.
Tilia utile lignum, & hic Tilia utilissima fuit.
Utilis Patriæ, Viris Magnis, Suis, Cognatis, Pauperibus,
Dignus qui diutius stetisset, dignus qui nominetur hîc.
VIR NOBILIS, A PRUDENTIA ET INTEGRITATE SATIS NOTUS
DOMINUS CHRISTIANUS von der Linden/
Statuum Electoratus Brandenburgici, Cis Viadrum & ultra
Albim, Quæstor quondam Spectatissimus, fidelissimus.
Cecidit hic Eheu! defectus falce acutissima fati inexorabilis,
Mense Julio die XXIIX. anni M. DC. LXXIII. Ætat. vero
LXX. fere ann. Non tamen plane cecidit, adhuc durat
progenies, durat Virtus, durat nomen.
Sit hæc TILIA
Cippo Sacra
extollat frondes
In honorem beati,
Tu Lector optime,
qui nosti Virum,
hanc dignaberis
Inspectione
mecumque repetes
Viri hujus nomen
meritissimum
Linde/Linde/
Fama perennis eat.
In Memoriam
Sempiternam
Posita hæc à filio ex
fratre Beati natu minimo.
JOANNE MICHAELE von der Linden.

4 Figurengedicht. 3/4 in: Ee 700-1993 S. [16]

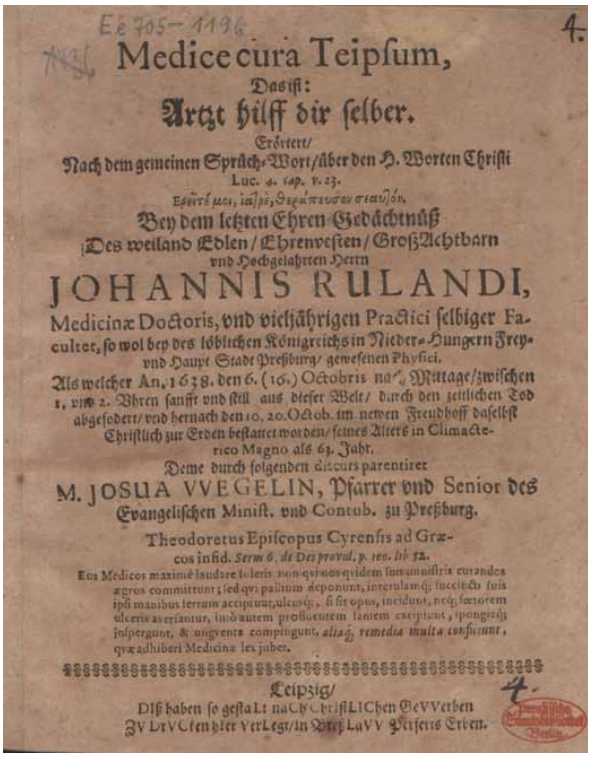
rument für Genealogie und Biographik dar – eine Qualität, die bereits im frühen 20. Jahrhundert ihre besondere Wertschätzung begründete.“[12] Derzeit stellen Volltexte von Funeralschriften ein Desiderat dar. Ein semantisch ausgezeichneter Volltext könnte die oben genannten Forschungsgebiete sinnvoll befördern; statistische Analysen sowie semantische Netze zu extrahierender Personen rücken in greifbare Nähe.

2.2 Sammlungsspezifische Aspekte

Im historischen Druckschriftenbestand der Staatsbibliothek zu Berlin befinden sich von jeher eine große Anzahl Funeralschriften. In den 30er Jahren des 20. Jahrhunderts gelang es der Bibliothek, vier geschlossene Leichenpredigtensammlungen zu erwerben, so dass heute in der Summe von 15.000 Schriften gesprochen werden kann. Der Bestand wurde um die Leichenpredigten aus der Bibliothek des Börries Freiherrn von Münchhausen auf Windischleuba (bei Altenburg/Thüringen) sowie um die Wernigeroder und die Bückeburger Sammlung vermehrt. Für das durchgeführte OCR-Projekt war vor allem eine vierte vom Antiquariat J. A. Stargardt (Berlin) 1936 vermittelte Leichenpredigtensammlung (Signatur Ee 705) von Interesse. Alle weiteren Aussagen beziehen sich vor allem auf diese Sammlung. Den Schwerpunkt bilden Funeralschriften (1.700 Nummern), die im mitteldeutschen Raum erschienen sind. Dies belegen Erscheinungsorte wie Halle, Magdeburg, Erfurt, Altenburg, Weimar, Wittenberg, Zeitz, Merseburg, Zerbst, Jena, Gotha, Leipzig, Dresden, Zwickau, Hannover, Braunschweig, Helmstedt, Coburg, Hof, Marburg, Kassel, Hanau, Giessen, Schweinfurt und Frankfurt/Main. Hinzu treten ostdeutsche Orte wie Elbing, Küstrin, Königsberg, Stettin, Breslau, Lignitz, Schweidnitz, Groß Glogau

[12] Scheibe, Michaela zur Sammlung Personaler Gelegenheitsschriften auf: http://staatsbibliothek-berlin.de/die-staatsbibliothek/abteilungen/historische-drucke/sammlungen/bestaende/personale-gelegenheitsschriften/ [Stand: 02/2013]

und Brieg. Durch die Verbreitung des Pietismus im süddeutschen Raum (z. B. Württemberger Pietismus) gibt es auch eine nicht geringe Anzahl von Predigten aus diesem Gebiet. Straßburg, Heidelberg, Ulm, Stuttgart und Tübingen sind hier zu nennen. Das breite regionale Spektrum dieser Sammlung belegen ferner Erscheinungsorte wie Berlin, Hamburg, Rostock und Basel.


Ee 705-1196

Medice cura Teipſum,

Das iſt:

Arzt hilff dir ſelber.

Erörtert/

Nach dem gemeinen Sprüch-Wort/ aber den H. Worten Christi Luc. 4. cap. v. 23.

Ἰατρὲ θεράπευσον σεαυτόν.

Bey dem letzten Ehren-Gedächtnüß

Des weiland Edlen/ Ehrenvesten/ Groß-Achtbarn vnd Hochgelahrten Herrn

JOHANNIS RULANDI,

Medicinæ Doctoris, vnd vieljährigen Practici ſelbiger Facultet, ſo wol bey des löblichen Königreichs in Nieder-Hungern Frey- vnd Haupt Stadt Preßburg/ geweſenen Physici,

Als welcher An. 1638. den 6. (16.) Octobris nach Mittage/ zwiſchen 1. vnd 2. Uhren ſanfft vnd ſtill aus dieſer Welt/ durch den zeitlichen Tod abgefodert/ vnd hernach den 10. 20. Octob. im newen Freudhoff daſelbſt Chriſtlich zur Erden beſtattet worden/ ſeines Alters in Climacterico Magno als 63. Jahr.

Deme durch folgenden discurs parentiret

M. JOSUA VVEGELIN, Pfarrer vnd Senior des Evangeliſchen Miniſt. vnd Contub. zu Preßburg.

Theodoretus Episcopus Cyrenſis ad Græcos infid. Serm. 6. de Deo provid. p. 100. lib. 12.

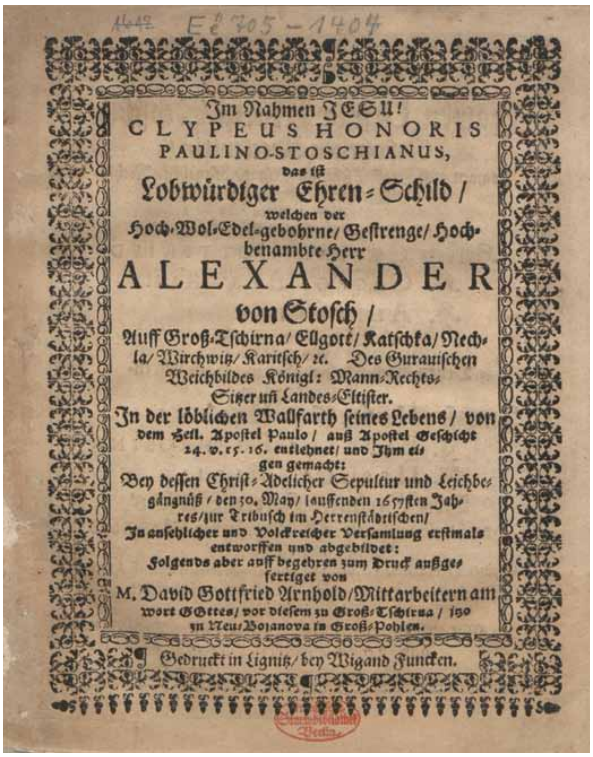
Eos Medicos maximè laudare licet, non qui nos quidem ſine omniſeris curandos ægros committunt; ſed qui palium deponunt, interulam; ſuccinctis ſuis ipſi manibus terrum accipiunt, ulcuſq; ſi ſit opus, incidunt, neq; ſœtorem ulceris averſantur, imò autem profluentem ſanies excipiunt, ſpongiisq; inſpergunt, & unguenta compingunt, aliaq; remedia multa conficiunt, quæ adhiberi Medicina lex jubet.

Leipzig/

Diß haben ſo geſtALt nACh ChriſtLIChen GeVVerben

ZV DrVCken hIer VerLegt/ In VerLaVVPrIſens Erben.

5 Leichenpredigt auf einen Arzt. Leipzig. Ee 705-1196 Titelblatt


Ee 705-1404

Im Nahmen JESU!

CLYPEUS HONORIS PAULINO-STOSCHIANUS,

das iſt

Lobwürdiger Ehren-Schild/

welchen der

Hoch-Wol-Edel-gebohrne/ Geſtrenge/ Hochbenambte Herr

ALEXANDER

von Stoſch/

Auff Groß-Tſchirna/ Ellgot/ Katſchka/ Nechla/ Wirchwitz/ Karitſch/ ꝛc. Des Gurauiſchen Weichbildes Königl: Mann-Rechts-Sitzer uñ Landes-Eltiſter.

In der löblichen Wallfarth ſeines Lebens/ von dem Heil. Apoſtel Paulo/ auß Apoſtel Geſchicht 24. v. 15. 16. entlehnet/ und Ihm zu gen gemacht:

Bey deſſen Chriſt-Adelicher Sepulcur und Leichbegängnüß/ den 30. May/ lauffenden 1675ſten Jahres/ zur Tribuſch im Herrenſtädtiſchen/

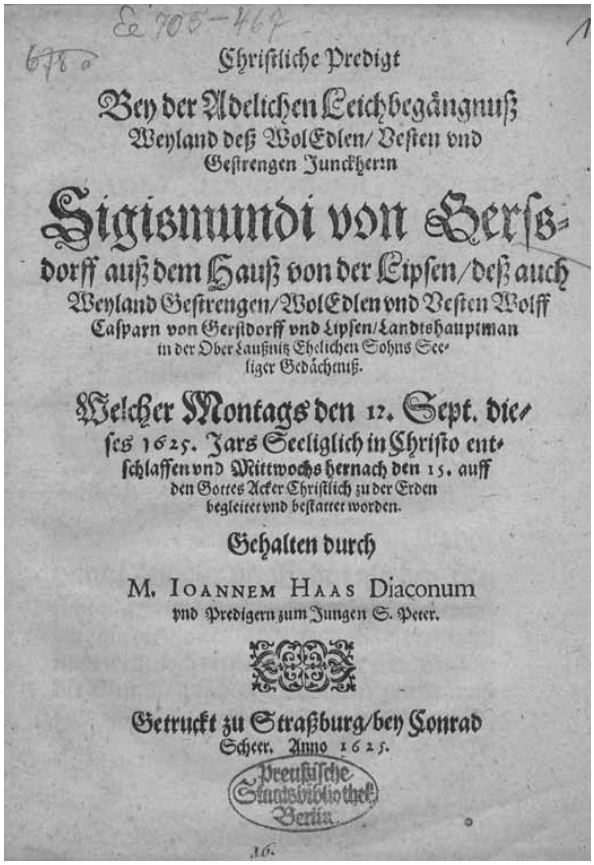
In anſehlicher und Volckreicher Verſamlung erſtmals entworffen und abgebildet:

Folgends aber auff begehren zum Druck außgefertiget von

M. David Gottfried Arnhold/ Mittarbeitern am Wort GOttes/ vor dieſem zu Groß-Tſchirna/ izo zu Neu-Bojanova in Groß-Pohlen.

Gedruckt in Lignitz/ bey Wigand Funcken.

6 Leichenpredigt. Lignitz. Ee 705-1404 Titelblatt


Ee 705-467

Christliche Predigt

Bey der Adelichen Leichbegängnuß Weyland deß WolEdlen/ Vesten vnd Gestrengen Junckherrn

Sigismundi von Gersdorff auß dem Hauß von der Lipsen/ deß auch Weyland Gestrengen/ WolEdlen vnd Vesten Wolff Casparn von Gersdorff vnd Lipsen/ Landtshauptman in der Ober Laußnitz Ehelichen Sohns Seeliger Gedächtniß.

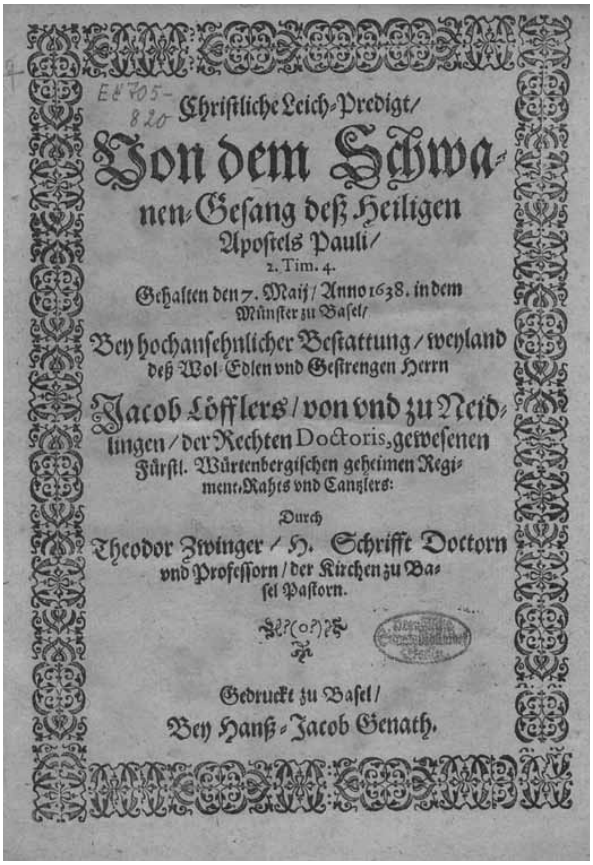
Welcher Montags den 12. Sept. dieſes 1625. Jars Seeliglich in Christo entſchlaffen vnd Mittwochs hernach den 15. auff den Gottes Acker Christlich zu der Erden begleitet vnd beſtattet worden.

Gehalten durch

M. IOANNEM HAAS Diaconum vnd Predigern zum Jungen S. Peter.

Getruckt zu Straßburg/ bey Conrad Scheer. Anno 1625.

7 Leichenpredigt. Straßburg. Ee 705-467 Titelblatt


Ee 705-820

Christliche Leich-Predigt/

Von dem Schwanen-Gesang deß Heiligen Apoſtels Pauli/

2. Tim. 4.

Gehalten den 7. Maij/ Anno 1638. in dem Münſter zu Basel/

Bey hochanſehnlicher Beſtattung/ weyland deß Wol-Edlen vnd Gestrengen Herrn

Jacob Löfflers/ von vnd zu Neidlingen/ der Rechten Doctoris, geweſenen Fürſtl. Würtenbergiſchen geheimen Regiment-Rahts vnd Canzlers:

Durch

Theodor Zwinger/ H. Schrifft Doctorn vnd Profeſſorn/ der Kirchen zu Baſel Paſtorn.

Gedruckt zu Basel/

Bey Hanß-Jacob Genath.

8 Leichenpredigt. Basel. Ee 705-820 Titelblatt

Die zeitliche Abdeckung der genannten Sammlung spiegelt die Blütezeit der Leichenpredigt[13] wider. So stehen zwei Drittel Schriften des 17. Jahrhunderts einem Drittel des 18. Jahrhunderts gegenüber. Dreißig Funeralschriften datieren ins 16. Jahrhundert. Aufgrund einfacher Auswertungen der Volltexte unter Hinzuziehung des Thesaurus Professionum der Forschungsstelle für Personalschriften Marburg[14] konnten folgende Personenkreise (in originaler Schreibung) ermittelt werden, die in den Schriften Erwähnung finden:

- Prediger / Pfarrer / Pastor / Seelsorger / Bischoff / Beichtvater / Theologus / Diaconus
- Lehrer / Rector / Praeceptor
- General / Hauptmann / Obrister / Leutenant / Soldat / Oberster / Wachmeister / Kriegsmann
- Arzt / Medicus / Medico
- Poet / Mahler
- Rath / Cantzler / Bürgermeister / Amptman / Senator
- Buchhalter / Weber / Zöllner / Pfleger / Wirth

Diese Berufsbezeichnungen beziehen sich nicht nur auf die Verstorbenen, sondern auch auf die Angehörigen und Freunde, sofern sie in der Funeralschrift erwähnt wurden. Ein Abgleich der Liste historischer Krankheitsbezeichnungen des Vereins für Computergenealogie[15] mit den Texten hinsichtlich der genannten Leiden und Todesursachen vermittelt einen ersten Eindruck in die Lebensumstände der Zeit.

Häufigste Krankheiten:

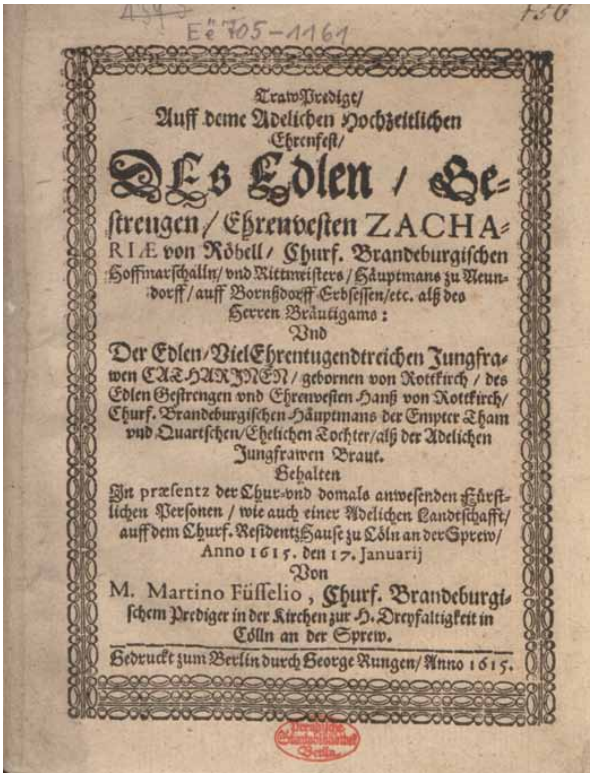
- Hitze
- Fieber
- Schlag
- Biß
- Brand
- Flecken
- Pest
- Husten

2.3 Typographische und buchgestalterische Aspekte

Neben inhaltlichen Besonderheiten weisen die Funeralschriften einige typographische und buchhistorische Gattungsspezifika auf. Wie andere Druckschriften der jeweiligen Zeit auch zeichnet sich die Mehrzahl der Drucke aus dem 17. Jahrhundert durch ein sehr gedrängtes Druckbild mit übervoll gesetzten Seiten aus. Um Papier zu sparen, wurden zudem anfangs kleine Schriften (12 Pkt) verwendet; der Satzspiegel weist fast keine Absätze im Text auf (vgl. Abb. 13). Üblicherweise gibt es Titelblätter, die in bis zu drei Schriftarten und zusammen zehn verschiedenen Schriftgraden in barocker und ausschweifender Weise den Titel nennen (vgl. Abb. 5-8 u. 10-12). Üblich sind hier zunächst stereotype Titelanfänge wie „Christliche Leichpredigt/ Bey ...“. Wesentlich direkter und variantenreicher erscheinen sie später (z. B. „Tröstende Uranie bey einem zu früh verwelckten Orangen-Zweige, Die ...“ Weimar 1718).

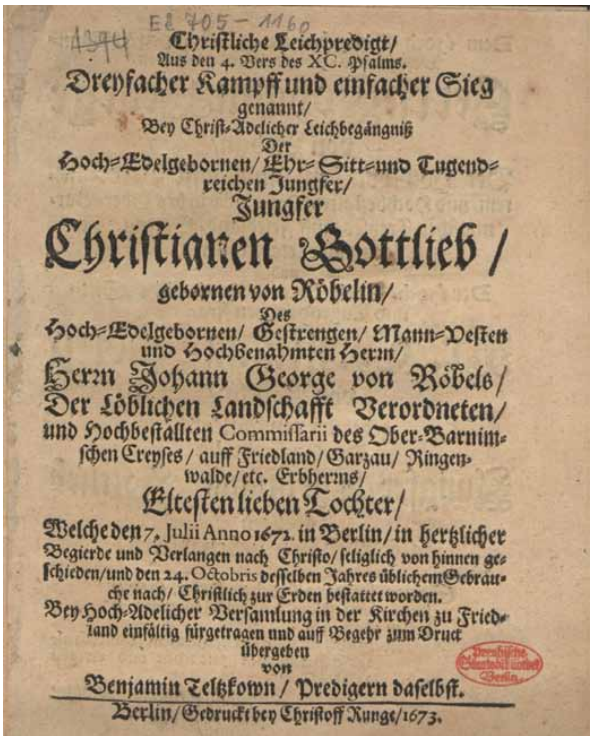
[13] Als Blütezeit gilt die Zeit vor dem 30jährigen Krieg und die zweite Hälfte des 17. Jahrhunderts.
[14] http://www.personalschriften.de/datenbanken/thepro.html [Stand: 02/2013]
[15] http://wiki-de.genealogy.net/Kategorie:Krankheitsbezeichnung [Stand: 02/2013]

9 Kupfertitel - Hamburg, 1687. Ee 705-1472



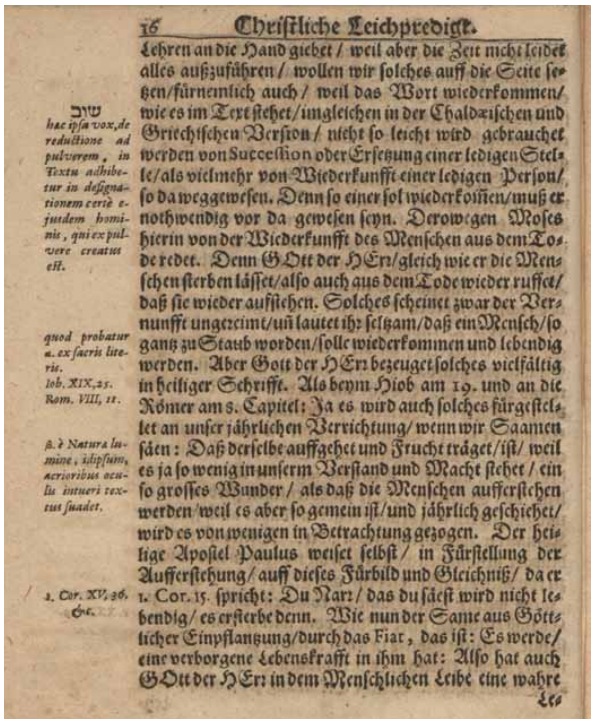
10 Titelblatt mit Titeleinfassung - Berlin, 1615. Ee 705-1161

11 Illustriertes Titelblatt 1. Hälfte 17. Jh. Hamburg, 1610. Ee 705-189



12 Titelblatt 2. Hälfte 17. Jh. Berlin, 1673. Ee 705-1160

Wie bis ins 18. Jahrhundert hinein üblich verwendete man auf den Titelblättern oft Titeleinfassungen (vgl. Abb. 6, 8 u. 10), die zunächst aus ornamentalen Motiven über Holzschnitte, später vermehrt über Kupferstiche realisiert wurden. Auch Kupfertitel und Frontispize sind zu finden (vgl. Abb. 3 u. 9). Die Rückseite des Titelblatts enthält in der Regel eine Widmung, zumeist den Angehörigen und Hauptleidtragenden des Verstorbenen zugeeignet; Vorreden können sich anschließen und leiten zur Predigt über.


16 Christliche Leichpredigt.

Lehren an die Hand giebet / weil aber die Zeit nicht leidet alles außzuführen / wollen wir solches auff die Seite setzen / fürnemlich auch / weil das Wort wiederkommen / wie es im Texte stehet / imgleichen in der Chaldæischen und Griechischen Version / nicht so leicht wird gebrauchet werden von Succession oder Ersetzung einer ledigen Stelle / als vielmehr von Wiederkunfft einer ledigen Person / so da weggewesen. Denn so einer sol wiederkommen / muß er nothwendig vor da gewesen seyn. Derowegen Moses hierin von der Wiederkunfft des Menschen aus dem Tode redet. Denn GOtt der HErr / gleich wie er die Menschen sterben lässet / also auch aus dem Tode wieder ruffet / daß sie wieder aufstehen. Solches scheinet zwar der Vernunfft ungereimt / uñ lautet ihr seltzam / daß ein Mensch / so gantz zu Staub worden / solle wiederkommen und lebendig werden. Aber Gott der HErr bezeuget solches vielfältig in heiliger Schrifft. Als beym Hiob am 19. und an die Römer am 8. Capitel: Ja es wird auch solches fürgestellet an unser jährlichen Verrichtung / wenn wir Saamen säen: Daß derselbe auffgehet und Frucht träget / ist / weil es ja so wenig in unserm Verstand und Macht stehet / ein so grosses Wunder / als daß die Menschen aufferstehen werden / weil es aber so gemein ist / und jährlich geschichet / wird es von wenigen in Betrachtung gezogen. Der heilige Apostel Paulus weiset selbst / in Fürstellung der Aufferstehung / auff dieses Fürbild und Gleichniß / da er 1. Cor. 15. spricht: Du Narr / das du säest wird nicht lebendig / es ersterbe denn. Wie nun der Same aus Göttlicher Einpflantzung / durch das Fiat, das ist: Es werde / eine verborgene Lebenskrafft in ihm hat: Also hat auch GOtt der HErr in dem Menschlichen Leibe eine wahre

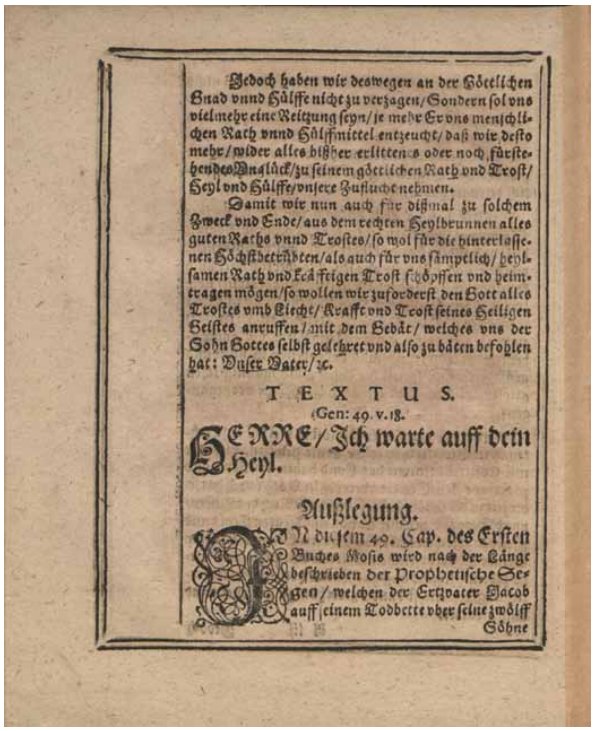
שוב hac ipsa vox, de reductione ad pulverem, in Textu adhibetur in designationem certè ejusdem hominis, qui ex pulvere creatus est.

quod probatur a. ex sacris literis. Iob. XIX, 25. Rom. VIII, 11.

β. è Naturæ lumine, adipiscimur, acrioribus oculis intueri textus suadet.

1. Cor. XV, 36. &c.

13 Text mit Marginalien und Kolumnentitel. Berlin, 1673. Ee 705-1160 S. 18


Jedoch haben wir derwegen an der Göttlichen Gnad vnnd Hülffe nicht zu verzagen / Sondern sol vns vielmehr eine Reitzung seyn / je mehr Er vns menschlichen Rath vnnd Hülffmittel entzeucht / daß wir desto mehr / wider alles bißher erlittens oder noch fürstehendes Vnglück / zu seinem göttlichen Rath vnd Trost / Heyl vnd Hülffe / vnsere Zuflucht nehmen.

Damit wir nun auch für dißmal zu solchem Zweck vnd Ende / aus dem rechten Heylbrunnen alles guten Raths vnnd Trostes / so wol für die hinterlassenen Höchstbetrübten / als auch für vns sämptlich / heylsamen Rath vnd kräfftigen Trost schöpffen vnd heimtragen mögen / so wollen wir zuforderst den Gott alles Trostes vmb Liecht / Krafft vnd Trost seines Heiligen Geistes anruffen / mit dem Gebät / welches vns der Sohn Gottes selbst gelehret vnd also zu bäten befohlen hat: Vnser Vater / ꝛc.

TEXTUS.
(Gen: 49. v. 18.
HERRE / Ich warte auff dein Heyl.

Außlegung.
IN diesem 49. Cap. des Ersten Buches Mosis wird nach der Länge beschrieben der Prophetische Segen / welchen der Ertzvater Jacob auff einem Todtbette vber seine zwölff Söhne

14 Text erscheint in Umrahmung ohne Kolumnentitel. Berlin, 1645. Ee 705-669 S. [8]

Den Hauptteil einer Funeralschrift bildet die Leichpredigt, die zunächst den Leichtext vorstellt und in den folgenden Teilen eine Vielzahl Bibelzitate enthält, welche in Form von Marginalien am Rand wiederholt werden (vgl. Abb. 13). Partiell direkt bezeichnet unterscheidet die Predigt Exordium (Vorrede, Einleitung in Redesituation), Propositio (Einleitung des Redegegenstands) und Consolatio (Trostworte). Das Ehrengedächtnis stellt üblicherweise die Schilderung des Lebenslaufs dar. Häufig wird an dessen Ende eine Schlussvignette als Kupferstich gesetzt, die Symbole der Vanitas zeigt (Schädel, Urnen; vgl. Abb. 20). Zumeist finden sich Epicedien gesammelt am Ende der Funeralschrift. Sind diese Beiträge bibliographisch selbständig erschienen, weisen sie ein eigenes Titelblatt auf. Oftmals sind diese Seiten ungezählt und fallen durch eine typographisch abweichende Gestaltung auf. Vor allem verschiedene Gedichte werden nach Nennung des Beiträgers hintereinander, zuweilen durch horizontale Linien voneinander getrennt, aufgeführt. An deren Ende erscheint eine weitere Schlussvignette.

Bis zur Mitte des 17. Jahrhunderts wurde häufig der gesamte Text doppelt gerahmt (Randleisten im Inneren des Drucks; vgl. Abb. 14). Anfangs waren Blätter und Seiten oft ungezählt, später erscheinen die Drucke des 17. Jahrhunderts mit Kopfzeilen (Kolumnentitel und Paginierung) in diesem Rahmen. Ab der zweiten Jahrhunderthälfte findet man durch horizontale Linien abgeteilte Kopfzeilen.

Sind die meisten Funeralschriften anfangs eher selten und gleichförmig geschmückt (Abbildung von Sarg und Schädel, wiederholte Verwendung gleicher Schlussvignetten mit Motiven[16], die in der Ikonographie vergangener Jahrhunderte in der Heraldik eine Rolle spielten; vgl. Abb. 19 u. 21), so zeigen sie ab 1650 deutlich mehr Buchschmuck mit vielfältigen Symbolen der Vanitas (Putti pusten Seifenblasen, qualmende Töpfe, ausgeblasene Kerzen; vgl. Abb. 15-18). Dieser über Kupferstiche realisierte Buchschmuck der Darstellung von Vergänglichkeit nennt des öfteren auch die Initialen des Künstlers. Tragen die Darstellungen im allgemeinen bis zur Mitte des 17. Jahrhunderts

[16] Beispielhaft soll der Wilde Mann genannt werden, dessen Darstellung auch als Fabelwesen (Halb Mensch, halb Tier) gedeutet werden kann und häufig in den Schlussvignetten des Druckers Runge vorkommt.

15 Gleitendes Schiff als Vanitassymbol. Oehringen, 1737. 1 in: Sh 6986

16 Putto mit qualmender Urne als Vanitassymbol. Dresden, 1713. Ee 705-6 S. 3

eher den Charakter des Buchschmucks, so muss man in der Folgezeit (beispielsweise bei Porträts) zunehmend von Illustrationen sprechen, die einen eigenständigen künstlerischen Beitrag darstellen. Sie wurden individuell für spezielle Drucke angefertigt.

Erscheinen Anfang des 17. Jahrhunderts Funeralschriften geringen Umfangs vorwiegend im Oktavformat, so ist ab 1650 und für das 18. Jahrhundert das Quartformat kennzeichnend. Zudem nehmen die Schriften an Umfang zu, eine vermehrte Spationierung (Vergrößerung horizontaler Zeichenabstände) und Gliederung der Texte ist zu beobachten. Auch größere Schriften kommen zum Einsatz.[17]

Die Entwicklung des Druckbilds, die Verwendung von Buchschmuck und andere typographische Besonderheiten im 17. Jahrhundert wurden vor allem anhand der Funeralschriften der Familie Runge (Georg, Christoph und ihre Witwen) untersucht. Ihre Drucke sind zahlenmäßig stark und zeitlich kontinuierlich im Bestand der SBB vorhanden.

17 Vielfältige Vanitassymbole. Gotha, 1701. Ee 705-44 S. [36]

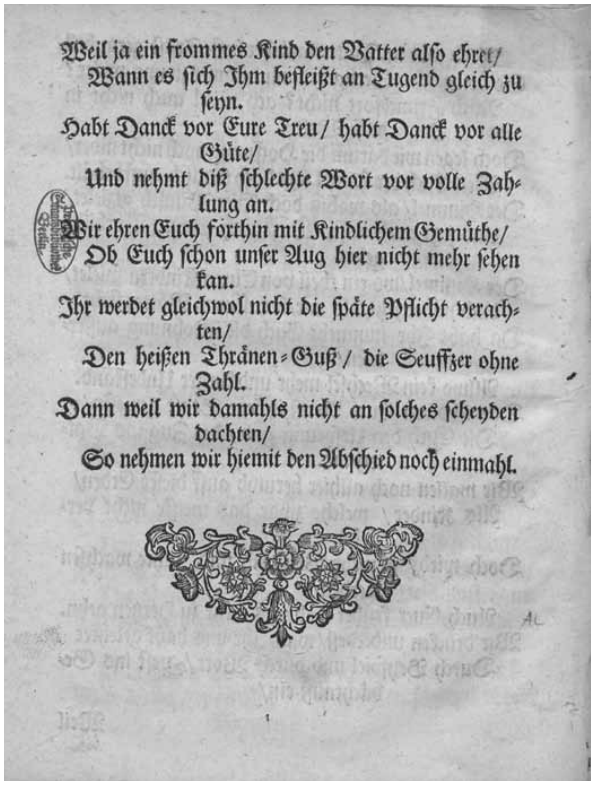
18 Putto mit Schädel u. Seifenblasen als Vanitassymbol. Leipzig, 1705. Ee 705-21 S. 3

[17] Aussagen zu Buchgewerbe und Druckqualität im 17. Jahrhundert finden sich unter vielfältiger Quellenangabe in KILLIUS 1999, S. 90 ff.

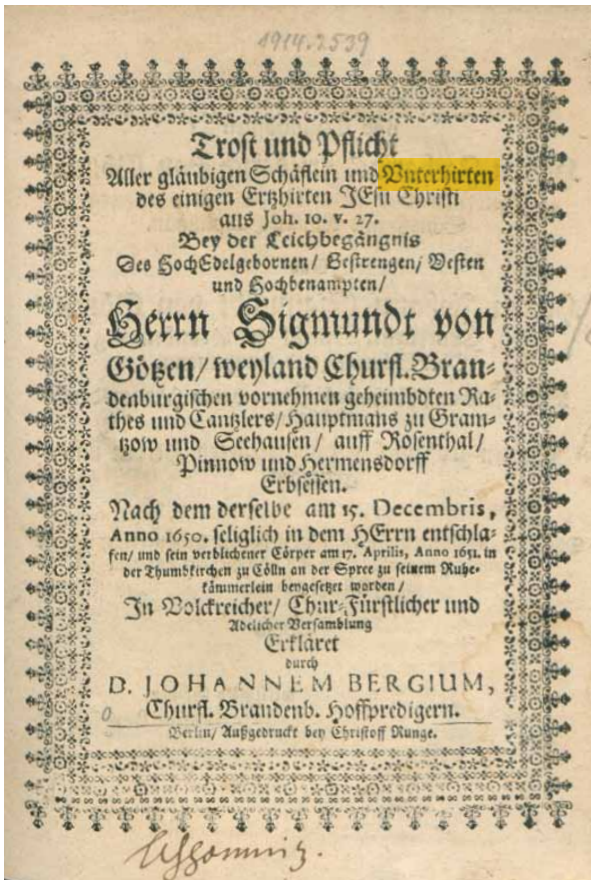
19 Schlussvignette mit Darst. „wilder Mann“. Berlin, 1664. Ee 705-1592 S. 78

20 Schädel als Vanitassymbol. Berlin, 1674. Ee 705-1159 S. 40

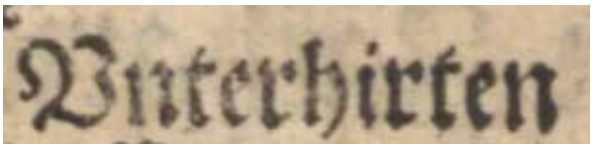
21 Sarg mit Schädel als Vanitassymbol. Berlin, 1666. Ee 705-1590 S. [150]


Weil ja ein frommes Kind den Vatter alſo ehret/
Wann es ſich Ihm befleißt an Tugend gleich zu ſeyn.
Habt Danck vor Eure Treu/ habt Danck vor alle Güte/
Und nehmt diß ſchlechte Wort vor volle Zahlung an.
Wir ehren Euch forthin mit Kindlichem Gemüthe/
Ob Euch ſchon unſer Aug hier nicht mehr ſehen kan.
Ihr werdet gleichwol nicht die ſpäte Pflicht verachten/
Den heißen Thränen-Guß / die Seufftzer ohne Zahl.
Dann weil wir damahls nicht an ſolches ſcheyden dachten/
So nehmen wir hiemit den Abſchied noch einmahl.

22 Schlussvignette mit Blumendarstellung. Straßburg, 1699. Ee 705-159 S. [32]



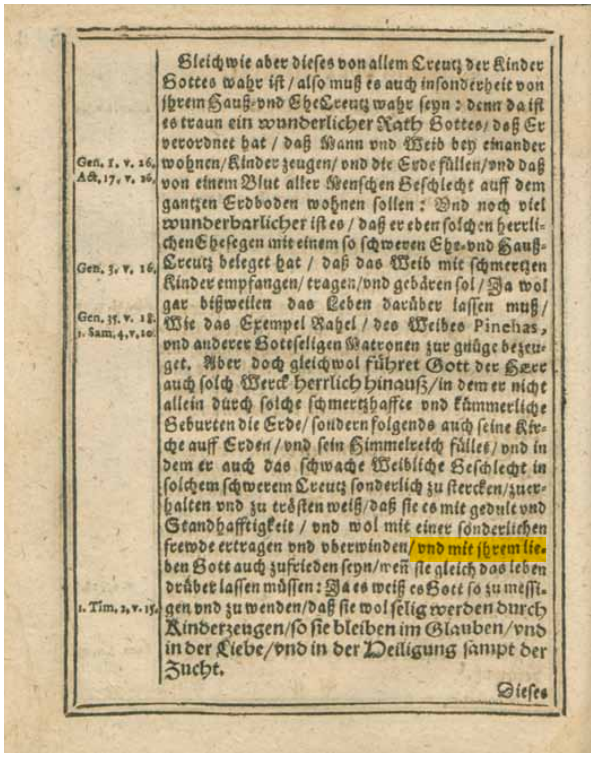
23 Verwendung von „V“ als „U“. Berlin, 1651. 11 in: Ee 1485 Titelblatt



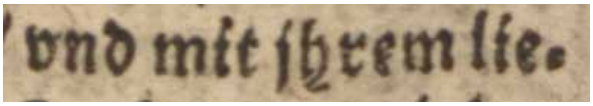
23 Markierter Ausschnitt

2.4 Orthographische Besonderheiten

Besonderheiten in der deutschen Rechtschreibung, die ohnehin für das 17. und 18. Jahrhundert nicht einheitlich war und starken Abweichungen unterlag, müssen hervorgehoben werden, auch wenn sie ein allgemeines Phänomen darstellen und nicht auf Funeralschriften beschränkt sind.[18] Bekanntermaßen findet man in den Drucken der ersten Hälfte des 17. Jahrhunderts überwiegend noch keine Unterscheidung von u und v. So lässt sich im Satz feststellen, dass ein gesprochenes „u“ am Wortanfang als „v“ gesetzt wird, im Wortinnern jedoch als „u“. Am häufigsten fällt diese Verwendung beim Wörtchen „vnd“ auf. Auch bei Großbuchstaben wird diese Schreibregel angewandt. Zu nennen wären hier Wörter wie „Vnser“ oder „Vnterthan“. Differenziert man zunächst in



24 Verwendung von „j“ in „jhr“. Berlin, 1633. an: 50 MA 42058 S. 511

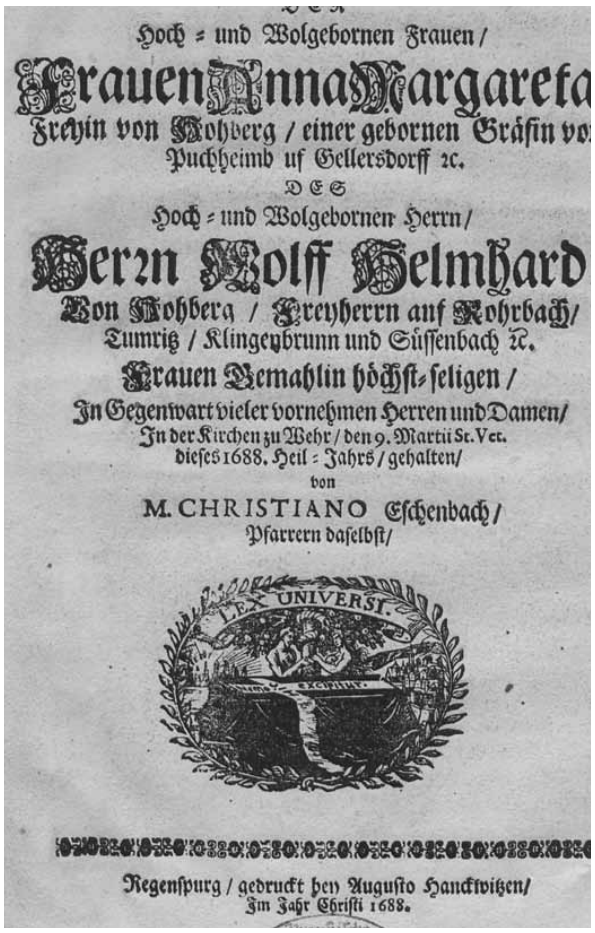


24 Markierter Ausschnitt

der zweiten Jahrhunderthälfte die Kleinbuchstaben „u“ und „v“ konsequent lautbezogen, so dauert es bei den Großbuchstaben einige Jahrzehnte länger. Es finden sich reichlich Vorlagen mit dem Gebrauch der alten Regel für Großbuchstaben; parallel werden im Text die Kleinbuchstaben jedoch schon eindeutig verwendet (vgl. Abb. 23).

In diesem Zusammenhang ist auf die ähnliche Problematik bei i/j zu verweisen, wobei hier das große „J“ durchgängig für I oder J in Frakturtexten steht. Hingegen kommt es bei den Kleinbuchstaben längere Zeit zu einem gemeinsamen Gebrauch von „i“ und „j“ am Wortanfang. Auf einer Seite von 1625 beispielsweise findet man in derselben Schrift Worte wie „jhrer“,“jhn“ und „in“, „im“, „inbrünstiglich“, „ist“ gemeinsam, was auf eine Unterscheidung im Gebrauch des langen und kur-

[18] Allen Drucken gleich ist die Darstellung der Umlaute durch Vokale mit übergesetztem kleinen „e“. Typisch für lange Zeit ist ebenso die Gliederung der Texte durch Virgel; Kommata kommen erst später in der Frakturschrift in Gebrauch. Weitere Merkmale historischer Drucke sind Kustoden und Bogensignaturen, Hilfsmittel für den Buchbinder. Vgl. weiterführend: SCHNEIDER und ZIPPEL 2009. Zur Varianz der deutschen Orthografie siehe VOESTE 2008.


Hoch = und Wolgebornen Frauen /
Frauen Anna Margareta
Freyin von Wohberg / einer gebornen Gräfin von
Puchheimb uf Gellersdorff ꝛc.
DES
Hoch = und Wolgebornen Herrn /
Herrn Wolff Helmhard
Von Wohberg / Freyherrn auf Rohrbach /
Tumritz / Klingenbrunn und Süssenbach ꝛc.
Frauen Gemahlin höchst = seligen /
In Gegenwart vieler vornehmen Herren und Damen /
In der Kirchen zu Wehr / den 9. Martii St. Vet.
dieses 1688. Heil = Jahrs / gehalten /
von
M. CHRISTIANO Eschenbach /
Pfarrern daselbst /
LEX UNIVERSI
Regenspurg / gedruckt bey Augusto Hanckwitzen /
Im Jahr Christi 1688.

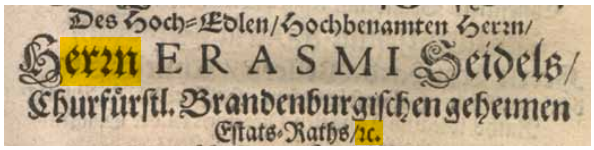
27 Titelblatt mit verschnörkelten Versalien, Regenspurg, 1688. Ee 705-1081


Jobitisch = Brandensteinische
Glaubens = volle Hoffnung zu GOTT
Bey dem letzten
Ehren = Gedächtniß
Des Weyland
Reichs = Frey = Hoch = Wohlgebohrnen Herrn,
HERRN
Hanß Ernsts
von
Brandenstein
Auf Zöschen, Zschepen und Selben
Erb = Lehn = und Gerichts = Herrn,
Sr. Hoch = Fürstl. Durchl. zu Sachsen = Merse-
burg hoch = bestalten Cammer = Junckers,
Welcher am 22. Octobr. 1716.
Abends zwischen 8. und 9. Uhr nach GOttes Willen im 37. Jahr seines Alters
in seinem Gott und Erlöser sanfft und seelig in Zschepen entschlaffen, und der Seelen nach ins
himmlische Freuden = Reich versetzet ist,
In der am 10. Decembr. St. N. 1716.
gehaltenen Leichen = Predigt,
Aus dem Buch Hiobs am 19. Cap. v. 25. 26. und 27.
Ich weiß, daß mein Erlöser lebet, ꝛc.
Der hohen Trauer = Versammlung vorgestellet,
Von
Johanne Martino Mellen,
Pfarrern in Selben und Zschepen.

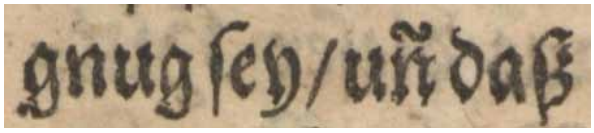
28 Titelblatt mit illustrierten Versalien, Delitzsch, 1716. Ee 705-167

zen i in der Schriftsprache hindeutet und so über mehrere Jahrzehnte zu beobachten ist bis auch hier immer „i“ geschrieben wird, wenn es sich um ein solches handelt (vgl. Abb. 24)[19].

Weitere Auffälligkeiten entstehen durch die Verwendung des runden „r“[20]. Zunächst verwendet für die Abkürzung „etc.“, erscheint ein rundes „r“ gefolgt von einem „c“ und einem Punkt. Später verbreitet sich der Satz des runden „r“ bei Buchstabenverdopplung (Herr, herrlich, Herrn) oder auch im Zusammentreffen mit „t“ (aufgefordert; vgl. auch Abb. 25). Zusätzlich uneindeutig können Abbreviaturen (zumeist in Gestalt von Nasalstrichen) über dem Buchstaben „n“ oder „m“ sein. Überwiegend zeigen diese eine Verdopplung an; es gibt aber auch andere Anwendungen wie bei


Des Hoch=Edlen/Hochbenamten Herrn/
Herrn ERASMI Seidels/
Churfürstl. Brandenburgischen geheimen
Estats=Raths/ꝛc.

25 Rundes „R“ bei Buchstabenverdopplung u. in „etc.“. 7 in: Ee 1593-2 Titelblatt


gnug sey/uñ daß

26 Nasalstrich über „n“ zeigt Verdopplung an. 5 in: Ee 1593-2 S. 72

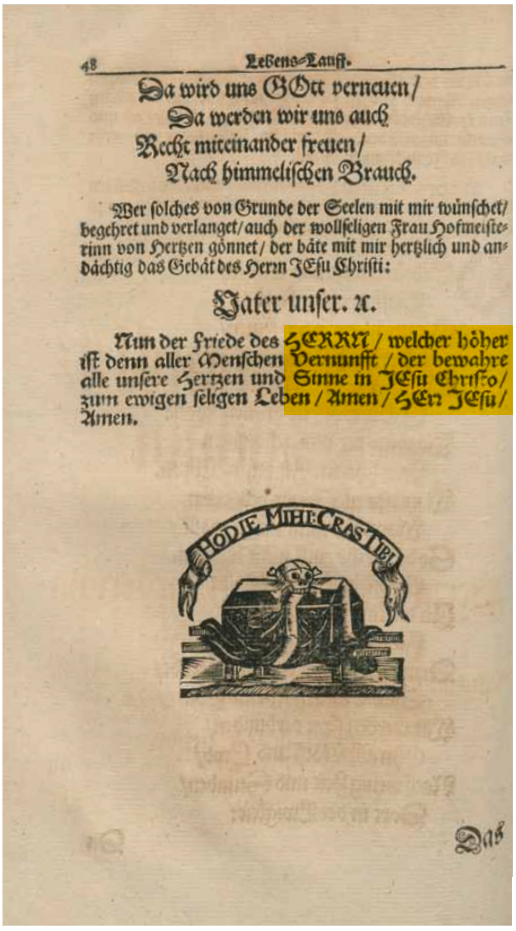
„un“; welches zu „und“ aufzulösen ist (vgl. Abb. 26). In lateinischen Zitaten sind natürlich weitere Abbreviaturen anzutreffen, deren Auflösung wort- oder wortstammbezogen vorzunehmen ist. Für den geübten Leser ist das meist kein Problem; für die Anwendung automatischer Verfahren der Texterkennung tritt es erschwerend hinzu.

Auch die Groß- und Kleinschreibung soll Erwähnung finden, da sie für die Schriftenvielfalt eine bedeutende Rolle spielt. So schreibt Kapr sehr treffend: „Leider blieb das schnörkelhafte Wesen der barocken Schriftformen nicht ohne Einfluss auf die deutsche Rechtschreibung. Der Wunsch zur Auszeichnung drängte die Schreiber dazu, möglichst viele Wörter mit Großbuchstaben zu schreiben, die mehr Gelegenheit zu schwungvollen Variationen gaben. … Erst begann man GOTT und HERR groß zu schreiben, dann folgten Papst, Kaiser, Bischof, Kirche, Fürst, und die Ergebenheitsfloskeln in den Vorreden der Buchausgaben des Barocks taten das übrige zur Überhäufung des deutschen Schriftbildes mit Versalien.“[21]

[19] Vgl. weiterführend HORN 1894

[20] „Rundes r; entstanden aus der rechten Hälfte des Unzial-R. Es wurde hinter runde Buchstaben oder als zweites r gesetzt; verlor aber im 19. Jh. an Bedeutung und verschwand.“ REHSE 1998 S. 78

[21] KAPR 1959 S. 61


48 Lebens-Lauff.

Da wird uns GOtt vernewen/
Da werden wir uns auch
Recht miteinander freuen/
Nach himmelischen Brauch.

Wer solches von Grunde der Seelen mit mir wünschet/ begehret und verlanget/ auch der wollseligen Frau Hofmeisterinn von Hertzen gönnet/ der bäte mit mir hertzlich und andächtig das Gebät des Herrn JEsu Christi:

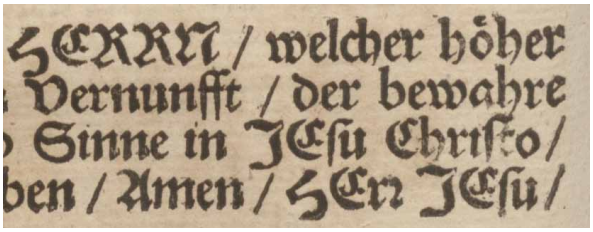
Vater unser. ꝛc.

Nun der Friede des HERRN/ welcher höher ist denn aller Menschen Vernunfft/ der bewahre alle unsere Hertzen und Sinne in JEsu Christo/ zum ewigen seligen Leben/ Amen/ HErr JEsu/ Amen.

HODIE MIHI CRAS TIBI

Das

29 Häufung von Versalien. Berlin, 1683. 3 in 4° Ee 660 S. 48


HERRN/ welcher höher
Vernunfft/ der bewahre
Sinne in JEsu Christo/
ben/ Amen/ HErr JEsu/

29 Markierter Ausschnitt


herausextrahiren

30 Ein in Antiqua gesetztes Fremdwort hat Fraktur-Suffix

So erkennt man an der Entwicklung der Titelblätter der Funeralschriften neben der Neigung zur Verwendung möglichst vieler Schriften eine Zunahme verschnörkelter Versalien bis hin zur Verwendung von Barockinitialen (vgl. Abb. 27 u. 28). Offenbar war es auch im Text nicht ausreichend, Wörter wie GOtt und HErr oder JEsu durch Großschreibung hervorzuheben; meist waren sie auch noch in einer größeren Schrift gesetzt (vgl. Abb. 29). Auch der Satz von Kapitälchen wird für Hervorhebungen genutzt. Üblich ist durchaus auch die Hervorhebung nur einer Silbe oder gar das Setzen eines Fremdworts in zwei Schriften (Wortstamm Antiqua, Endung Fraktur; vgl. Abb. 30).

2.5 Verwendung gebrochener Schriften

In den hier betrachteten Funeralschriften werden überwiegend gebrochene Schriften für die laufenden Texte verwendet. Es handelt sich vornehmlich um verschiedene Arten von Frakturschriften ergänzt um die Schwabacher Schrift, die sichtbar für Hervorhebungen (v. a. von Namen) eingesetzt wurde. Ferner finden Antiquaschriften für lateinischsprachige Zitate in verschiedenen Graden und Schnitten Verwendung.

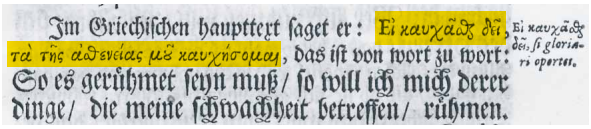
Die heutige Definition dieser Schriften (oft verkürzt auch nur als Frakturschriften[22] bezeichnet) macht sich vor allem an den Brechungen der einzelnen Buchstaben fest, hat sie doch ihren Ursprung in der gotischen Textura – einer Schrift ohne jegliche Rundungen. Nachdem in der Renaissance vor allem die aus ihr entstandene offenere und leichter lesbare Schwabacher Schrift Verbreitung fand, ging man in der Mitte des 16. Jahrhunderts dazu über, schlankere Buchstaben mit klarerer Trennung und weniger Verzierungen zu verwenden: die Fraktur war entstanden. Unterschiede zwischen den vorstehend genannten Schriften findet man beim Vergleich der Punzen (Innenraum von Buchstaben). Zeigen gotische Schriften durchweg eckige und schmale Punzen in Form von verschobenen Rechtecken bei den Kleinbuchstaben, so haben Frakturschriften abgerundete Punzen, z. B. beim b, d, o und h. Bei der Schwabacher bilden die Punzen der Kleinbuchstaben Oval und Halboval.[23] Frakturschriften laufen recht eng, d. h. sie haben kleine Wortzwischenräume.

Ein harmonisches und freundliches Schriftbild entsteht aus der Grauwirkung der Fraktur. Kapr beschreibt sie wie folgt: „Das o, das a und alle daraus entwickelten Kleinbuchstaben bestehen aus einem geraden und einem geschwungenen Abstrich mit relativ spitzem Fuß. ... geschnäbelte Oberlängen der Kleinbuchstaben, und die sogenannten Elefantenrüssel, die den Buchstaben einleitenden Schwünge, geben den Versalien A, B, M, N, P, V und W ihren besonderen Ausdruck. [Vgl. Abb. 33 u. 34] Die Versalien sind im allgemeinen breit, die Gemeinen jedoch schmal gehalten. Der Gesamteindruck ist

[22] Die umfassendere Definition der Fraktur ist belegt in Adelung FRACTUR 1808 bzw. im großen Brockhaus FRAKTUR 1954.
[23] Vgl. REHSE 1998 S. 15-21

schwungvoll und reich bewegt, er entspricht unserer konsonantenreichen deutschen Sprache und gestattet das schnelle Erfassen ihrer verhältnismäßig langen Wörter.“[24] „Obwohl die Fraktur viele Ausschmückungen erfahren hat, sind die ursprünglichen Federzüge stets erkennbar geblieben. Vier Jahrhunderte lang war die Fraktur die am meisten verbreitete Schrift in Deutschland. Nahezu alle Bücher, sogar wissenschaftliche, fast alle Tageszeitungen und Zeitschriften, in der Regel aber auch Akzidenzen (Geschäfts- und Kleindrucksachen) wurden in Frakturschrift gedruckt“[25], wie Rehse weiter ausführt. Als besondere Frakturschrift des ausgehenden 17. Jahrhunderts ist die Luthersche Fraktur zu nennen, die Kapr charakterisiert: „Die Großbuchstaben sind teilweise mit phantasievollen doch überflüssigen Schnörkeln kalligrafisch moduliert. Aber das Hervorstechende bei der Luther-Fraktur ist der Einfluß des Stichels, der den geschriebenen Duktus maßvoll korrigiert.“[26]

Von den oben genannten Schriften finden sich vor allem die verschiedensten Frakturschriften in den betrachteten Leichenpredigten; Zahlen erscheinen immer als Mediävalzahlen, also mit Ober- und Unterlängen. Je nach Entstehungszeit stehen am Anfang eher schmucklose Schriften, die in verschiedenen Schriftgraden kombiniert werden. Auszeichnungen finden zunächst fast immer in der Schwabacher Schrift statt. Im laufenden Text springen derlei Zitate sofort ins Auge. Im Barock hingegen erscheinen viele Frakturschriften, die zunehmend verschnörkelte Versalien, aber schließlich auch verzierte Kleinbuchstaben aufweisen. Auszeichnungen werden nun durch größere und fettere Schriftgrade oder abweichende Frakturschriften erzielt; die Schwabacher Schrift tritt in ihrer Verwendung in den Hintergrund. Wie eingangs bereits erwähnt, kommen zunächst kleinere Schriften (12 pt) zum Einsatz. Im ausgehenden 17. und beginnenden 18. Jahrhundert hingegen findet man oft 16-pt- oder gar 18-pt-Grundschriften auch für längere Texte. Unterbrochen wird das Druckbild im Stile der Zeit durch in Antiqua gesetzte Versalien für Namen oder in Antiqua (auch kursiver Antiqua) gesetzte lateinische Zitate. Bibelzitate sind auch direkt in hebräischer oder griechischer Schrift wiedergegeben (vgl. Abb. 31). Hervorzuheben sind hier ebenfalls die mannigfaltigen Formen und Größen von Initialen, deren Inhalt oft nur im Kontext gelesen und verstanden werden kann (vgl. Abb. 32). Das Zusammentreffen all dieser Schriften führt zu einer großen Vielfalt, die eine Gruppierung von Drucken außer nach ihrem Entstehungszeitraum sehr erschwert. Möglichkeiten der Gruppierung eröffnen vornehmlich die Großbuchstaben M, I, C und U, deren Formen und Elefantenrüssel sich oft von Schrift zu Schrift unterscheiden. Allerdings ist ihr Auftreten im Verhältnis zu den Kleinbuchstaben eher selten, so dass längere Textabschnitte gelesen werden müssen. Es bleibt abzuwarten, ob sich ein solches Vorgehen als praktikabel erweist.


Im Griechiſchen haupttext ſaget er: Εἰ καυχᾶσ δεῖ, τὰ τῆς ἀσθενείας μου καυχήσομαι, das iſt von wort zu wort: So es gerühmet ſeyn muß / ſo will ich mich derer dinge / die meine ſchwachheit betreffen / rühmen. Ei καυχᾶσ δεῖ, ſi gloriari oportet.

31 Zitate in griechischer Sprache

32 Initiale, Leipzig, 1691. Ee 705-32 S. 3

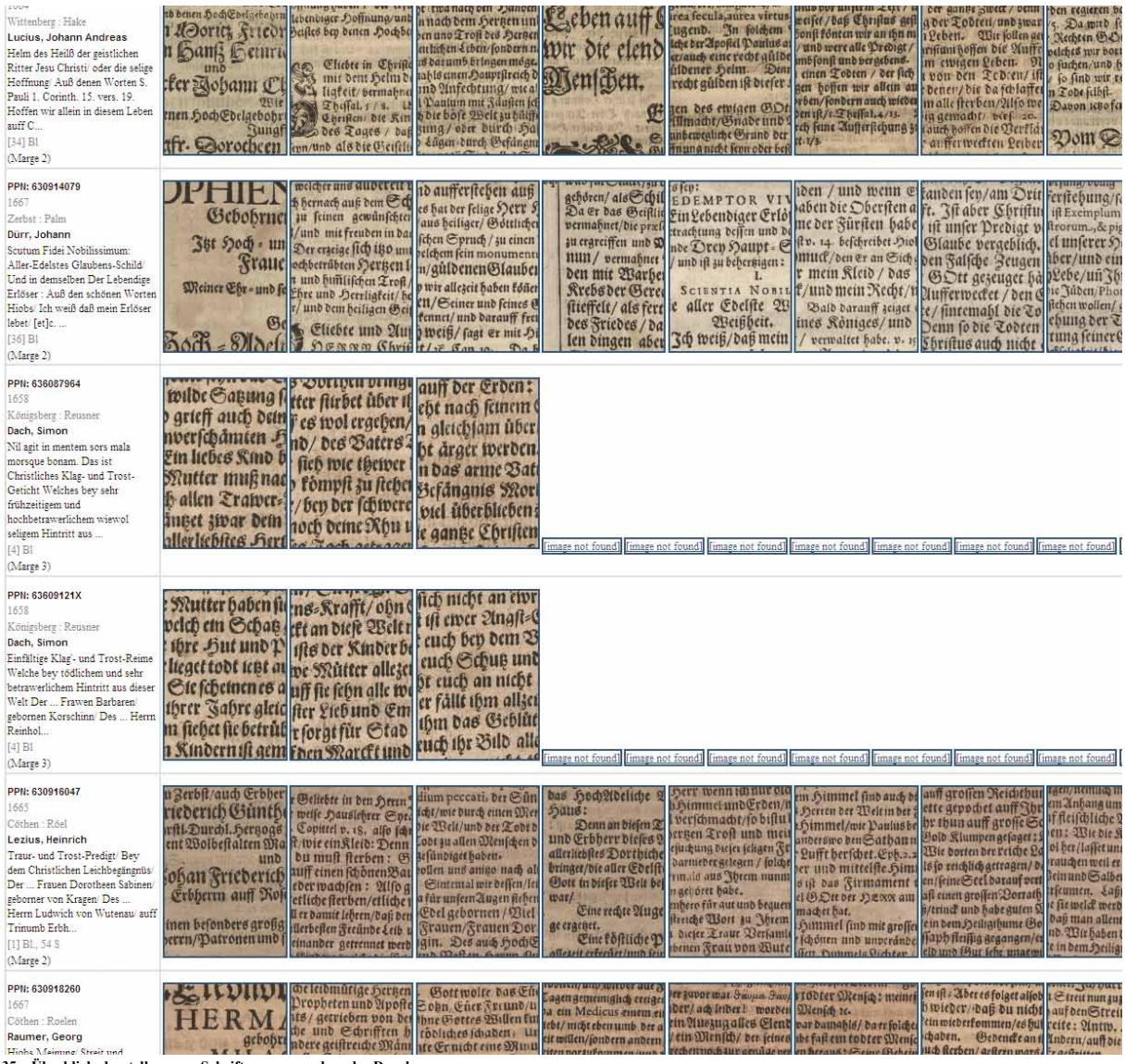
33 Vorkommende Muster für Versalie A

34 Vorkommende Muster für Versalie M

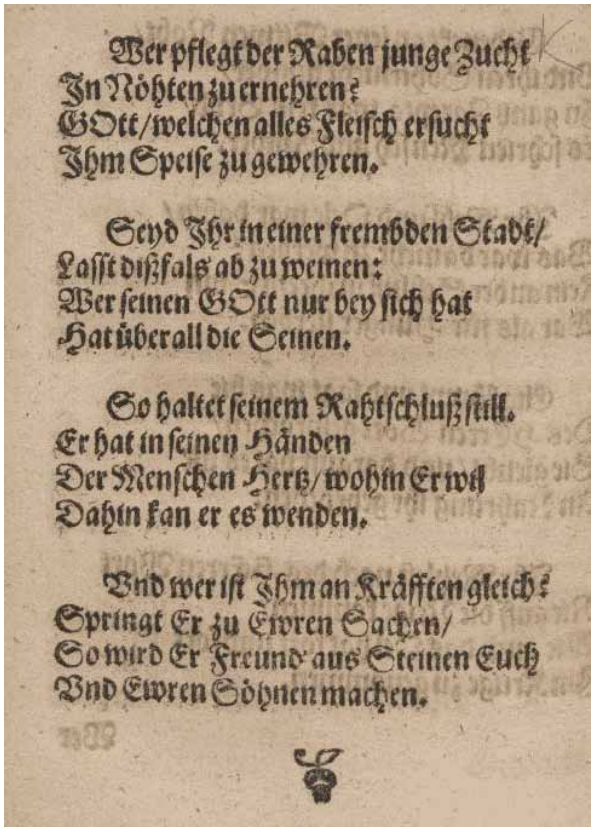
[24] KAPR 1959 S. 49
[25] REHSE 1998 S. 21
[26] KAPR 1959 S. 59

2.6 Diversität vorkommender Schrifttypen

Eine der Schwierigkeiten bei der OCR Alter Drucke besteht in den verwendeten Typen, die von der speziell in den FineReader-Modulen bereits gut erschlossenen Frakturschrift des 19. Jahrhunderts deutlich abweichen. Es sollte daher versucht werden, zum Material passende Muster zu trainieren. Um die Tests unter Laborbedingungen möglichst gut interpretieren zu können, wurde dabei mit einer sehr homogenen Schriftart begonnen. Diese Homogenität zeigte sich in einem Teil der in Königsberg gedruckten Funeralschriften des bekannten Dichters Simon Dach. Wurde diese erste aus 112 Drucken bestehende Marge fast ausschließlich aus Trauergedichten gebildet, so war eine zweite Schriftgruppe etwas offener angelegt. Sie konzentrierte sich aber auch auf denselben Zeitraum, wenige Drucker und zwei dominante Schriftgrößen. Jene Marge bestand aus 141 Leichenpredigten, die Schriften mit hoher Ähnlichkeit aufwiesen, allerdings auch im deutschsprachigen Fließtext mit etlichen Einsprengseln davon abweichender Typen. Um eine schnelle Zuordnung zu erreichen, wurden automatisch und stichprobenartig Teilbild-Serien erzeugt und visuell in Gruppen aufgeteilt (vgl. Abb. 35 u. 36).



35 Überblicksdarstellung zur Schriftgruppenanalyse der Drucke

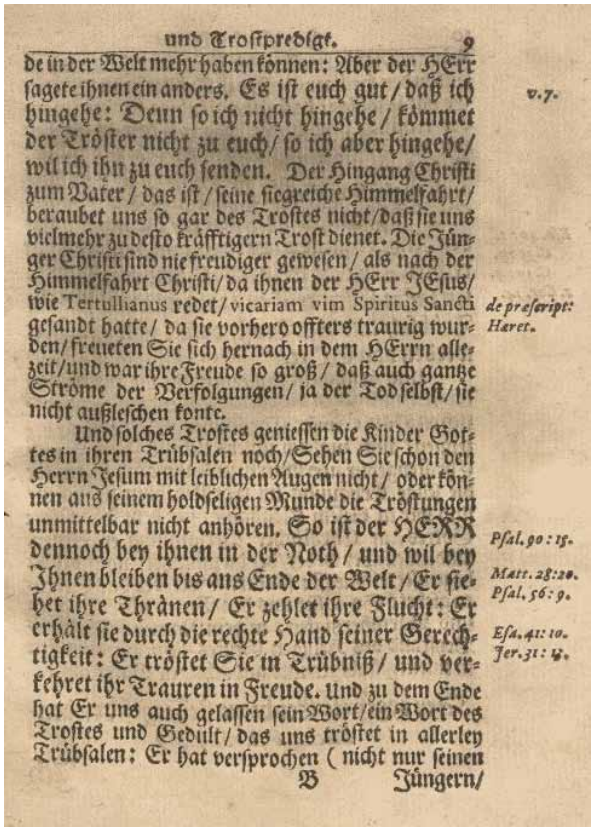


Wer pflegt der Raben junge Zucht
In Nöhten zu ernehren?
GOtt/welchen alles Fleisch ersucht
Ihm Speise zu gewehren.

Seyd Ihr in einer frembden Stadt/
Lasst diß fals ab zu weinen:
Wer seinen GOtt nur bey sich hat
Hat überall die Seinen.

So haltet seinem Rahtschluß still.
Er hat in seinen Händen
Der Menschen Hertz/wohin Er wil
Dahin kan er es wenden.

Vnd wer ist Ihm an Kräfften gleich?
Springt Er zu Ewren Sachen/
So wird Er Freund aus Steinen Euch
Vnd Ewren Söhnen machen.



de in der Welt mehr haben können: Aber der HErr sagete ihnen ein anders. Es ist euch gut/daß ich hingehe: Denn so ich nicht hingehe/kömmet der Tröster nicht zu euch/so ich aber hingehe/ wil ich ihn zu euch senden. Der Hingang Christi zum Vater/das ist/seine siegreiche Himmelfahrt/ beraubet uns so gar des Trostes nicht/daß sie uns vielmehr zu desto kräfftigern Trost dienet. Die Jünger Christi sind nie freudiger gewesen/als nach der Himmelfahrt Christi/da ihnen der HErr JEsus/ wie Tertullianus redet/vicariam vim Spiritus Sancti gesandt hatte/da sie vorhero offters traurig wurden/freueten Sie sich hernach in dem HErrn allezeit/und war ihre Freude so groß/daß auch gantze Ströme der Verfolgungen/ja der Tod selbst/sie nicht außleschen konte.

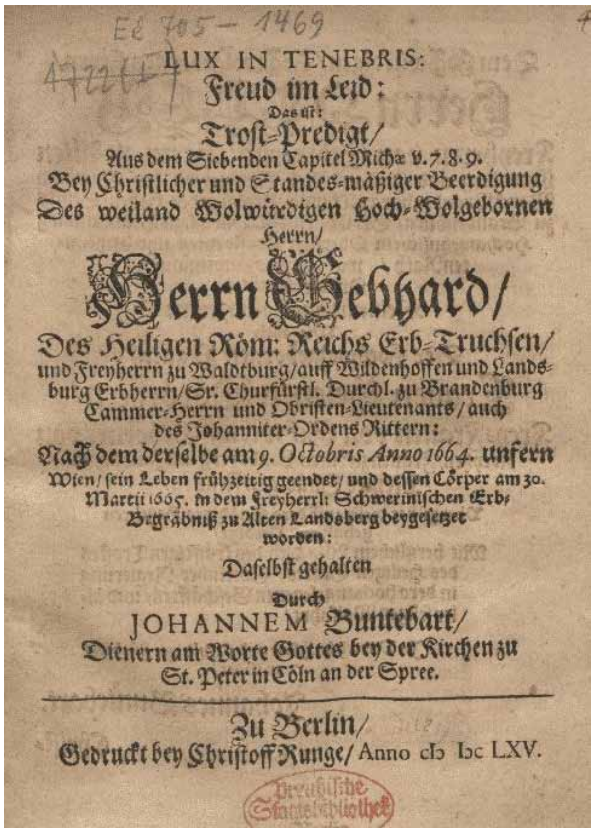
Und solches Trostes geniessen die Kinder Gottes in ihren Trübsalen noch/Sehen Sie schon den Herrn Jesum mit leiblichen Augen nicht/oder können aus seinem holdseligen Munde die Tröstungen unmittelbar nicht anhören. So ist der HERR dennoch bey ihnen in der Noth/und wil bey Ihnen bleiben bis ans Ende der Welt/Er siehet ihre Thränen/Er zehlet ihre Flucht: Er erhält sie durch die rechte Hand seiner Gerechtigkeit: Er tröstet Sie in Trübniß/und verkehret ihr Trauren in Freude. Und zu dem Ende hat Er uns auch gelassen sein Wort/ein Wort des Trostes und Gedult/das uns tröstet in allerley Trübsalen: Er hat versprochen (nicht nur seinen

v.7.

de præscript: Hæret.

Psal. 90: 15.
Matt. 28: 20.
Psal. 56: 9.
Esa. 41: 10.
Jer. 31: 4.

B Jüngern/



LUX IN TENEBRIS:
Freud im Leid:
Das ist:
Trost-Predigt/
Aus dem Siebenden Capitel Michæ v. 7. 8. 9.
Bey Christlicher und Standes-mäßiger Beerdigung
Des weiland Wolwürdigen Hoch-Wolgebornen
Herrn/
Herrn Gebhard/
Des Heiligen Röm: Reichs Erb-Truchsen/
und Freyherrn zu Waldtburg/auff Wildenhoffen und Landsburg Erbherrn/Sr. Churfürstl. Durchl. zu Brandenburg Cammer-Herrn und Obristen-Lieutenants/auch des Johanniter-Ordens Rittern:
Nach dem derselbe am 9. Octobris Anno 1664. unfern Wien/sein Leben frühzeitig geendet/und dessen Cörper am 30. Martii 1665. in dem Freyherrl: Schwerinischen Erb-Begräbniß zu Alten Landsberg beygesetzet worden:
Daselbst gehalten
Durch
JOHANNEM Buntebart/
Dienern am Worte Gottes bey der Kirchen zu St. Peter in Cöln an der Spree.

Zu Berlin/
Gedruckt bey Christoff Runge/Anno ↄ Ↄc LXV.

Versuch der Gruppierung

– ***nach Merkmalen***

Zur Gruppierung, also Clusterung, wurden unter typenkundlicher Beratung bestimmte individuell verwendete markante Großbuchstaben als Unterscheidungsmerkmal erwogen, sowie einige weniger markante, im Text aber durchgängiger auftretende Kleinbuchstaben. Um für eine effektive OCR noch hinreichend große Mengen von Bildern gemeinsam verarbeiten zu können, konnte und brauchte die Aufteilung andererseits nicht beliebig weit getrieben zu werden. Eine erste Sortierung innerhalb der Drucke aus der Mitte des 17. Jahrhunderts trennte (außer nach Schriftgröße) nach deutlich unterscheidbaren Formen des kleinen runden „s“ (mit Rüssel wie in Fraktur und Textur oder ohne wie in Schwabacher und Rundgotisch, gleich oder ungleich gewichtete vertikale Hälften usw.; vgl. Abb. 37). Für in weiteren Versuchen hinzukommende spätere Drucke eigneten sich etwa der Übergang von Virgeln zu Kommata im Satz oder – bei in das späte 18. Jahrhundert hineinreichenden Teilbeständen – eine bereits merkliche Annäherung an die schlankere Fraktur des 19. Jahrhunderts als initiales Unterscheidungsmerkmal für separat zu trainierende Teilbestände. Im letzten Fall (zunehmende Nähe zu Schriften des 19. Jahrhunderts) spricht viel für den Verzicht auf eigenes Training bei zunehmender Brauchbarkeit der in Softwareprodukten (z. B. in der FineReader-Engine) bereits integrierten Muster (Vgl. Kap. 5).

37 Verschiedene kleine „s“ als Gruppierungsmerkmal

– ***alternativ: Gruppierung empirisch***

Sobald erste trainierte Muster schon vorliegen, bietet es sich in der Praxis auch an, die Zuordnung von Bildern zu einem Musterbestand nicht vorab nach visuellen Merkmalen, sondern nach einem OCR-Gang empirisch anhand der Ergebnisse zu beurteilen. Im Idealfall liegen für Teile des zu prozessierenden Bestands Transkriptionen oder andere Ground-Truth-Daten (manuell erfasste, auch semantisch ausgezeichnete fehlerfreie Daten) vor, so dass für diese eine echte Erkennungsrate ermittelt und verglichen werden kann. Meist wird solches Referenzmaterial aber fehlen, so dass – mit vorsichtiger Interpretation und je nach Güte der vorliegenden Wörterbücher – hilfsweise z. B. der Anteil der sowohl im Lexikon als auch in den OCR-Texten enthaltenen Wörter als Indikator dienen kann. Wo dieser Anteil deutlich sinkt, sind oft Schriftartwechsel, Sprachwechsel und dergleichen die Ursache; im getesteten Material waren z. B. sporadische Seiten mit deutlichem Antiqua-Anteil auf diese Art schnell nachträglich zu erkennen. Praktisch sinnvoll wird diese nachträgliche Zuordnung immer dann sein, wenn Rechenzeit für Mehrfach-OCR zur Verfügung steht und die Bearbeiter die nachträgliche Auswahl im Vergleich zu einer vorherigen Sichtung jeder Seite schneller vornehmen können (Vgl. Kap. 4.5).

3 Typische OCR-Teilschritte und Einflussgrößen
3.1 Typischer Workflow

Für den Zweck der Darstellung des Softwaretests soll die Verarbeitungsschrittfolge wie folgt vergröbert (vgl. Abb. 38) behandelt werden:[27]

- Sichtung und Vorsortierung des Bildmaterials;
- Binarisierung (inkl. vorheriger Bearbeitung des Farb- bzw. Graustufenbilds und Nachbearbeitung des bitonalen Bilds);
- Segmentierung (Layoutanalyse, Blockerkennung, Segmentierung auf Wort- und Zeichenebene);
- OCR (inkl. unmittelbar verbundener Fehlerkorrekturmechanismen);
- Lexikalische Nachkorrektur[28].

Neben den Einzelschritten begrenzt nicht zuletzt die stapelweise Verarbeitung an sich die mögliche Erkennungsqualität.

Dieser Workflow setzt das Vorhandensein einsatzbereiter OCR-Musterbibliotheken voraus. Wenn diese nicht oder nicht passend bereitstehen, ist deren Erstellung bzw. Bearbeitung voranzuschicken. Im praktischen Softwarevergleich (Vgl. Kap. 4) wird dieser Arbeitsgang separat behandelt.

3.1.1 Bereitstellung des Bildmaterials

Stufeninput und -output

Eingang: Bildmaterial

Ausgang: Bildmaterial selektiert nach Eignung, z. B. gruppiert in Margen, die gemeinsame OCR-Läufe zulassen

Typische Informationsverluste und Fehlerquellen

Unvermeidlich:

a) Das menschliche Vorwissen über Textsorten geht i. d. R. nicht in den OLR-/OCR-Prozess mit ein.
b) Schlechte Vorlagen aufgrund von Druckqualität, Durchdruck, Papierverfärbung, Verschmutzungen und Annotationen müssen in gewissem Maße hingenommen werden.

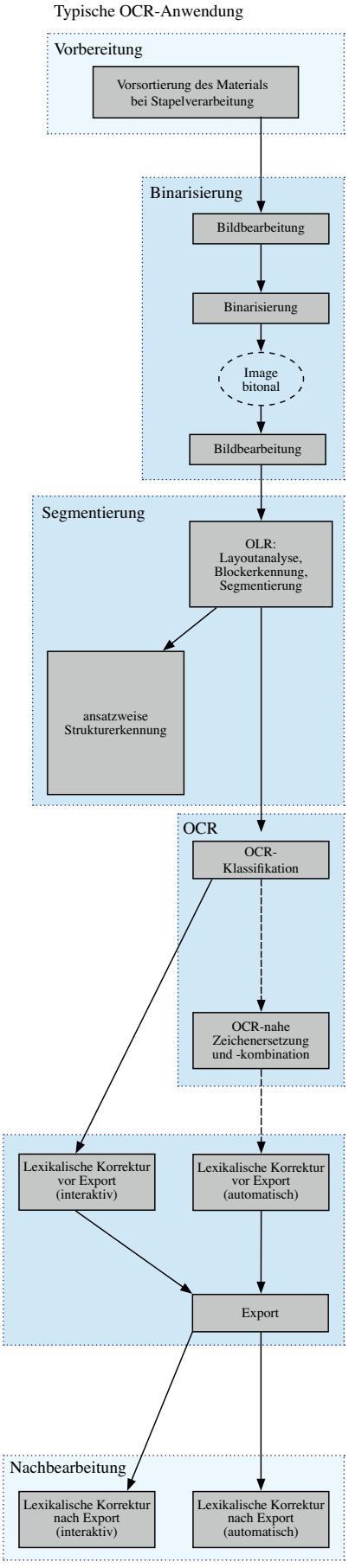
Potentiell beeinflussbar:

c) Die Scans schwanken in Bildhelligkeit und -kontrast sowie Randmaßen.
d) Innerhalb des OCR-Laufs (Seite, Stapel) kommt es zum Wechsel von Schriftart, Sprache oder typographischen Konventionen.
e) Ein Finetuning anhand von bestimmten Beispielseiten kann für andere Seiten desselben Stapels nachteilig sein.

Diese Verluste können oft durch Sichtung und Vorsortierung des Materials, durch eine Vorverarbeitung der Bilder (Randentfernung oder einheitliche Randdarstellung, einheitliche Satzspiegel, Teilung von Margen in rechte/linke Buchseiten usw.) begrenzt werden.

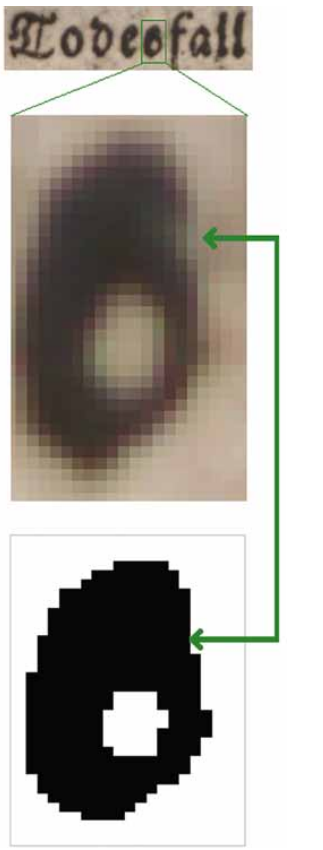
[27] Detaillierte Flussdiagramme gibt es in der Literatur zu konkreten OCR-Untersuchungen: einige Abläufe sind vergleichend dargestellt in KÄMMERER 2009. Nicht dargestellt sind interne Rückkopplungsmechanismen, wie sie ansatzweise auch in den getesteten Programmen integriert sind (je nach Ergebnis adaptiv wiederholte Anwendung von Segmentierungsalgorithmen, Zeichenzuweisungen oder Wortkorrekturschritten).

[28] Linguistische Nachkorrektur wird von beiden Programmen nicht angeboten und daher aus der Betrachtung ausgeschlossen.



38 Typischer Workflow

3.1.2 Bildvorverarbeitung und Binarisierung



39 Informationsverlust durch Binarisierung

Da die zur Zeichenerkennung gespeicherten Muster-Repräsentationen sich auf Schwarzweiß-Zeichen beziehen, muss die Vorlage in ein bitonales Bild gewandelt werden. Vor und/oder nach diesem Schritt werden i. d. R. verschiedene Filterungen angewandt wie Helligkeits- und Kontrastnormalisierungen, Ausgleich von Drehwinkeln, Wölbungen und Verzerrungen („Skew correction"), Ausfilterung von Verunreinigungen, die anhand ihrer (kleinen) Größe oder ihres Abstands zu offensichtlichen Zeichen- und Textregionen erkennbar sind (typisch für altes Papier wie für Mikrofilme), Abschneiden von erkennbaren Rändern und Nicht-Text-Bereichen usw.

Stufeninput und -output

Eingang: Farb- bzw. Graustufenbild
\+ *Steuerdaten:*
potentiell: Bildvorverarbeitungs- und Binarisierungsparameter

Ausgang: Bitonales, unter Umständen beschnittenes oder gefiltertes Bild

Typische Informationsverluste und Fehlerquellen

Unvermeidlich:

a) Die Vergröberung der Vorlage durch Reduktion der Farb- oder Graustufeninformation auf ein bit je Pixel führt dazu, dass relative Helligkeitsunterschiede innerhalb des hellen oder dunklen Teils des Zeichenbilds der Erkennung nicht mehr zur Verfügung stehen. Wegen der schon bei wenigen Graustufen exponentiell ansteigenden Anforderungen an Musterrepräsentation, Speicherplatz, Rechenzeit und Muster-Trainingszeit wird dieser erste entscheidende Informationsverlust der OCR noch auf längere Sicht nicht überbrückbar sein.

Potentiell beeinflussbar:

b) Scan-bedingte Verluste entstehen durch:
- ungünstige Farbintensität und Kontrast;
- Verzerrungen (Wölbung, Knicke);
- Verunreinigungen;
- Ränder und graphische Elemente; usw.

Als wichtigste Abhilfemöglichkeiten sind zu nennen:
- eine Normalisierung im Voraus oder durch die OCR-Software
- eine Gruppierung nur der einander vergleichbaren Bilder für eine gemeinsame Verarbeitung (s. o.);
- eine Entzerrung im Voraus oder durch die OCR-Software;
- eine softwareseitige Layouterkennung;
- ein manuelles Schneiden der Bilder;
- softwareseitige Filter;
- manuelle Filter.

c) Die Unterschiede zwischen „gleich hellen“ unterschiedlichen Farben gehen verloren. Abhilfe kann durch eine passende Farbfilterung geschaffen werden.
d) Die Anwender verzichten auf ein Finetuning wegen fehlender oder zu unkomfortabler und zeitaufwendiger Konfigurierbarkeit der Software.
Abhilfe ist allenfalls möglich durch:
– den Einsatz von gut eingearbeitetem Personal (anwenderseitig),
– den Einkauf des Finetunings als Dienstleistung (organisatorisch),
– die Optimierung der Software auf leichte und verständliche Konfigurations-Interfaces mit übersichtlicher Vorschau der Ergebnisse jeder Parameteränderung (entwicklerseitig).

Wann unschädlich: Unter günstigen Umständen kann dieser Informationsverlust unbedeutend werden (und im wesentlichen darauf sind die in bestimmten Office-Umgebungen heute gelegentlich erreichbaren hundertprozentigen Erkennungsleistungen zurückzuführen): nämlich, wenn z. B. durch hohen und gleichmäßigen Kontrast, durch unbeschädigtes Druckbild (keine „brüchigen“ Lettern, keine bedeutungsunterscheidenden Serifen ...) usw. das relevante Drucktypenbild entweder bereits ohnehin in einer nahezu bitonalen Ausprägung vorliegt oder Farb- und Helligkeitsstufen von vornherein nur redundante Information tragen, und wenn ein geschlossener Typenvorrat eine trennscharfe Abbildung auf die Mustermenge erlaubt. In Alten Drucken ist dieser Idealzustand nur in Ausnahmefällen gegeben.

Fast immer unschädlich bzw. beherrschbar ist der bloße Verlust der Farbinformation gegenüber Graustufen; speziell in Alten Drucken kommen zwar Wechsel zwischen rotem und schwarzem Textdruck vor, aber kaum Situationen, in denen allein die Farbqualität zwischen Vorder- und Hintergrund unterscheidet, die Farbintensität aber nicht.

3.1.3 Segmentierung (Layoutanalyse, Blockerkennung, Segmentierung auf Wort- und Zeichenebene)

Stufeninput und -output

Eingang: Binärbild
\+ *Steuerdaten:*
potentiell: Segmentierungsparameter wie Abstandsmaße, Mindest- und Höchstgrößen von Zeichen usw.

Ausgang: Satz einzelner Binärbilder bzw. Koordinaten der Bildregionen

Typische Informationsverluste und Fehlerquellen

Unvermeidlich:
a) Wo typographische Merkmale (Zwischenraum zwischen Spalten/Absätzen/Wörtern/Zeichen, konsistente Linienführung, Zeilenausrichtung usw.) fehlen, können diese Segmente maschinell nicht richtig erkannt werden.

Potentiell beeinflussbar:
b) Für menschliche Gestaltwahrnehmung intuitiv bedeutungstragende Layoutmerkmale können nicht in für alle Materialien gleichermaßen gültige Operationalisierungen umgesetzt werden.

c) Lexikalische und semantische Indikatoren bleiben für die Strukturierung ungenutzt.[29]
d) Etwaige in Katalogen oder Datenbanken bereits erfasste Strukturinformationen bleiben ebenfalls ungenutzt.

Weil aus verlorener Strukturinformation regelmäßig eine in der Vorlage nicht gegebene Mehrdeutigkeit der Zeicheninterpretation resultieren kann, bildet diese Stufe den zweiten entscheidenden Informationsverlust der maschinellen OCR. *Abhilfe* wird hier vor allem entwicklerseitig durch eine konsequente Unterstützung der Hinzuziehung solcher Strukturinformationen möglich sein. Bis dahin kann eine maschinelle Layoutanalyse stets nur Teile der Anforderungen erfüllen.

Wann unschädlich: Am wenigsten problematisch ist dieser Verlust in Fällen sehr einfacher Layouts und strukturell möglichst gleichwertiger Zeilen.

Manche Strukturinformation wird sich nach der OCR noch aus Merkmalen des gelesenen Texts in Nachkorrekturschritten rekonstruieren lassen. Für eine schon bei der Zeichenerkennung als Musterrestriktion wirksame Vorab-Segmentierung bleibt sie aber auch dann ungenutzt.

3.1.4 OCR (inklusive unmittelbar verbundener Fehlerkorrekturmechanismen)

Für die Klassifikation von in bitonaler Form vorliegenden Zeichenbildern gibt es sehr unterschiedliche leistungsfähige Verfahren und einen entwickelten Forschungsstand, so dass bei geeignetem Input an Bildern und Musterklassen potentiell eine nahezu verlustfreie Zuordnung erreichbar ist. Dabei ist es gleich, ob konventionelle Ähnlichkeitsmaße berechnet oder neuronale Netze trainiert, ob eine topologische Abstraktion des Zeichens oder nur aus dem Bild abgeleitete Kennzahlen für die Klassifikation herangezogen werden.[30]

Die Abgrenzung des OCR-Schritts von der vorangehenden Segmentierung einerseits und der nachfolgenden Anwendung erster Zeichenersetzungsregeln andererseits ist unscharf. Ursache dafür ist die Klassifikation, die in guter OCR-Software zum Teil adaptiv auf die Segmentierung zurückwirkt (wenn etwa ein möglicherweise teilbares Zeichensegment mehrfach gelesen wird und je nach Möglichkeit in zwei Segmente geteilt wird oder nicht, oder wenn bestimmte Teilmuster sofort zu einem kombinierten Zeichen zusammengesetzt werden).

Stufeninput und -output

Eingang:
Bildregionen auf Textblock-, Zeilen- und Einzelzeichenebene
\+ *Steuerdaten:*
Mustersammlung
potentiell: Wörterbuch
potentiell: unmittelbar auf Zeichen(-teil)-Ebene ansetzende Ersetzungsregeln

Ausgang:
OCR-Text
Koordinaten der gelesenen Zeichen oder „Wörter“
\+ *Evaluationsdaten:*
potentiell: statistische Auswertungen wie Anteil erkannter Buchstaben und Wörter, Konfidenzwerte zur vermuteten Erkennungssicherheit auf Zeichen- und Wortebene usw.;
potentiell: beliebige computerlinguistische Auswertungen

[29] Zu Möglichkeiten der regelbasierten Dokumentstrukturanalyse *nach* der OCR vgl. Arbeiten an der Universitäts- und Landesbibliothek Tirol im Rahmen des IMPACT-Projekts, s. MÜHLBERGER 2011; der Nutzen der Strukturinformation wird aber auch dort überwiegend für die Nach-OCR-Phase gesehen.
[30] Zu Klassifikationsverfahren s. u. a. SCHULZ 2006

Typische Informationsverluste und Fehlerquellen

Unvermeidlich:

a) Es kommt zu falscher Musterzuordnung von beschädigten bzw. verunreinigten Drucktypen, wenn eine hohe Ähnlichkeit zu anderen Zeichen besteht.

b) Einige Zeichenwerte auf Muster überlappen sich. Das ist der Fall, wenn im gleichen Musterbestand eine Schrift für ein Zeichen X Druckbilder zeigt, die in einer anderen Schrift desselben Musterbestands für ein Zeichen Y stehen. Fehlerhaft ist hier bereits der Zustand der Musterdatei, besonders wenn die Software eine derartige Vermengung schon beim Anlegen der Musterdatei erzwingt und keine separaten Musterbestände ermöglicht, die später fallweise kombiniert werden können.

c) Bei Klassifikation eines Zeichens – auch wenn sie durch Vermerk von Konfidenzwerten gewissermaßen „unter Vorbehalt“ geschieht – geht die Information über die Alternativkandidaten und ihre Rangfolge verloren. Dies muss als der dritte wesentliche Informationsverlust heutiger maschineller OCR beachtet werden: er beeinträchtigt von vornherein die Treffsicherheit jeder nachfolgenden lexikalischen Korrektur, die nun unter allen *generell möglichen* Verwechslungen (im besten Fall sprachbezogen gewichtet) auswählen muss und sich nicht mehr gezielt auf die wenigen am Ort *für das konkrete optische Muster* allein möglichen Verwechslungen beschränken kann.

Potentiell beeinflussbar:

d) Zeichen, die nicht zur aktuell geladenen Musterbibliothek passen, werden nicht erkannt. Eine universelle Mustersammlung (wie z. B. die von FineReader mitgelieferte Fraktur-Bibliothek) bietet zwar einen optimalen Durchschnitt, verschenkt aber potentiell mögliche Erkennungsquoten auf Drucken, in denen eben nicht die ganze Breite der Muster vorkommt und daher zu viele irrelevante optische Muster angewendet werden. Abhilfe ist abhängig davon, ob die Software ein Vorhalten separater „reiner“ Musterbestände zum Zweck fallweiser Kombination überhaupt zulässt.

e) Zeichen, die in Alphabet- und Sprachdefinitionen nicht vorhergesehen wurden, werden nicht erkannt.

Wann unschädlich:

Echte OCR-Fehler sind im Prinzip nur dann unschädlich, wenn sie durch lexikalische oder linguistische Nachkorrektur rückgängig gemacht werden können. Hierzu sind sehr genaue Annahmen über Regelhaftigkeiten des Ausgangstexts nötig, was zumindest in formalisierten Textsorten oder Texten mit inhaltlich „geschlossenen Welten“ oft gegeben sein kann. Wenn im industriellen Umfeld eine praktisch fehlerfreie OCR durchaus anzutreffen ist, dann i. d. R. unter solchen Umständen. Andererseits ist eine Vermeidung von OCR-Fehlern um so aussichtsreicher, je überschaubarer und trennschärfer der verwendete Drucktypenvorrat ist und je vollständiger und exklusiver er durch Training in die Musterbibliothek eingearbeitet werden konnte.

– Zwischenanmerkung –
Will man eher auf Fehlervermeidung als auf nachträgliche Fehlerkorrektur setzen, muss bis zu dieser Stufe das Maximum an Lesetreue erreicht werden.

3.1.5 Lexikalische und linguistische Nachkorrektur

Jede auf die OCR folgende Fehlerkorrektur versucht, einen bereits eingetretenen Wissensverlust nachträglich zu kompensieren. Das ist naturgemäß nur unter Heranziehung von Regularitätshypothesen über die Vorlage möglich, die oft heuristisch bleiben müssen und im Einzelfall versagen können. Man wird auf diese Weise auch in Fällen „richtiger" Korrektur i. d. R. nicht wissen können, ob die Korrektur richtig war. Hinzu kommt die Anforderung, dass reale Druckvorlagen den angenommenen Regularitäten ihrer Textsorten nicht immer folgen (im einfachsten Fall: Druckfehler oder Schreibvarianten enthalten) und auch dann von der OCR treu wiedergegeben werden sollen.[31]

Diesen Nachteilen nachträglicher lexikalischer und „linguistischer" (morphologischer, syntaktischer, kontextbezogener usw.) Fehlerkorrektur steht der große Vorteil gegenüber, dass ab dieser Stufe auf Textformaten gearbeitet werden kann, d. h. mit der Computerlinguistik die Ressourcen einer ganzen Wissenschaft und eine große Vielfalt an Werkzeugen zur Verfügung stehen.[32]

Es gibt ab dieser Stufe daher zusätzliche Alternativen:

- Zwischen in OCR-Software integrierten Korrekturverarbeitungsschritten und einer externen Bearbeitung durch andere Tools kann gewählt werden.

- Ein einmal vorliegendes Roh-OCR-Ergebnis kann (nacheinander oder parallel-verzweigend) mehrfachen Korrekturschritten zugeführt werden.

Erschwerend kommt bei alten Schriften hinzu, dass die zur Nachkorrektur verwendeten Wörterbücher die historische und uneinheitliche Rechtschreibung berücksichtigen müssen; eine gewisse Aufblähung mit Schreibvarianten ist daher nicht zu vermeiden.

Auf echte linguistische Korrekturmöglichkeiten wird hier nicht eingegangen, weil beide untersuchten Softwareprodukte diese nicht anbieten.[33] Der Übergang ist aber fließend, da auch in die lexikalische Korrektur linguistische Komponenten eingehen können, z. B. wenn das Lexikon mit morphologischen Ableitungen, abgeleiteten historischen Schreibvarianten oder Kollokationen angereichert werden konnte.

Gesichtspunkte zur Beurteilung der lexikalischen Korrektur können sein:

- ob sie vollautomatisch oder (teil-)manuell erfolgt;
- ob erfasste Wortkoordinaten angepasst werden oder nicht;
- welche sprachbezogene Zusatzinformation verwendet wird (Worthäufigkeiten? Konfusionswahrscheinlichkeiten? Zeichenübergangswahrscheinlichkeiten? usw.);
- ob diese Zusatzinformation implementiert ist oder vom Nutzer fallbezogen parametriert werden kann; usw.[34]

[31] Hinweis u. a. von Marco Büchler, der als „big challenge" der OCR nennt: „How can we correct OCR without correcting spelling errors?" BÜCHLER 2011

[32] Detaillierte sprach- und dokumentenspezifische Fehlerprofile wurden - mit dem Ziel der Entwicklung gut bedienbarer Korrekturwerkzeuge - u. a. im IMPACT-Projekt untersucht, vgl. REFFLE 2011. - Ein am CIS München vorangehendes DFG-Projekt „Domänen- und dokumentenadaptive Verfahren zur Nachkorrektur von OCR-Ergebnissen" versuchte die Leistungsfähigkeit von Korrekturverfahren dadurch zu erhöhen, dass explizit „Domäne, sprachliches Bild des Gesamttexts und logischer Kontext von Dokumenten*teilen* bei der Korrektur mitberücksichtigt werden" (Zu Projektbeschreibung mit Publikationsangaben s. NACHKORREKTUR VON OCR-ERGEBNISSEN). Es ist davon auszugehen, dass das im Anschluss an IMPACT gegründete Kompetenzzentrum einen umfassenden Informationszugang zu solchen Möglichkeiten bereitstellt.

[33] Diese Thematik wird u. a. behandelt in HAUSER 2007, STROHMAIER 2004.

[34] Vgl. SCHULZ 2006, RINGELSTETTER 2003 u. a.

Stufeninput und -output

Eingang:
OCR-Textergebnis
\+ *Steuerdaten:*
Wortliste
potentiell: Gewichte für bestimmte Wörter oder Lexikon-Teilbestände (Vorrang bestimmter Kandidaten vor anderen)
Parameter für erlaubte Distanz, bis zu der ersetzt werden darf
potentiell: Gewichte für bestimmte Zeichenersetzungsschritte[35]

Ausgang:
Textexport
\+ *Evaluationsdaten:*
potentiell: Vermerke der vorgenommenen Ersetzungen (Original-Lesart, Kandidaten) im Textexport
potentiell: beliebige computerlinguistische Auswertungen

Typische Informationsverluste und Fehlerquellen
Eine mögliche Fehlerklassifikation[36]

Da sich die lexikalische Nachkorrektur weitgehend materialunabhängig und in computerlinguistisch zugänglichen Textformaten abspielt, gibt es eine reichhaltige, theoretisch gut durchgearbeitete Literatur zu Einflussfaktoren, Fehlersituationen und Korrekturstrategien unter Verwendung verschieden aufbereiteter Korpora, was hier nicht wissenschaftlich referiert werden kann.

Exemplarisch soll hier eine Fehlerklassifikation von Christian M. Strohmaier kurz vorgestellt werden, die zu praktischen Vorüberlegungen bei der Anlage der verwendeten Wörterbücher beitragen kann, an die teils gegensätzliche Anforderungen bestehen. Denn einerseits sollen alle im Original tatsächlich vorkommenden Wörter im Lexikon auch enthalten sein (d. h. akzeptiert und nicht fälschlich korrigiert werden), andererseits soll das Wörterbuch nicht unnötig viele Wörter enthalten, was zur unnötig vermehrten Akzeptanz auch von OCR-Fehlern führt, die eigentlich korrigiert werden müssten.[37]

[35] Die Korrektur von als „häufig“ bekannten Verwechslungen „verbraucht“ dann weniger der zulässigen Distanz, bis zu der die Korrektur stattfinden darf.
[36] STROHMAIER 2004
[37] Im Idealfall würde auch erkannt, was für Wörter in einem Textkontext überhaupt vorkommen dürfen, und es würden auch nur solche als Korrekturkandidaten erscheinen.

Strohmaier gliedert die Korrekturfehler in sieben Fehlerklassen auf:

„1. **Falscher Freund (false friend)**. Falsche Freunde sind eine nur schwer aufzuspürende Fehlerklasse. Die Nachkorrektur hat keinen Anlass, an dem vorgefundenen, falschen OCR-Token zu zweifeln, da es lexikalisch ist. Entdeckungsstrategien sind die Verwendung mehrerer OCR-Engines, die Berücksichtigung von Konfidenzwerten der Wort- bzw. Charakter-Erkennung der OCR-Engine selbst und stärkere Gewichtung von Kontextinformationen. Eine Vermeidungsstrategie ist die Verkleinerung des Lexikons. Es gilt, je kleiner das Lexikon D ist, desto geringer ist die zu erwartende Anzahl falscher Freunde.

2. **Zu vorsichtig (too cautious)**. Das korrekte Wort ist Spitzenkandidat der Vorschlagsliste. Allerdings wird die Korrektur nicht ausgeführt, da der Konfidenzwert zu gering ist $con\ f(w^{ocr}) < 1$. Vermeidungsstrategie ist eine mutigere Nachkorrektur, die alle Konfidenzwerte erhöht. Allerdings verhält sich die Fehlerklasse der unglücklichen Korrekturen (5) antagonistisch zu dieser Klasse. Die Einstellung des Konfidenzwertes *con f* muss zwischen diesen beiden Klassen austariert werden.

3. **Falscher Kandidat und falsche Korrekturgrenze (wrong candidate and threshold)**. Das richtige Wort ist im Lexikon enthalten. Allerdings ist es nicht auf die Position des Spitzenkandidaten der Liste gerankt. Ausserdem sind die Konfidenzwerte aller Kandidaten des OCR-Tokens zu niedrig, so dass keine Korrektur vorgenommen wird. Fehlt das richtige Wort ganz in der Kandidatenliste, wurde es entweder zu stark von der OCR-Engine verunstaltet oder die Grobfilterung für die Kandidatenliste ist mangelhaft. Ansonsten ist die Vermeidungsstrategie eine verbesserte Konfidenzwertberechnung, d. h. geeignetere und/oder mehr Faktoren zur Kandidatenbewertung sowie eine bessere Kombination dieser Faktoren.

4. **Falscher Kandidat (wrong candidate)**. Die Nachkorrektur erkennt richtig, dass ein Korrekturvorschlag ausgeführt werden soll und auch das richtige Wort ist in Lexikon enthalten, jedoch nicht an erster Stelle der Kandidatenliste. Auch in diesem Fall sollte die Grobfilterung überprüft werden, wenn das richtige Wort in der Kandidatenliste fehlt. Ansonsten ist die Vermeidungsstrategie wie für die vorherige Fehlerklasse (3) eine verbesserte Konfidenzwertberechnung.

5. **Unglückliche Korrektur (infelicitous correction)**. Korrekturvorschläge, die ausgeführt werden, obwohl das Groundtruth-Token nicht im Lexikon war, führen immer zu einem Fehler ($w^{orig} \notin D$). Der drastischere Fall tritt ein, wenn die OCR-Engine das Originaltoken bereits richtig gelesen hat $w^{ocr} = w^{orig}$. Dadurch verschlechtert die Nachkorrektur das Endergebnis. Vermeidungsstrategien sind die Vergrößerung des Korrekturlexikons und eine vorsichtigere Berechnung des Konfidenzwertes. Allerdings steigen mit einer Lexikonvergrößerung die potentiellen, false friends (1) und mit einer vorsichtigeren Konfidenzwertberechnung die potentiellen Fehler der Klasse too cautious (2).

6. **Unvermeidbar I (no chance I)**. Die Nachkorrektur wird [wegen zu niedrigem Konfidenzwert des Ersetzungskandidaten] nicht aktiv, obwohl eine Fehlerkennung vorliegt. Da aber das richtige Wort nicht im Lexikon und daher auch nicht in der Kandidatenliste ist, ist die Nachkorrektur ohnehin chancenlos. Die einzige Vermeidungsstrategie ist eine Vergrößerung des Korrekturlexikons *D*.

7. **Unvermeidbar II (no chance II)**. Die Nachkorrektur erkennt richtig, dass ein Korrekturvorschlag ausgeführt werden soll und wird [aufgrund des ausreichenden Konfidenzwerts des Ersetzungskandidaten] aktiv. Es besteht jedoch keine Chance richtig zu korrigieren, da das originale Wort nicht im Lexikon enthalten ist $w^{orig} \notin D$. Auch hier ist die einzige Vermeidungsstrategie eine Vergrößerung des Korrekturlexikons *D*.“

und er schließt:

„Je größer das Korrekturlexikon desto geringer die Fehleranzahl des gesamten rechten Astes (Klasse (5), (6) und (7)), aber desto größer ist auch die Anzahl falscher Freunde, Klasse (1). Dies ist allerdings kein streng funktionaler Zusammenhang, sondern eine statistische Gesetzmäßigkeit. Die Fehlerverteilung innerhalb der restlichen Klasse (2), (3) und (4) bleibt im statistischen Sinne konstant.“[38]

[38] STROHMAIER 2004, S. 96f.

Diese Fehler einbeziehend, wäre folgende Gruppierung möglich:

Unvermeidlich:

Dass jede der Fehlerklassen 1-7 in automatischer Korrektur vorkommt, dürfte nicht zu verhindern sein, zumal dieselben Fehler auch in einer manuellen Korrektur mit unvollständiger Sachkenntnis auftreten.
Hinzu kommt, dass zwar die Lagekoordinaten der gelesenen Zeichen(ketten) anhand der Bildvorlage ermittelt wurden, für korrigierte Formen aber nur indirekt, z. B. durch Interpolation, auf Koordinaten geschlossen werden kann. Es wird hier daher geringe Abweichungen geben.

Potentiell beeinflussbar:

Die Wahrscheinlichkeit des Auftretens aller dieser Fehler ist weitgehend steuerbar, wenn auch zum Teil auf gegenseitige Kosten.
Aus dem Gesagten ergibt sich als Abhilfe:

- die Wahl eines Wörterbuchs optimaler Größe;
- gut eingestellte Gewichtungen der Ersetzungskandidaten auf Zeichen- und Wortebene.

Außerdem sollte beachtet werden:

- die Wahl geeigneter Distanzmaße[39] zwischen dem als falsch erkannten OCR-Wort und dem Ersetzungskandidaten, anhand derer eine Korrektur vorgenommen oder unterlassen wird. Sprach- und schrifttypisch gewichtete Maße sind hier eindeutig vorzuziehen, da nur sie es ermöglichen, eine im tatsächlichen Schriftbild wahrscheinliche Verwechslung leicht zu machen (z. B. „ni“ > „m“), ohne zugleich auch alle möglichen anderen Ersetzungen mit rechnerisch vergleichbar „niedriger“ Distanz anzustoßen. Die Gewichtungen sollten asymmetrisch (für jede Ersetzungsrichtung separat) zugeteilt werden können.
- dass verschiedene Teile der eingelesenen Vorlage unterschiedliche Wortbestände erwarten lassen, d. h. man Situationen einschränken sollte, in denen Textteile nicht gut zum aktuell eingestellten Wörterbuch passen.
- dass das Lexikon in zwei Rollen auftritt, die möglicherweise unterschiedlich von Unvollständigkeit oder Aufblähung des Wörterbuchs betroffen sein können: (a) als Akzeptor einer OCR-Lesart, (b) als Repertoire für Ersetzungskandidaten.

Wann unschädlich:

Per definitionem können Text-Fehler der maschinellen Nachkorrektur nie unschädlich sein, außer im seltenen Fall anschließender manueller Korrektur mit hoher Aufmerksamkeit und perfekter Kenntnis des Textgegenstands.
Dagegen geht bei jedem nachgeschalteten weiteren automatischen Korrekturschritt die Information über die Nähe zur optischen Vorlage verloren und hat keine Restriktionswirkung mehr; die in der Nachkorrektur erst entstandenen Fehler können daher in weiteren Korrekturschritten besonders weit vom Original wegführen.
Wenig problematisch ist dagegen der Informationsverlust bezüglich der genauen geometrischen Zeichen- und Wortkoordinaten, solange diese Information nicht zum akkuraten Ausschneiden der Bildteile benötigt wird, sondern z. B. zur Hervorhebung einer Fundstelle auf dem Bild dient. Ob hier z. B. bei einer in der Nachkorrektur erfolgten Zerlegung eines falsch zusammen gelesenen Worts in zwei Teile die interpolierte Lücke auf dem Bildschirm um wenige Pixel neben der Lücke im Original angezeigt wird, dürfte an der Brauchbarkeit der Ortsinformation für die meisten Anwendungsfälle wenig ändern.

[39] Einige Distanzmaße werden vorgestellt in der generell als technische Einführung in OCR lesenswerten Arbeit SCHULZ 2006. Üblich sind zum Beispiel die Levenshtein-Distanz, die die Anzahl notwendiger Zeichenersetzungen zählt bzw. gewichtet aufsummiert.

3.1.6 Semantische Erschließung: Strukturdaten und Named Entities

Auf die im OCR-Zusammenhang besonders interessante semantische Auswertung nicht-textueller Merkmale wurde in Abschnitt 3.1.3 bereits eingegangen.

Für die semantische Interpretation von bereits auf Textebene vorliegendem Input (Post-OCR) hat der Anwender prinzipiell die Wahl, auch außerhalb des jeweiligen OCR-Programms nach geeigneten Werkzeugen und Verfahren zu suchen. Im Rahmen des beschriebenen Projekts konnten semantische Auszeichnungen innerhalb der Softwareprodukte praktisch nicht realisiert werden; möglich waren einige automatische statistische Auswertungen durch Anwendung semantischer Entitäten auf die OCR-Exporte. Möglicherweise ist die Erwartung semantischer Auszeichnung aber generell erst dann berechtigt an OCR-Software zu stellen, wenn aus dem Bibliotheksbereich selbst abgesicherte und operationalisierbare Vorgaben zur semantischen Interpretation konkreter Layout- oder textueller Merkmale angegeben werden. Softwarehäuser müssten sich hierzu sehr tief und fundiert mit buchkundlichen Fragen auseinandersetzen und scheinen sich aus Aufwands- und vielleicht auch Gewährleistungsgründen zu scheuen, selbst weitgehende Annahmen darüber in operationalisierte Definitionen zu gießen, die absehbar immer wieder in Einzelfällen oder im nächsten Bilderstapel versagen müssen. Hier „hartcodierte“ Programmierungsleistungen zu bestellen, scheint daher vorläufig keine sinnvolle Option zu sein. Vielleicht ist eine arbeitsteilige Lösung denkbar, indem Softwarehäuser vorkonfigurierte Schnittstellen für typische Strukturelemente und Named Entities[40] programmieren, deren konkrete Erkennungsmerkmale vom Nutzer verfeinert werden müssen.[41] Wenn das gelänge, könnten Anwender, die ihr Material gut kennen, Information gewissermaßen wieder „in die Texte hineintragen“.

3.1.7 Einfluss der Verarbeitung ganzer Seiten

Die passgenaue Anwendung der richtigen Spracheinstellungen und Mustersammlungen wird umso anspruchsvoller, je kleinteiliger die Regionen werden, die eine verlustfreie Gleichbehandlung noch vertragen. In Alten Drucken stehen lateinische, griechische oder hebräische Wörter in entsprechender Schrift inmitten von Fraktur-Kontext. Im deutschsprachigen Satz zieht sich das bis in das einzelne Wort hinein, wenn lateinische Wortstämme in Antiqua, deren der deutschen Grammatik entstammenden Flexionssuffixe aber – wie damals häufig üblich – in Fraktur gesetzt sind. Griechische und hebräische Buchstaben dienen gelegentlich als bloße Aufzählungszeichen, stehen damit einzeln und sind der jeweiligen Sprache oft nur durch semantisches Verständnis ihrer Funktion bzw. nur durch Hinzuziehung distalen Kontexts ihrer Schriftfamilie zuzuordnen. Frakturtexte, die keine Zahlen enthalten, könnten aussichtsreicher mit anderen („reineren“) Mustersammlungen gelesen werden als Texte, in denen arabische wie römische Ziffern an unvorhersehbaren Orten stehen können; umgekehrt könnte oft eine textsorten- und layoutgesteuerte Erwartung römischer Zahlen deren richtige Erkennung überhaupt erst ermöglichen usw.

Der vierte wesentliche Informationsverlust der maschinellen OCR ergibt sich daher nicht aus Problemen in einem konkreten der oben genannten OCR-Teilschritte, sondern aus der bei Massen-OCR kaum vermeidbaren Anwendung pauschalisierender Einstellungen, wenn sie im Verarbeitungsstapel weder von Seite zu Seite noch von Textblock zu Textblock oder gar innerhalb einer Zeile bzw. eines Worts flexibel genug gewechselt werden können, um die Potentiale vorhandener Muster- und Sprachinformation tatsächlich wirksam werden zu lassen.

[40] Zumindest für *named entities* können aus der Computerlinguistik formale Kriterien bezogen werden; eine als Anregung lesbare Arbeit ist z. B. GEIERHOS 2007, wo über den Titel „Grammatik der Menschenbezeichner in biographischen Kontexten“ hinaus auch auf grammatische Merkmale weiterer Entitäten Bezug genommen wird.

[41] Beispielsweise als „Kolumnentitel“ gelte wahlweise „die erste Zeile“ / „der Text oberhalb der ersten durchgezogenen Linie“ / „die erste Zeile, wenn ihr Abstand zur zweiten größer als NN mm ist“; oder: Als „Bibelstellenangabe“ gelte „jede Zeichenkette, die in Antiqua gesetzt ist, zum Regulären Ausdruck X passt und sich innerhalb eines Strukturelements Y befindet“ usw.

Batchinput und –output

Eingang:
Menge an Images
Musterbibliothek(en)
Wortliste(n)
\+ *Steuerdaten:*
Verarbeitungsparameter, Angaben zu Schriftfamilien, Alphabeten und Sprachen, Zielformate, Zielspeicherorte

Ausgang:
Menge an Exportdateien, ggf. in verschiedenen Formaten
\+ *Evaluationsdaten:*
Protokolle
beliebige statistische und computerlinguistische Auswertungen

Typische Informationsverluste und Fehlerquellen

Unvermeidlich:
Da diese Fehlerquellen allein durch die Verarbeitungsorganisation entstehen, sind sie nicht aus sich heraus unvermeidlich, sondern „nur“ in dem Maße, wie der nötige Aufwand zur Zusammenstellung homogener Stapel nicht geleistet werden kann.

Potentiell beeinflussbar:
Verluste auf den jeweils schlechter zur aktuellen Konfiguration passenden Regionen/Seiten
Abhilfe ist z. B. zu erreichen durch:
- die Zerlegung von Batches in kleinere;
- die Teilung der Seiten vorab in separat zu verarbeitende Images bzw. Image-Regionen;
- eine Mehrfachverarbeitung unter verschiedenen Bedingungen und den anschließenden Versuch, pro Textblock die im Nachhinein günstigste Ausgabe zu selektieren.

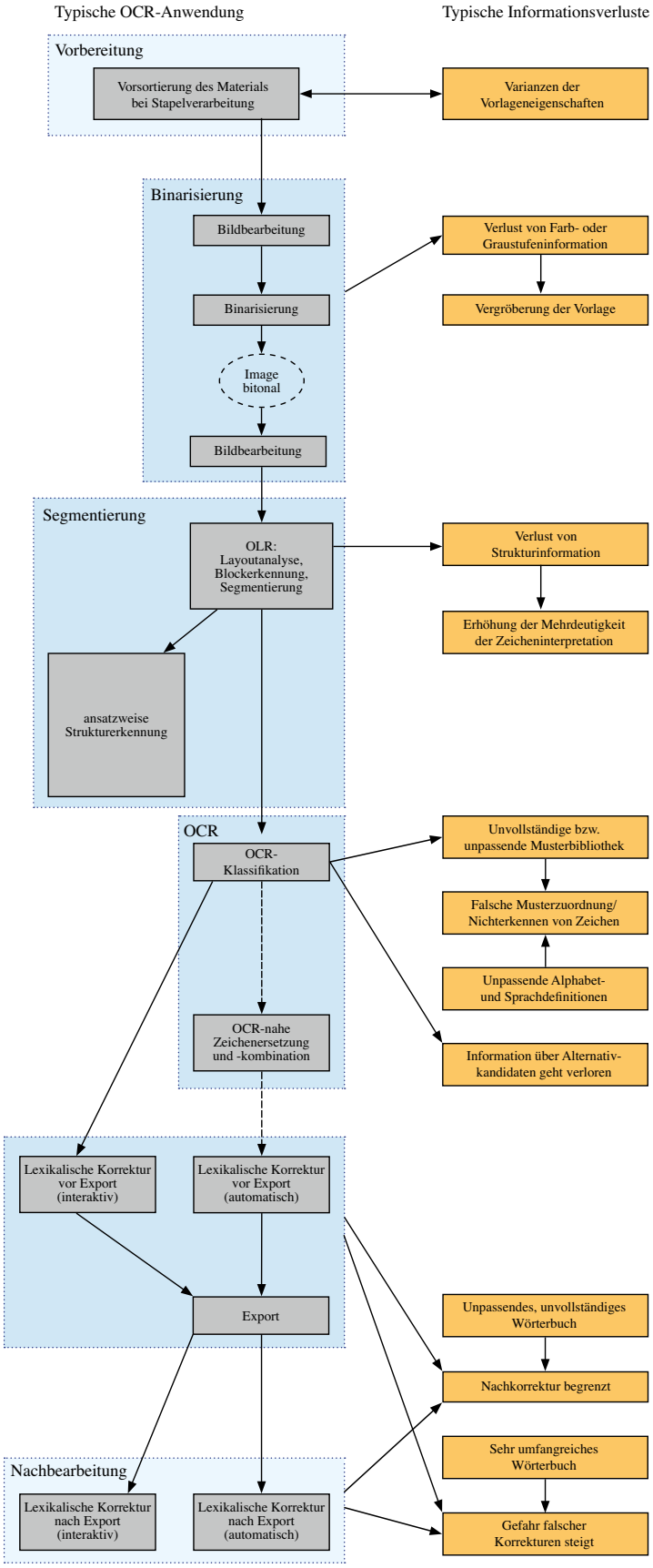
Diese Aufteilung kann idealerweise vor dem Batchlauf vorgenommen werden. Wenn das nicht praktikabel ist, könnten Seiten mit unnötig schlechter Erkennung auch nach dem Batchlauf einer Neubearbeitung mit anderen Parametern zugeführt werden. Software könnte dies unterstützen, wenn sie Evaluationskennwerte zur Ermittlung von zweckmäßiger Stapelaufteilung bereitstellt und Stapelprozesse mit wechselnden Parametern zulässt.

Wann unschädlich:
Im einfachsten Fall verursacht die Gesamtseiten- und Seitenstapel-Verarbeitung dann keine eigenen Fehler, wenn innerhalb der Seiten bzw. des Stapels gar keine inhomogenen Situationen auftreten, die eine Mischung separater Muster- oder Lexikonbestände erfordern.

Im Fall von inhomogenen Seiten oder Seitenstapeln gilt:
Bezogen auf die reine Zeichenerkennung können Misch-Musterbibliotheken dann unschädlich sein, wenn die verwendeten Drucktypen der beteiligten Schriftarten ausreichend disjunkt sind, d. h. sich immer nur da überlappen, wo dasselbe Zeichen gelesen werden soll.
Analog können gemischte Wörterbücher dann unschädlich sein, wenn aufgrund konservativ gewählter Distanzmaße und Konfidenzwerte keine Gefahr besteht, fälschlich „in die falsche Sprache bzw. den falschen Wortbestand hinein“ zu korrigieren.

3.2 Zusammenfassung

Wenn man im Vorgriff auf die im nächsten Kapitel folgende Beschreibung der Software auch die Hilfsobjekte (Parameter-, Muster- und Wörterbuchdateien) heranzieht, mit denen die einzelnen Schritte determiniert werden, dann könnten die typischen Fehlerquellen und die wichtigsten vom Anwender zu beachtenden qualitätsbestimmenden Faktoren wie folgt in Abb. 40 dargestellt werden:



40 Workflow und beeinflussende Faktoren sowie resultierende Fehlerquellen

4 Praktischer Softwaretest und -vergleich I: Die Programme
4.1 Übersicht über Parameter- und Hilfsdateien jedes Produkts

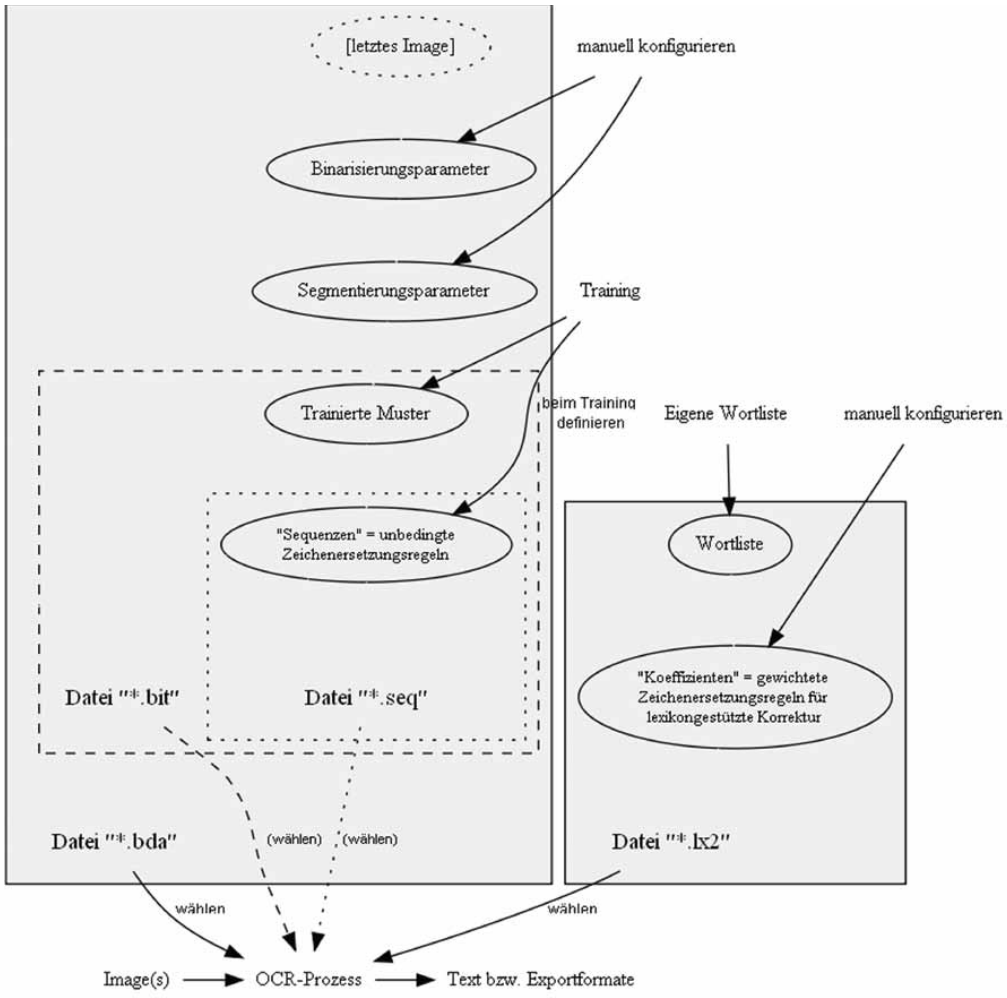
Der wesentliche Teil des Pilotprojekts bestand in der Untersuchung, wie ein Anwender mit Hilfe der jeweiligen Software zu den bestmöglichen Bedingungen eines OCR-Laufs gelangt. Dabei war der Frage nachzugehen, wie nachnutzbar diese sind, z. B. durch Fixierung erreichter Optimierungen in Parameter- bzw. Konfigurationsdateien bzw. in als Dateien speicherbaren Muster- und Wortbibliotheken. Entsprechend nimmt die Erstellung bzw. Beschaffung dieser Parameter- und Hilfsdateien einen großen Teil der (menschlichen) Bearbeitungszeit ein, ehe die eigentliche OCR-Produktion dann automatisch im Stapelbetrieb läuft und vorwiegend Rechenzeit beansprucht.

Zum besseren Verständnis der folgenden einzelschrittbezogenen Details folgt hier zunächst ein Überblick über die Speichermodelle der jeweiligen Software.

(a) BIT-Alpha[42]

Beteiligt sind pro OCR-Lauf zwei bis vier Parameterdateien, die in BIT-Alpha explizit als Datei geladen werden müssen und teilweise überlappende Funktionen haben:

- eine Konfigurationsdatei „.bda“ für Binarisierungs- und Segmentierungsparameter, optional auch für Musterbibliothek und Sequenzen;[43]
- soweit Muster nicht in „.bda“ bereitgestellt werden: ein separates Speicherformat „.bit“ für die Musterbibliothek, optional auch für Sequenzen (s. u.);
- soweit Sequenzen nicht in „.bda“ oder „.bit“ bereitgestellt werden: ein separates Speicherformat „.seq“ für Ersetzungsregeln (Sequenzen) zur sofortigen und lexikonunabhängigen Kombination bestimmter Zeichen(teile);
- eine Datei „.lx2“ für Wortbibliothek und Parameter der lexikalischen Nachkorrektur.



41 Parameter- und Hilfsdateien BIT-Alpha

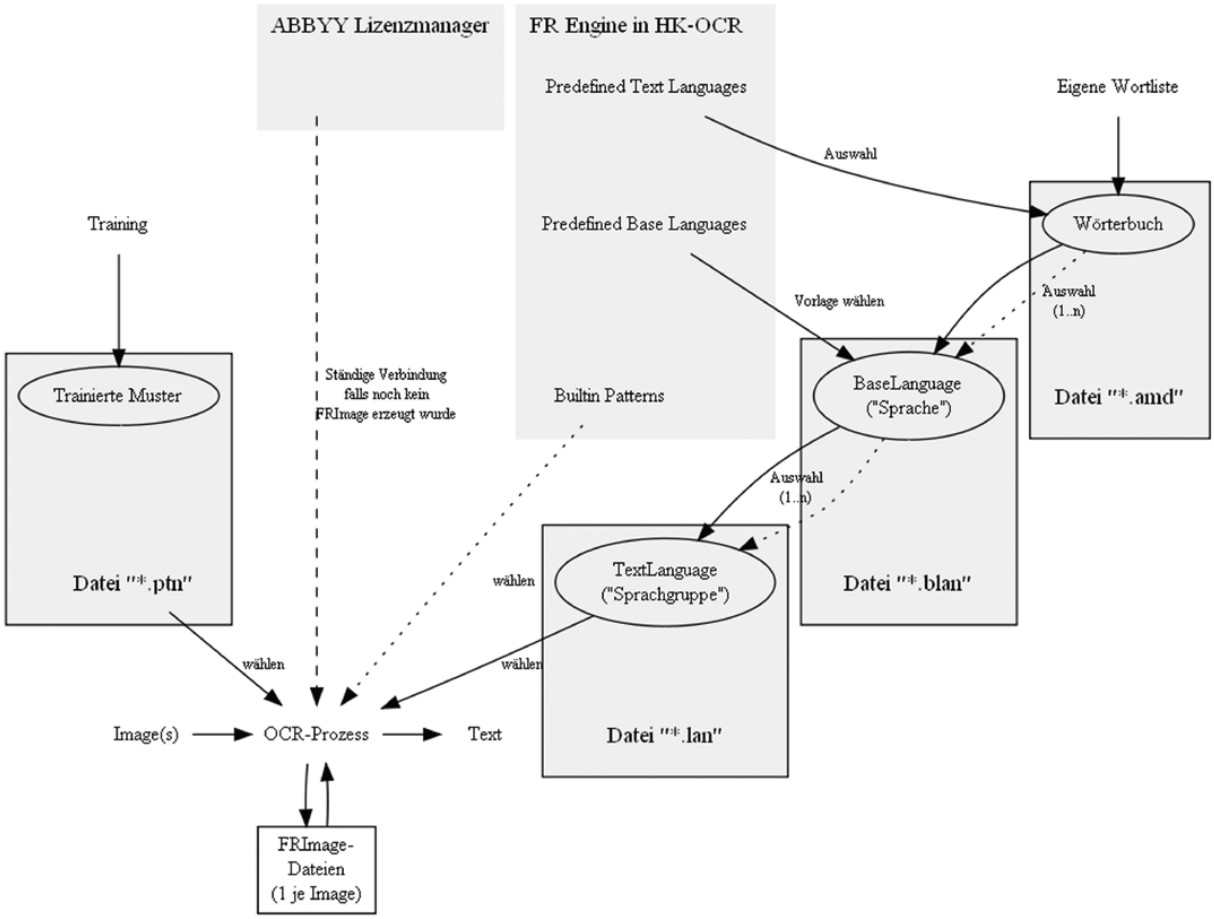
Für die Muster und Sequenzen kann während jeder BIT-Alpha-Sitzung neu entschieden werden, in welchem der alternativen Formate sie abgelegt werden; so würde auch eine Übertragung von Mustern zwischen bda- und bit- bzw. von Sequenzen zwischen bda-, bit- und seq-Dateien stattfinden.

(b) HK-OCR / FREngine 9

Abgesehen von impliziten Programmkonfigurationsdateien[44] nutzt HK-OCR pro OCR-Lauf vier eigentliche Parameterdateien:

- drei Dateitypen zur Wahl einer eigenen „Sprachengruppe“: eine Wörterbuchdatei „.amd“, eine *Base Language* Datei „.blan“ und eine *Text Language* Datei „.lan“; hier können Wortlisten importiert und Listen erlaubter Zeichen bearbeitet werden. Zu diesen Dateitypen muss nur das Verzeichnis eingegeben werden (Reiter „Weitere“ im Einstellungsbereich von HK-OCR), dann kann im Reiter „Einstellungen“ die Auswahl einer „Sprachengruppe“ erfolgen.
- eine Musterdatei „.ptn“, die explizit als Datei ausgewählt und geladen wird.

Hinzu kann je Einzelbild eine sogenannte (während einer früheren OCR angelegte) FR-Image-Datei kommen, die Vollbild- und Layout-Informationen enthält und die Einzelseitenlizenz verkörpert. Liegt diese FR-Image-Datei für das zu lesende Bild nicht vor, dann wird sie bei der OCR automatisch angelegt; wenn sie bereits im Bildstapelverzeichnis vorliegt, dann ist eine erneute OCR auch ohne weiteren ABBYY-Lizenzverbrauch möglich.[45]



42 Parameter- und Hilfsdateien HK-OCR/FREngine 9

[42] Alle folgenden Anmerkungen beziehen sich auf die im Projekt getesteten Versionen; zuletzt BIT-Alpha 2.0.38.594.

[43] Zur Funktionsweise bei Zeichenkombinationen s. 4.3.3

[44] HK-OCR verwaltet ohne Benutzereingriff im Programmverzeichnis und im Nutzer-Anwendungsverzeichnis je eine XML-Konfigurationsdatei, wo aktuelle Programm-(Session-)Einstellungen gespeichert werden, die aber im gegenwärtigen Design nicht zur Speicherung oder gar Änderung verschiedener später wählbarer Profileinstellungen vorgesehen sind, sondern lediglich – ähnlich wie bei Registry-Einträgen – beim Neustart einige Pfade und Optionen der letzten Sitzung wieder bereitstellen.

[45] Das Lizenzmodell basiert auf der seitenweisen Abrechnung, wobei auch Images Lizenzen verbrauchen, die keine Texte enthalten (z. B. Leerseiten). Die Prozessierung übergroßer Seiten (Folianten) mit viel Text benötigt unter Umständen zwei Lizenzen.

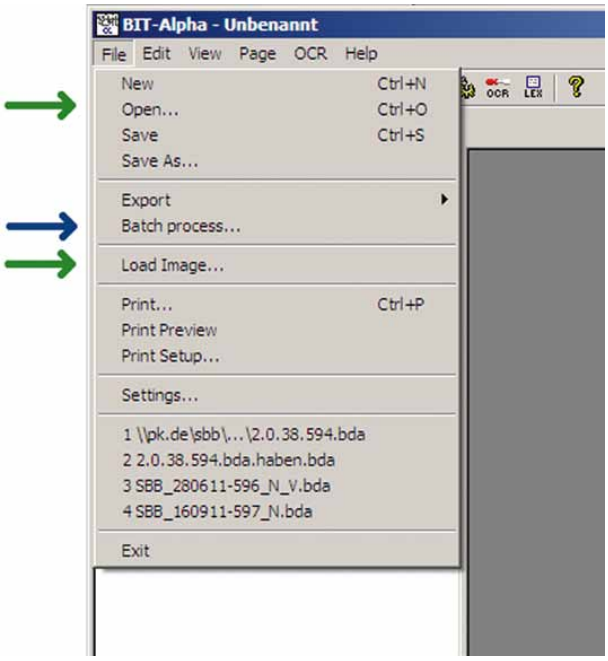
Desiderata

Für alle diese Hilfs- und Parameterdateien gilt, dass sie bisher in proprietären Binärformaten vorliegen. Für eine effektive Protokollierung, den Vergleich und die auszugsweise Übernahme von einzelnen Parametern, Mustermengen, Ersetzungsregeln oder Wortbeständen wäre es wünschenswert, dass diese Dateien jeweils

- auch außerhalb eines OCR-Prozesses mindestens passiv auslesbar sind;
- insbesondere miteinander verglichen werden können („In welchen Parametern haben sich OCR-Lauf A und OCR-Lauf B unterschieden?“);
- eine leichte Übernahme von (Teil-)Inhalten einer Konfigurationsdatei in eine andere ermöglichen (Ersatz, Kombination usw.).

Hierzu eignen sich naturgemäß am besten konventionelle offene Textformate wie Config- bzw. INI-Dateien, auch XML, die sowohl von Menschen als auch maschinell gelesen und bearbeitet werden können, die jederzeit mit Standardtextfiltern auswertbar sind und die somit auch von einem beliebigen einbettenden Workflow aus verwaltet werden können.

4.2 Start der OCR-Verarbeitungskette in der Benutzeroberfläche



43 Arbeitsbeginn mit BIT-Alpha

(a) BIT-Alpha

Die OCR einer Einzelseite startet selbsttätig,

- wenn die bda-Konfigurationsdatei ein gespeichertes Image enthält, und zwar bereits beim Öffnen der bda-Datei[46] oder
- beim Laden eines Bilds („File > Load Image“)[47].

In der Ansicht (Layer) „OCR“ wird angezeigt, ob und (durch Farben unterschieden) mit welcher Sicherheit BIT-Alpha die Zeichen erkannt hat.

Der Batch-Lauf wird (nach Öffnen/Laden der Konfigurationsdateien und/oder Einstellung aller gewünschten Parameter) angestoßen im Menü „File > Batch process ...“, wo Quell- und Zielverzeichnisse sowie die Exportformate eingestellt werden können, per Button „OK“.

(b) HK-OCR / FREngine 9

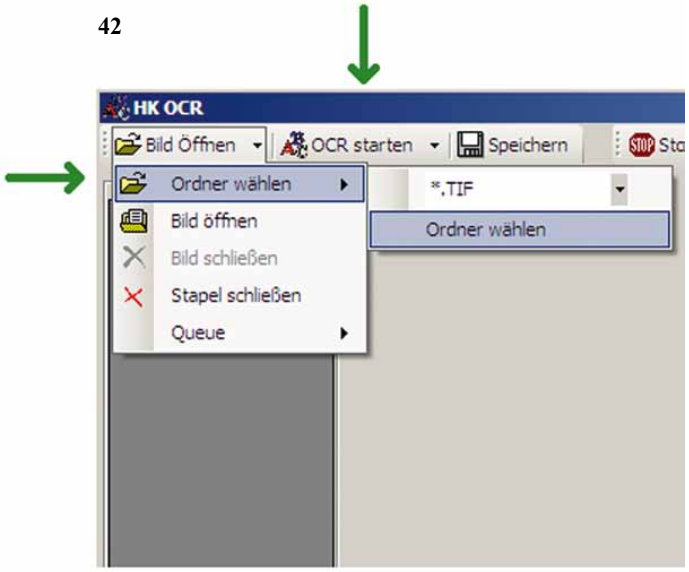
Nachdem die Grundeinstellungen eingerichtet sind (einmalige Angabe von Pfaden im Reiter „Weitere“ im Einstellungsbereich von HK-OCR[48]), wird für den aktuellen Lauf der Bild-Quellordner gewählt („Bild öffnen > Ordner wählen > Ordner wählen ...“) und werden in den Reitern „Einstellungen“ und „Export“ die aktuell gewünschten Einstellungen vorgenommen.

Danach kann die OCR einer Einzelseite gestartet werden durch „OCR starten > aktuelles Bild lesen“. Ein Batch-Lauf wird analog ausgelöst durch „OCR starten > alle Bilder lesen“.

[46] In der getesteten Version von BIT-Alpha war das automatische Abspeichern des letzten Images Standardverhalten und nicht ganz glücklich, da jedes neue Öffnen einer bda-Datei dann aufgrund des damit automatisch beginnenden OCR-Laufs viel Zeit verbraucht. Verhindert werden konnte die Speicherung des je aktuellen Bilds durch explizites Entfernen („Page > Remove“) vor dem Speichern der bda-Datei.

[47] Auch dies hat Nachteile: ein Bearbeiter könnte z. B. nach dem „Laden“ des Bilds noch Einstellungen vornehmen wollen, ehe die OCR beginnt.

[48] Wird über verschiedene Profile (ohne Administrationsrechte) auf die Software zugegriffen, ist darauf zu achten, dass in den profilbezogenen Anwendungsdaten eine vollständige alle Pfade enthaltende Konfigurationsdatei (user.config) liegt. Andernfalls öffnet sich das Programmfenster nicht und es können keinerlei Einstellungen vorgenommen werden.



44 Arbeitsbeginn mit HK-OCR/FREngine 9

Desiderata

Aus dieser Benutzerführung ergeben sich vor allem einige unnötige Beschränkungen des Batch-Betriebs. So müssen gegenwärtig alle verarbeiteten Bilder in demselben Verzeichnis liegen (das bei HK-OCR zugleich die Exporte aufnehmen muss), was dazu zwingt, große Mengen Bilddaten unnötig zu bewegen. Bildverzeichnisse werden komplett verarbeitet. Und für den gesamten Batchlauf bleibt die Konfiguration starr.

Im Abschnitt 6.4 wird darauf eingegangen, wie deutlich flexiblere Stapelverarbeitungen möglich wären, wenn die Programme die Parameter, die ihnen ohnehin übergeben werden müssen (Namen von Verzeichnissen und Konfigurationsdateien usw., die derzeit einmalig per GUI eingestellt werden und daher starr für den gesamten Batch-Lauf gelten), auch als konventionelle Kommandozeilenparameter akzeptieren würden.

4.3 Einzelne OCR-Verarbeitungsstufen (ohne Trainings-Phase)

Jede OCR setzt gespeicherte Repräsentationen optischer Muster voraus, anhand derer die Drucktypen klassifiziert und einem Zeichen bzw. einer Zeichenfolge zugeordnet werden. Die Beschaffung (Training, Bearbeitung, Kombination) der Musterbibliothek geht dem produktiven Einsatz i. d. R. voraus, wird aber in einem späteren Abschnitt dieses Kapitels (s. Kap. 4.4) beschrieben.

4.3.1 Bildvorverarbeitung und Binarisierung

Beide Produkte kommen am besten mit Bildern zurecht, die bereits mit normalisierten Helligkeits- und Kontrastwerten, ohne Ränder und Verzerrungen, in Stapeln mit vergleichbaren Weißraum-Maßen vorliegen usw. Dennoch nehmen beide Programme auch selbst etliche Bildbearbeitungen vor, was bei HK-OCR weitgehend automatisch in der eingebetteten FineReader-Engine geschieht und bei BIT-Alpha detailliert konfiguriert werden kann und muss.

(a) BIT-Alpha

BIT-Alpha bietet eine Vielzahl von die Binarisierung beeinflussenden Einstellungen, die ein großes Potential zur Anpassung an spezielle Vorlageneigenschaften bieten, aber auf eine schwer durchschaubare Weise komplex zusammenwirken.

Die Einstellungen werden in einer Konfigurationsdatei im proprietären Format „.bda“ gespeichert (s. Abb. 41 am Anfang des Kapitels). Die Arbeit in BIT-Alpha beginnt folglich mit dem Anlegen oder der Auswahl solch einer Konfigurationsdatei. Das initiale Programm-Menü „File > Open“[49] bezieht sich hierauf (nicht etwa auf ein zu lesendes Image); entsprechend bedeutet „File > Save“ das Abspeichern der aktuell im Programm BIT-Alpha geltenden Parameter in einer solchen bda-Konfigurationsdatei. Images werden dann zur Verarbeitung hinzugeladen: bei Einzelverarbeitung per „File > Load Image“, zur Stapelverarbeitung über „File > Batch process ...“ und die Angabe des Quellverzeichnisses.

Neben Einstellungen zur Bildvorverarbeitung wie Helligkeits-, Kontrast- und Gammakorrekturreglern, der Behandlung von Umrandungen usw. kann insbesondere sowohl auf Seitenebene (erste Binarisierung für Segmentierungszwecke) als auch auf der Ebene von „Regionen“ (zweite Binarisierung für Zeichenerkennung) zwischen verschiedenen Binarisierungsalgorithmen gewählt werden. Zur Auswahl stehen ein auf Farbanalyse, ein auf Farbintensitätsschwellen beruhender, ein modifizierter Niblack- und ein von B.I.T. selbst entwickelter Algorithmus. B.I.T. empfiehlt für Antiqua den eigenen Algorithmus, für Frakturschrift dagegen den Niblack-Algorithmus, der besonders die lokalen Helligkeitsunterschiede um die Zeichenkonturen herum auswertet.

Mit den von BIT-Alpha angebotenen Vorschau-Ansichten („Layern“) kann die Auswirkung aktueller Einstellungen auf die Binärbildqualität einzelseitenbezogen unmittelbar visuell beurteilt werden. Für den produktiven Einsatz fehlt hier eine Mehr-Seiten-Vorschau, um vorab zu prüfen, wie die anhand einer aktuellen Seite vorgenommenen Parameteränderungen sich auf andere Seiten desselben Stapels auswirken werden. Diese oft erheblichen Nebenwirkungen stellen sich daher häufig erst nach dem OCR-Lauf heraus.

Die binarisierten Seiten-Images können exportiert werden, aber erst beim OCR-Lauf. Nach Bedarf wurden in verschiedenen Projektphasen veränderte Binarisierungsoptionen angewandt. Die insgesamt sehr komplexe Justierung auf bestimmte Vorlageneigenschaften hin wurde meist von B.I.T. übernommen. Zu einigen Parametern sind typische Wertebereiche im Anhang aufgeführt (s. Anh. 7.3). Ein den Nutzer zur eigenständigen Konfiguration befähigendes Handbuch stand bis zum Projektende nicht zur Verfügung.

(b) HK-OCR / FREngine 9

Eingriffsmöglichkeiten für den Anwender fehlen fast völlig; die Bildnormalisierung, Entzerrung und Binarisierung laufen eingekapselt in der FineReader-Engine. Man hat lediglich die Möglichkeit, die automatische Dreh-Ausrichtung der Seite ein- oder auszuschalten. Tests der Auswirkung verschiedener Binarisierungsoptionen konnten daher nicht gemacht werden. Entsprechend gibt es keine Konfigurationsdatei für etwaige Vorverarbeitungs- oder Binarisierungsparameter. Spezielle Probleme wurden mit dem vorliegenden Material dennoch nicht beobachtet, offenbar sind die werkseitigen Einstellungen sehr gut für die meisten Anwendungsfälle geeignet. Ein Export der binarisierten Bilder war in HK-OCR nicht möglich.

Desiderata, Kriterien für Alternativen

Modularität

Bisher bietet keines der beiden Programme eine Ein- und Ausgabeschnittstelle nach der Binarisierung an.[50] Für eine modulare Verwendbarkeit der Produkte, d. h. entweder das in einem Programm X binarisierte Bild in anderen Workflows weiterzuverarbeiten oder ein bereits binarisiertes Bild direkt in die Weiterverarbeitung mit dem Programm X einzuspeisen, wäre eine Schnittstelle wünschenswert.[51]

[49] Dem entspricht ein Aufruf einer „bda“-Datei im Windows-Explorer, wenn bei der Installation unter Windows ein Dateityp „BIT-Alpha.Document“ mit der Extension „bda“ und BIT-Alpha als Standardanwendung registriert wurde.

[50] Die in BIT-Alpha gegebene Möglichkeit eines Exports des binarisierten Bilds mit der OCR bedeutet zumindest immer einen zeitlichen Mehraufwand.

[51] Im bis 2011 laufenden IMPACT-Projekt wurden u. a. Ein- und Ausgabeanforderungen verschiedener OCR-Werkzeuge beschrieben, um deren Kombinierbarkeit systematisch darzustellen; das 2012 als Nachfolgeinstitution entstandene Kompentenzzentrum kann als Kommunikationsvermittler in Anspruch genommen werden.

Evaluation

Eine schnelle Vorschau der Binarisierungsqualität mehrerer Seiten (z. B. durch Vorab-Export einer Bildmenge ohne zeitaufwendigen OCR-Lauf) könnte den Zyklus der Parameterbeurteilung vor dem OCR-Lauf wesentlich verkürzen. (s. Kap. 6.4)

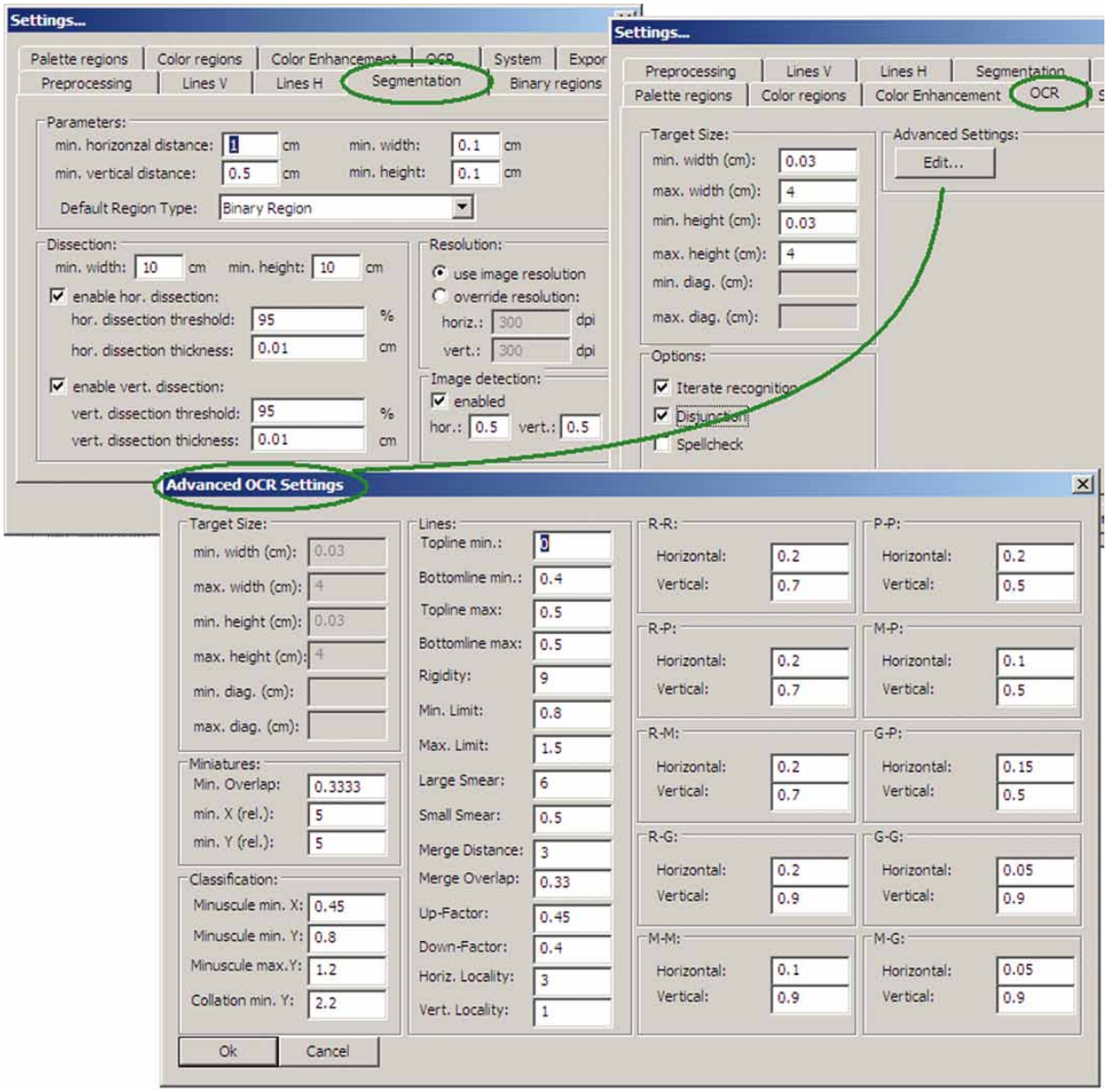
Benutzerschnittstelle

Die in BIT-Alpha fein konfigurierbaren Möglichkeiten werden in Anwenderhand erst dann wirksam eingesetzt werden können, wenn herstellerseitig eine entsprechende Dokumentation und eine Art Leitfaden zur Parametrierung bereitgestellt wird.

Von HK-OCR (bzw. der eingebauten FineReader-Engine) wäre dagegen zu fordern, überhaupt mehr Konfigurationsmöglichkeiten für den Anwender zu öffnen, die intern durch Abbyy – wie die im Durchschnitt guten Ergebnisse zeigen – durchaus genutzt und beherrscht werden, und die in FineReader-Consumer-Produkten gelegentlich auch nutzerkonfigurierbar sind. Alle in der FineReader-Engine potentiell einstellbaren Optionen[52] sollten in der Programmoberfläche zugänglich gemacht werden.

4.3.2 Segmentierung

(a) BIT-Alpha



45 Nur einige der konfigurierbaren Parameter von BIT-Alpha (File > Settings > ...)

Es gilt die gleiche Anmerkung zur Parametervielfalt wie bei der Binarisierung.

Eingestellt werden können u. a. minimale und maximale Distanzen zwischen Textblöcken und Einzelzeichen, die Behandlung von Linien, Parameter der „elastischen“ Zeilenerkennung bei gewölbten Aufnahmen, Erkennung von Nicht-Text-Regionen (Abbildungen, Ornamentik) und etliches mehr.

Die aktuell erreichte Segmentierung kann wieder mit verschiedenen Vorschau-Ansichten beurteilt werden; allerdings ist die Schlussfolgerung, wann welche Parameter geändert werden sollten, sehr kompliziert und bisher nicht in einem Handbuch dokumentiert.

Ebenfalls eingestellt werden können Parameter, die später die OCR steuern wie z. B. Größe und Abstand als irrelevant zu entfernender Punkte, der Größenbereich der zu erkennenden „Targets“, sogar relative Größenverhältnisse der Drucktypen wie Ober-, Mittel- und Unterlänge der Zeichen u. a. m.
Alle diese Einstellungen werden mit in der bda-Konfigurationsdatei, die auch die Binarisierungsparameter enthält, gespeichert bzw. daraus gelesen.

(b) HK-OCR / FREngine 9

Wie die Binarisierung verläuft auch die Segmentierung in der getesteten HK-OCR-Version vollautomatisch ohne nutzerseitige Eingriffsmöglichkeiten.

HK-OCR sieht vor dem OCR-Start offenbar keine Ansicht und Beurteilung der entstandenen Bildblöcke vor; sie können dennoch über einen Button der Korrektur-Einstellungen (Reiter „Korrektur“ > „Anzeige“ > „Blöcke“) angezeigt werden.
Layouterkennungs-Ergebnisse dürften mit in die bei der OCR pro Bild entstehenden binären Layout- und FR-Image-Dateien eingehen. Für den Anwender sind diese Informationen nicht auswertbar.[53]

Desiderata, Kriterien für Alternativen

Modularität
In beiden Produkten ist weder die Eingabe fremder binarisierter Bilder zur Segmentierung noch die Ausgabe von Segmentierungsergebnissen vor der OCR vorgesehen.

Evaluation
Für den Anwender auf dieser Stufe zugängliche Layout-Ergebnisse (Blockkordinaten o.ä.) gibt es in keinem der beiden Programme.

Benutzerschnittstelle
Für BIT-Alpha gilt die bereits im Binarisierungsschritt gemachte Anmerkung zum Erfordernis eines den Nutzer unterstützenden Handbuchs. Für etliche Parameter, die geometrische Größen wie Abstandsmaße, Höchst- und Mindestgrößen beschreiben, könnte alternativ zur numerischen Eingabe eine graphische Eingabe vor dem Hintergrund einer binarisierten Originalseite angeboten werden.

Für HK-OCR gilt hier noch mehr als im vorangehenden Binarisierungsschritt, dass alle in der FineReader-Engine potentiell einstellbaren Optionen in der Programmoberfläche auch zugänglich gemacht werden sollten. Erkennungsrelevante Eigenschaften wie Satzspiegel, Spaltenerkennung, Ausblenden von Ornamentregionen, zulässige Zeichenabstände und Wortzwischenräume usw. sollten vor der OCR eingestellt werden können. Blockgrenzen können in der derzeitigen HK-OCR-Version nicht verschoben werden.[54]

[52] Im Projekt war keine Gelegenheit, sich näher mit ABBYY-Entwicklerhandbüchern zu beschäftigen, dort scheinen aber durchaus Möglichkeiten angelegt zu sein.
[53] Probleme ergaben sich bei der Bearbeitung von Vorlagen, die über die eigentlich zu betrachtende Seite hinaus auch noch „Textreste“ der gegenüberliegenden Seite enthielten, die nach Möglichkeit ausgeblendet werden sollten.
[54] Während im späteren Nachkorrekturschritt („Erweiterte Validierung“) ein Verschieben der Blockgrenzen durchaus angeboten wird.

4.3.3 OCR

Optische Zeichenerkennung klassifiziert bildliche Darstellungen und ordnet sie einem Zeichen bzw. einer Zeichenfolge zu. Im Softwarevergleich spielte keine Rolle, wie diese Musterklassifikation im einzelnen mathematisch realisiert wurde (ob als Neuronale Netze oder mittels vorgegebener Ähnlichkeitsfunktionen über abstrakte optische Grundmuster oder Merkmalsvektoren usw.), da herstellerseitig kaum Informationen hierzu gegeben wurden und Qualitätsunterschiede absehbar nicht aus dem je verwendeten Paradigma resultieren würden. BIT-Alpha speichert offenbar verschiedene aus dem optischen Bild extrahierte Kennwerte, während FineReader vermutlich die binären Zeichen-Images selbst in der Musterdatei speichert. Für den Anwender relevant ist, dass BIT-Alpha infolgedessen die gelernten Muster-Images aus der gespeicherten Musterbibliothek nicht wiederherstellen kann.[55]

Hier wird der OCR-Schritt unter der Voraussetzung vorhandener Musterbibliotheken beschrieben; ihre Beschaffung beschreibt ein späterer Abschnitt (s. Kap. 4.4).

(a) BIT-Alpha

Vorgegebene Muster sind in BIT-Alpha nicht enthalten. Um überhaupt Texterkennung zu beginnen, muss man daher eine bereits trainierte Musterbibliothek verwenden. Diese Muster können entweder mit in einer bda-Konfigurationsdatei gespeichert sein oder in einer separaten Musterdatei im Format „.bit“ vorliegen. Wenn in BIT-Alpha sowohl (per „File > Open“) eine bda-Konfigurationsdatei als auch (per „OCR > Library > Load“) eine bit-Musterdatei geladen sind, dann gelten nur die Muster der bit-Musterdatei. Aus der bda-Konfigurationsdatei sind dann nur die Binarisierungs- und Segmentierungsparameter wirksam.

Wenn die Mustersammlung wie von B.I.T. empfohlen trainiert wurde, sind i. d. R. so genannte „Sequenzen“ für regelmäßige einfache wortunabhängige Zeichenersetzungen angelegt worden. Diese können wahlweise sogar an drei verschiedenen Orten gespeichert werden: wie die Muster entweder in der bda-Konfigurationsdatei oder in der bit-Musterdatei, oder separat in einer eigenen seq-Sequenzerdatei (dann zu laden über „OCR > Sequences > Load“)(s. Abb. 41 am Anfang des Kapitels). In der Praxis eignen sich „Sequenzen“ besonders für die Zusammensetzung von Buchstabenteilen[56] oder auch zur Auflösung von Abbreviaturen, die aus mehreren gültigen Zeichen bzw. Special Values (Sonderzeichen) bestehen.[57]

Weitere die OCR-Stufe bestimmende Parameter können in den Menüs „File > Settings > OCR“ und „OCR > Library > Settings“ eingestellt werden.[58] Hierzu gehört die Möglichkeit, den Erkennungslauf zu wiederholen sowie ein automatischer Versuch, abgetrennte Teile von Zeichenbildern alternativ als eigenes Muster zu lesen. Des Weiteren sind etliche Größenverhältnisse für die Zeichenklassifikation konfigurierbar.[59]

(b) HK-OCR / FREngine 9

In der HK-OCR-Oberfläche wird die OCR gesteuert durch die

- Wahl einer Musterbibliothek und/oder des eingebauten FineReader-Musterbestands;
- Wahl von ein bis drei Schriftfamilien (Fraktur/Antiqua/Typewriter);
- Wahl einer oder mehrerer vorgegebener „Sprachen“ oder selbstdefinierter „Sprachengruppen“ mit Wort- und Zeichenbeständen.

[55] Es ist aber beim OCR-Lauf von Einzelseiten ein (manueller) Export der bitonalen Zeichen-Images möglich; die einzelnen Image-Dateien werden dabei getrennt in Verzeichnissen gespeichert, die nach den zugeordneten Strings benannt sind, so dass außerhalb der Musterbibliothek die Bild-Zeichen-Zuordnung nachvollzogen werden kann.

[56] Wenn z. B. die Zeichensegmentierung dazu neigt, in blassem Druck Beine der Kleinbuchstaben „m“, „n“, „u“ zu separieren oder „w“ in „v“+„i“ aufzuteilen und dies in der Parameterkonfiguration nur um den Preis einer generell schlechteren Zeichenseparierung zu ändern wäre, dann können beispielsweise die einzelnen Beine als eigene Muster („special values“) trainiert werden und durch Sequenzen zusammengefügt werden. Sequenzen sind daher oft eng auf die trainierten Muster bezogen, d. h. andere Musterbibliotheken verlangen oft andere Sequenzen.

[57] Es bot sich z. B. an, das „runde r“ als *special value* zu codieren, um es im *sequencer* mit folgendem „c.“ zu „etc.“ aufzulösen, sonst aber zu „r“. Historische Abbreviaturen können gut als eigene Muster trainiert werden.

[58] Erstere werden wie Binarisierungsoptionen in bda-Dateien gespeichert; letztere wie Muster in bda- oder bit-Dateien.

[59] Vergleiche in der Parameterübersicht in 7.3 folgende Kategorien: Target Size (minimale und maximale akzeptierte Zeichengröße); Miniatures (zur Erkennung von Initialen); Classification (je nach Variabilität der Schriftform relativ zur Minuskelhöhe einstellbare Größenverhältnisse von: [P]atte (senkrechtes Bein bzw. kleiner senkrechter Strich, z. B. i ohne Punkt), [M]inuscule (Kleinbuchstabe), [G]rand (Großbuchstabe bzw. großer Strich), [R]este (kleineres Element als Patte, Minuscule oder Grand, z. B. „.“ oder „-“)). B.I.T. weist darauf hin, dass die Parametrierung dieses Menüs grundsätzlich von B.I.T. selbst und nicht vom Benutzer vorgenommen wird. Die Einstellungen sind für Fraktur, Kursivschrift der Renaissance oder gerade Antiqua unterschiedlich.

Die FineReader-Engine bringt einen Grundbestand an Mustern (bezeichnet als „Builtin-Patterns“) auch für Fraktur mit, man kann also bereits Texterkennung realisieren, bevor eine Musterdatei geladen wird. Für Mainstream-Frakturschriften besonders des 19. Jahrhunderts sind diese *built-in patterns* oft völlig ausreichend und es ist durch eigene Muster keine Verbesserung der Erkennung mehr zu erwarten. Anders gesagt: Es muss nur dann eine eigene Mustersammlung eingesetzt werden, wenn – wie bei Alten Drucken oder bei speziell gestalteten Lettern – die *built-in patterns* den Drucktypen der Vorlagen nicht gut genug entsprechen.

Trainierte Muster sind in Musterdateien im Finereader-Format „.ptn“ gespeichert (s. Abb. 42 am Anfang des Kapitels). Genau eine solche Musterdatei kann in HK-OCR im Reiter „Einstellungen“ ausgewählt und in der Checkbox „Musterdatei verwenden“ aktiviert werden – wahlweise allein ohne die FineReader-Muster (dann Checkbox „interne Muster verwenden“ deaktivieren) oder mit ihnen zusammen (Checkbox „interne Muster verwenden“ aktivieren).[60] Ein Mischen bzw. Zusammenfügen mehrerer in ptn-Dateien gespeicherter Musterbestände ist in HK-OCR nicht möglich.

Die Auswahl der Schriftfamilie(n) erfolgt über Checkboxen „Normal“, „Fraktur“ und „Schreibmaschine“. Es ist nicht erkennbar, ob dahinter nur eine Kombination von Muster-Teilbeständen steckt[61] oder ob weitere interne Parameter betroffen sind.

Die Wahl einer „Sprache“ bzw. „Sprachengruppe“ bestimmt vor allem die Lexik und den Satz zulässiger Zeichen. Auch wenn eine „Sprache“ „OldGerman“ zur Ausstattung gehört, war die Erkennungsleistung bei Einsatz einer speziell auf den Erscheinungszeitraum (17. Jahrhundert) und die Gattung bezogenen Wortliste besser. Hierzu musste aufbauend auf der vorgegebenen Sprache „OldGerman“ eine eigene „Sprache“ (base language) und „Sprachengruppe“ (text language) angelegt werden, in diese konnte dann der passende Wortbestand importiert werden.

Beobachtungen ließen vermuten, dass der Wortbestand recht früh bei der OCR in die Zeichenerkennung eingeht; mangels exportierbarer Zwischenstufen konnte das nicht genauer untersucht werden. Auch bei vermeintlicher Benutzung ausschließlich des eigenen Wortbestands gab es gelegentlich Erkennungsleistungen, die am ehesten mit weiterhin aktiven FineReader-Wortlisten erklärbar wären. Auch das Verhalten hinsichtlich der prinzipiell in der Konfiguration angebbaren erlaubten und unerlaubten Zeichen einer „Sprache“ ist nicht transparent genug bzw. wurde im Pilotprojekt nicht vollständig verstanden.[62]

Im OCR-Lauf werden von der FineReader-Engine proprietäre sogenannte Layout-Dateien angelegt, die offenbar variierenden Parametern des vorzunehmenden OCR-Laufs entsprechen. Neben diesem binären Format entstehen kurze, als Text lesbare Protokolldateien „layout.txt“ mit einigen statistischen Kenndaten zum OCR-Lauf des jeweiligen *layouts*. Es können pro Image mehrfache *layouts* angelegt werden, auf die dann wahlweise zurückgegriffen werden kann. HK-OCR lässt auch eine automatische Wahl des – nach einer internen Statistik vermeintlich erkannter Zeichen – „besten“ *layouts* zu (Reiter „Einstellungen“, Feld „Qualitätsoptimierung“). Man kann dadurch offenbar einige OCR-Parameter variieren, ohne allerdings zu wissen, welche und in welcher Richtung.

Eine Besonderheit der FineReader-Engine ist die bei der OCR ebenfalls entstehende FR-Image-Datei, die offenbar die volle (Farb-)Bildinformation und einige Zusatzinformationen enthält.[63] Im seitenbezogenen ABBYY-Lizenzmodell verbraucht die Erstellung jeder FR-Image-Datei eine Lizenzeinheit. Diese FR-Image-Datei kann später für erneute OCR-Läufe derselben Seite wieder verwendet werden, ohne dass dann eine neue Lizenzeinheit benötigt wird.[64]

[60] Achtung: Ohne Aktivierung der eigenen Muster (Checkbox „Musterdatei verwenden“) erscheint zwar die Checkbox „interne Muster verwenden“ inaktiv (ausgegraut), die internen Muster sind dann aber automatisch aktiv!

[61] Dazu passt nicht, dass auch bei alleiniger Nutzung einer eigenen (zwangsläufig ungeteilten) Musterbibliothek ein Unterschied der Fraktur-Erkennungsrate bei Zuwahl der Antiqua („Normal“)-Schrift beobachtet wurde. Allerdings schien die „Deaktivierung“ der *built-in patterns* ohnehin nicht vollständig.

[62] Über die Zeichen-Listen in den Menüs „Extras > Sprachen editieren > eigene Sprachen > unzulässige Zeichen“ und „Extras > Sprachen editieren > eigene BaseLanguages > Erkennungssatz“ gelang es nicht immer, die zulässigen und nicht zulässigen Zeichen wie erwartet zu steuern. Bei einem offenen Textformat der Sprachdefinitionen würde dieses Problem vermutlich gar nicht erst auftreten oder wenigstens leicht identifizierbar sein.

[63] Entsprechend liegt deren Dateigröße ein wenig über der Dateigröße der originalen, nicht binarisierten TIFF-Bilder, so dass im Dateisystem des Bilddatenverzeichnisses entsprechend Platz vorgehalten werden muss.

[64] Zu beachten ist allerdings, dass spätere Wiederholungen des OCR-Gangs zum Teil nur bei identischen Pfaden (absoluten Speicherorten) der Muster- und Sprach-Dateien möglich sind, und dass die Wiederverwendbarkeit nur für dieselbe Version der FR-Engine zugesichert wird.

Desiderata, Kriterien für Alternativen

Insbesondere sollte die jederzeitige Kombinierbarkeit separater Muster-Teilbestände ermöglicht bzw. konsequenter angeboten werden. Eine gute Anleitung und Dokumentation der Wirkung jedes einstellbaren Parameters wäre die Voraussetzung, die Möglichkeiten auszuschöpfen. Hinsichtlich der Benutzerschnittstelle konzentrieren sich die Anforderungen an die Programmoberfläche auf die Trainingssituation (s. Kap. 4.4).

4.3.4 Lexikalische Nachkorrektur

Idealerweise wird eine Wortliste aller in der Vorlage vorkommenden Wörter und Wortformen benötigt, die alle in der Vorlage nicht vorkommenden Formen nicht enthält. Strikt zu vermeiden sind falsche Wortformen, da sie nicht nur die Akzeptanz von Lesefehlern erhöhen, sondern auch Fehlkorrekturen provozieren.[65] Ähnlich problematisch sind Wortfragmente, wie sie aus Trennungen am Zeilenende resultieren können – es bedarf gründlicher Überlegung, ob und wann solche Wortteile ins Korrekturlexikon aufgenommen werden sollten.

Neben den in Kapitel 3 genannten grundsätzlichen Kriterien kann die Eignung von Nachkorrekturschritten davon abhängen, inwieweit Wort- und Zeichenkoordinaten an den von der Korrektur betroffenen Stellen mitgepflegt werden oder nicht, bzw. ob die Korrektur überhaupt erst auf abgeleiteten Formaten erfolgt. Allerdings sind korrekturbedingte Koordinaten-Ungenauigkeiten im praktischen Gebrauch (Suchanfragen, Finden der Stelle im Faksimile) meist unproblematisch, da auch die korrigierten Zeichen bzw. Wörter sich weiter in unmittelbarer Nähe der nicht mehr ganz zutreffenden Koordinaten befinden.

Eine linguistische Korrektur ist in beiden Produkten nicht enthalten, bzw. es müssten linguistische Kriterien vorab bei der Erstellung der Wortformenliste berücksichtigt werden.

(a) BIT-Alpha

Vor endgültiger Koordinatenspeicherung: automatische lexikalische Korrektur

Der OCR-Lauf erfolgt mit automatischer lexikalischer Korrektur, wenn in „File > Settings > OCR“ die Checkbox „Spellcheck“ aktiviert und per „OCR > Lexical Correction > Load“ eine lx2-Datei (s. Abb. 41 am Anfang des Kapitels) mit Wortliste und Ersetzungsparametern geladen wird. Das Bearbeitungsfenster für diese Einstellungen erreicht man im Menü „OCR > Lexical Correction > Edit“.

Sowohl in die Pflege der Wortliste als auch besonders in die Justierung der Ersetzungskoeffizienten lohnt es sich zu investieren, da hierdurch die blinde Ersetzung unbekannter Wörter durch *irgendein* bekanntes Wort auf die wahrscheinlicheren Alternativen hin gelenkt werden kann.

Die Wortliste sollte hinsichtlich Wortschatz und Schreibvarianten zwar so eng wie möglich am Textmaterial erstellt werden, aber auch Vollständigkeit anstreben – was besonders dann vorrangig ist, wenn ein „automatisches Mustertraining“ (beschrieben im Abschnitt „OCR-Musterbereitstellung“) genutzt wird.

Die Ersetzungskoeffizienten (Gewichte für Zeichenersetzungen, die zur Überbrückung der eingestellten maximalen Wortdistanz kombiniert werden dürfen)[66] sollten empirisch anhand der OCR-Fehler eingestellt werden, die in einem OCR-Stichprobenlauf des konkreten aktuellen Materials tatsächlich häufig festzustellen waren.

Nach endgültiger Koordinatenspeicherung: manuelle Korrektur für Export in „Leseformate“

In der Textansicht („View > Layer > Text“) bietet BIT-Alpha eine einfache Editiermöglichkeit, indem beim Anklicken eines Worts eine Eingabemaske erscheint. Die Arbeitsweise ist dem Vorgehen beim Training analog und intuitiv verständlich. Das Ergebnis der Korrektur kann aber nur in den für Einzelexport zur Verfügung stehenden Exportformaten gespeichert werden („Page > Export“), derzeit sind das insbesondere PDF, Text und HTML; das XML-Exportformat (hier: ALTO) gehört nicht dazu.

[65] Eine unbesehene Übernahme von historischen Transkriptionen bzw. E-Texten ins OCR-Lexikon ist daher auch bei einer „sehr geringen“ Fehlerquote problematisch. Ein Schreibfehler pro Seite kann in 1.000 Seiten Text zu 1.000 falschen Worteinträgen führen. In der OCR-Korrektur ebenfalls nur mit Vorsicht einzusetzen wären - grundsätzlich interessante - halbautomatisch generierte historische Wörterbücher, die von sprachgeschichtlich bekannten Lautwechseln und morphologischen Regelmäßigkeiten ausgehend historischen Wortbestand aufzubauen bzw. Korpora zu ergänzen versuchen, vgl. NEUMANN 2008.]

Erreichbares Ergebnis:
Man erhält ein koordinatenrichtiges XML-Ergebnis also nur in der Güte, die per automatischer Korrektur erreichbar war.

(b) HK-OCR / FREngine 9

Vor endgültiger Koordinatenspeicherung: manuelle Korrektur („Validierung“)
Beim Laden eines OCR-erkannten Bilds bietet der Reiter „Korrektur“ im Einstellungsbereich von HK-OCR die Anzeige einer Liste der von der OCR erkannten Wörter nebst einem Korrektureingabefeld an. In der Wortliste kann mit Cursortasten intuitiv auf- und abwärts navigiert werden, zugleich wird die Fundstelle im Faksimile farbig unterlegt und ein vergrößerter Ausschnitt der Trefferumgebung unterhalb des Seitenfaksimiles dargestellt. Der Korrekturgang („Validierung“) legt eine eigene Layout-Datei an; es kann demnach, solange der Zustand des OCR-Arbeitsverzeichnisses konsistent bleibt, wahlweise auf die OCR-Versionen vor und nach der Korrektur zurückgegriffen werden.

Die im Ergebnis der Korrekturen bzw. Worttrennungen und -zusammenlegungen angepassten Koordinaten werden in die Finereader-XML-Ausgabe eingepflegt.

Über das Menü „Erweiterte Validierung“ ist hier in der Nach-OCR-Phase sogar eine manuelle polygonale Blockgrenzen-Veränderung möglich, um z. B. aus dem Textblock herausgefallene Zeichen mit einzubeziehen oder fälschlich einbezogene Ornamentik auszuschließen.[67]

Automatische lexikalische Korrektur
Einige Korrektur-Leistungen werden offenbar OCR-nah bereits in Verbindung mit der Zeichenerkennung erbracht, wie der Einfluss der gewählten Wortliste auf die Erkennung zeigt. Ein separater automatischer Korrekturschritt ist sonst nicht vorgesehen; automatische lexikalische oder linguistische Nachbearbeitung kann also nur mit externen Werkzeugen erfolgen, folglich ohne – oder höchstens mit interpolierend schätzender – Koordinatenpflege.

Erreichbares Ergebnis:
Wenn der manuelle Korrekturaufwand geleistet werden kann, erhält man also ein koordinatenrichtiges Fertig-Ergebnis im FineReader-XML-Format und dessen Ableitungen, d. h. allen wählbaren Exportformaten.

Desiderata, Kriterien für Alternativen

Gebraucht wird auch hier zuallererst eine Dokumentation der Konfigurationsmöglichkeiten und ihrer Auswirkungen.

Für eine sachgerechte Pflege der Wortbestände und Ersetzungsregeln scheint das ungehinderte Vergleichen, Verändern, Übertragen und Kombinieren von beliebig großen Teilen solcher Wort- und Regellisten unverzichtbar. Beides wird im Regelfall vollständig vom Anwender bereitgestellt und ist hinreichend in Textzeilen beschreibbar. Proprietäre Binärformate, die nach einmaligem Einlesen einer Wortliste vom Anwender selbst weder gelesen noch frei bearbeitet werden können, entziehen dem Anwender dagegen ohne Notwendigkeit die Kontrolle über den ihm gehörenden Bestand. Wortbestände und Ersetzungsregeln könnten dabei unabhängig voneinander gespeichert sein. Gewichte für die Ersetzung eines Zeichens durch ein anderes sollten richtungsabhängig auch unterschiedlich eingestellt werden können (etwa wenn die Ersetzung von „1“ durch „i“ erleichtert werden soll, nicht aber die Ersetzung von „i“ durch „1“).

Modularität
Prinzipiell sind lexikalische Korrekturschritte Operationen auf Textdaten und daher zumindest dann nicht zwingend an das OCR-Werkzeug gebunden, wenn keine rückkoppelnden Abgleiche mit den optischen Vorlagen mehr erfolgen. Eingabeseitig läge daher nahe, durch eine Schnittstelle zum Korrek-

[66] Hoher Koeffizient bedeutet: Ersetzung bevorzugen, niedriger Koeffizient hingegen: Ersetzung „erschweren“.
[67] Es ist nicht recht einzusehen, warum das nicht auch schon für die Segmentierung vor der OCR möglich sein sollte. Offenbar ist die Funktionalität der erweiterten Validierung von der gewählten Lizenzierung abhängig; sie stand später im Projekt nicht mehr zur Verfügung.

turschritt die Einspeisung von bereits OCR-gelesenem Material zu ermöglichen, ohne dass hierzu ein erneuter OCR-Lauf und der nochmalige Zugriff auf das Image Voraussetzung sind.[68] Und warum sollen Korrekturmodule mit speziellen Stärken (wie die durch explizite Gewichte sehr fein steuerbare automatische Korrektur von BIT-Alpha oder die praktische Bedienerführung der manuellen Korrektur in HK-OCR) sich auf die Verarbeitung nur der in der eigenen OCR-Software erzeugten Texte beschränken?

Evaluation

Aufgrund der in Ansätzen gegebenen statistischen Auswertung[69] sowie der vielfältigen theoretisch einsetzbaren Möglichkeiten zu quantitativen Vergleichen der Versionen vor und nach einer Korrektur wäre es wünschenswert, wenn die Korrekturmodule selbst eine konfigurierbare Auswahl leicht handhabbarer Kennwerte verfügbar machen und vor allem eine Hilfe zur Interpretation anbieten würden.[70] Andererseits lässt das Text-(XML-)Format der Ergebnisse auch Fremdwerkzeuge zur Evaluation zu. Soweit entsprechende Tools zur Verfügung stehen, wäre das für Anwender eine potentiell unabhängigere und objektivere Alternative.

Benutzerschnittstelle

Auch wenn hier keine konkreten Anforderungen an eine ideale Korrekturbenutzerführung skizziert werden können: Die Bedeutung der Ergonomie eben des Korrektur-Arbeitsgangs kann nicht hoch genug angesetzt werden, wenn aus kompromissbehafteten Massen-OCR-Produktionen präsentierbare Volltexte hergestellt werden sollen. Das bezieht sich nicht nur auf die prinzipiell erreichbare Arbeitsgeschwindigkeit, sondern auch auf einen einarbeitungsarmen Einstieg externer, womöglich nur sporadisch mitarbeitender Korrektoren.

4.3.5 Semantische Erschließung: Strukturdaten und Named Entities

Weder Strukturerkennung (angefragt waren textsortenspezifisch z. B. Kolumnentitel, Kustoden, Bogensignaturen u. a.) noch die Auszeichnung von „Named Entities“ aufgrund vorgegebener Listen (Orte, Berufe, Krankheiten, Bibelstellenangaben) wird bisher durch die getestete Software unterstützt.

Für die Projektphase war zunächst vorgesehen, den Herstellern ein für 2011 innerhalb der SBB PK und der DBV-AG Alte Drucke erreichbar scheinendes gemeinsam abgestimmtes Application Profile, das die umfangreichen Möglichkeiten der TEI hinsichtlich Elementen und Attributen sinnvoll eingrenzt, vorzugeben. Zu dieser Abstimmung, sicher eine anspruchsvolle Aufgabe,[71] war es während der Projektzeit nicht mehr gekommen, so dass diese Entwicklungsleistung nicht mehr termingerecht verlangt werden konnte.

Für die getestete Software konnte eine Ausgabe in semantisch orientierten Formaten wie TEI daher nur sehr oberflächlich ohne die genannten bibliothekarischen Zusatzinformationen aus den Exporten abgeleitet werden. Die Dienstleister boten entweder eine manuelle Erfassung oder die Programmierung von entsprechenden Zusatzmodulen an, was aber mit weiteren Kosten verbunden gewesen wäre und mangels Demonstration der Leistungsfähigkeit solcher Zusatztools für den aktuellen Test auch nicht angezeigt schien.

Insoweit zur Erkennung von Strukturinformation und Named Entities reine Textmerkmale ausreichen, kann eine diesbezügliche Anreicherung der OCR-Texte selbstverständlich auch nachträglich außerhalb der ursprünglichen OCR-Software erfolgen.

[68] BIT-Alpha führt immer eine Neu-OCR-Analyse durch; HK-OCR kann bei Vorliegen des Images, der Layout-Datei und der Status-XML-Datei im OCR-Arbeitsverzeichnis auf den letzten OCR-Lauf zurückgreifen.

[69] So möglich in statistischen Attributen der XML-Formate und in den layout.txt-Dateien von HK-OCR/FineReader.

[70] Gefragt wären kurze Rekapitulationen des Für und Wider von Anteilsquoten „lexikalischer“ Tokens am Text, von Abdeckungs -Zahlen zum Wörterbuch, der Signifikanz von Zeichen- vs. Wortfehlern usw. - Trotz aller Mängel jedes automatisch gewonnenen Evaluierungskennwerts sind diese nutzbar, wenn sie richtig verstanden werden: „... it will give an overview of where the OCR engine thinks it‘s succeeding and where it thinks it‘s failing. If there are wide discrepancies between one batch of files and another, the software log will allow the institution to prioritise those files where OCR accuracy is low, and to manage (and hopefully mitigate) those discrepancies.“ (ANDERSON 2010, S. 2)

4.3.6 Export

Von OCR-Software wird man grob zwei Hauptexportmöglichkeiten erwarten:

- ein druckbares, im Idealfall sowohl das Faksimile als auch den Volltext enthaltendes Dokument in einem verbreiteten Format wie PDF, RTF oder HTML oder
- ein die volle Text-, Layout- und Strukturinformation enthaltendes maschinell auswertbares Dokument z. B. in XML, aus dem der Anwender alle weiteren Formate und Funktionen (Volltextpräsentation? Druckpublikation? Hervorheben der Trefferstelle im Faksimile? Datenbank? Suchindex für bloße Volltextsuche?) generieren kann.

(a) BIT-Alpha

Für den OCR-Export können die Formate (ALTO-)XML, PDF, HTML, UTF8-Text und ein proprietäres Format „B.I.T. portable“ eingestellt werden; dazu können die Farb- und binarisierten Bilder exportiert werden. Das PDF-Dokument enthält das farbige Faksimile der Seite mit dem für eine Volltextsuche mit Highlighting der Fundstelle hinterlegten erkannten Text, sowie die bitonalen Regionen, die ebenfalls durchsuchbar als einzelne PDFs aufgerufen werden können.

Der Export der Einzelseiten-OCR wird entweder über das Menü „Page > Export“ oder per Kontextmenü (Rechtsklick auf die Seite) veranlasst. Für Batch-OCR müssen die gewünschten Formate im Menü „File > Batch process ...“ ausgewählt werden. Ein automatischer Export der den erkannten Zeichen entsprechenden bitonalen Bildsegmente als einzelne, nach dem Zeichenwert benannte Bilddateien in eine Art Typenbibliothek ist lt. Angaben von B.I.T. gegenwärtig in Vorbereitung. Für eine kurze Beispielseite mit sechs Zeilen Text ist ein Quelltext-Screenshot des ALTO-Outputs im Anhang (s. 7.1) abgebildet.

(b) HK-OCR / FREngine 9

Sowohl für Einzelseiten- als auch für den Batch-Export gelten die im Reiter „Speichern“ im Einstellungsbereich aktivierten Formate; neben FineReader-XML und einem kompakteren, wortorientierten XML wird RTF (mit bildlicher Wiedergabe von nicht als Text erkannten Regionen) und PDF angeboten.[72]
Der Einzelseiten-Export für die gewählten Formate wird bei einer geladenen OCR-gelesenen Seite mit dem Menü „Speichern“ ausgelöst; im Batch-Lauf erfolgt der Export automatisch. Für eine kurze Beispielseite mit sechs Zeilen Text sind Quelltext-Screenshots der genannten XML-Formate im Anhang (s. 7.1) abgebildet.

Desiderata, Kriterien für Alternativen

Aufgrund der vielseitigen Verwendungsszenarien für OCR-Volltext wird jeder Satz von Exportformaten gelegentlich Wünsche offen lassen. Den Dienstleistern wurde deshalb als Anregung übermittelt, zusätzlich zu den ausprogrammierten Ausgabeformaten eine templateartige Schnittstelle anzubieten, für die der Benutzer selbst die Umwandlung des primären XMLs bereitzustellen hätte (z. B. als XSLT-Stylesheet, das einfach an ein im Programm vermutlich ohnehin existierendes XSL-Transformationsmodul zu übergeben wäre, oder als Datei von Ersetzungs-Regelsätzen aus Regulären Ausdrücken). Das könnte gerade in Batchläufen eine flexible Nutzung erlauben, ohne dass jeder Anwenderwunsch vom Hersteller ausprogrammiert werden müsste, und vor allem: ohne dass der Anwender sich schon vor Bestellung der Software auf jede Einzelheit seiner späteren Nutzungsanforderungen festlegen muss. Zwar sind Umformungen von XML-Formaten in beliebige Zielformate jederzeit auch extern möglich, wenn man sich

[71] Benötigt würde einerseits eine abschließende Definition einer als Minimalkonsens tauglichen, überschaubaren Untermenge eines aktuellen TEI-Formats (P5 oder Lite), s. a. einschlägige Vorarbeiten wie die an der HAB Wolfenbüttel für frühere TEI-Versionen prototypisch entwickelten „Barock-DTDs“, s. STÄCKER 2002, OPITZ 2002, andererseits hatten verschiedene Bibliotheken unterschiedliche Vorstellungen darüber, wie viele „technische“ OCR-Daten (Verarbeitungsparameter, Protokollwerte usw.) in das TEI-Format übernommen werden sollen und dürfen. Aus Sicht der SBB PK (Anforderungen der internen OCR-AG an ein notwendiges Standardformat wurden von Oliver Duntze formuliert) sollten folgende Daten (auch langfristig) in den Volltexten mitgeführt werden: Bildkoordinaten (buchstaben- oder wortweise), Schriftgröße und -schnitt, „Strukturdaten“ (Textblöcke auf der Seite), Alternative Lesarten, Linguistische Informationen (im Lexikon der OCR vorhanden? Zahlwort? Eigenname? etc.), Erkennungssicherheit. Hinzu kommen administrative Informationen wie Grundlage der OCR (Dateiname der Bilddatei), Datum des OCR-Laufs, verwendetes OCR-Programm, eingesetzte Wortliste und Musterdateien.

[72] Die Erzeugung des PDF-Formats ist von der Lizenz abhängig, die HK-OCR erfordert.

intensiv genug mit den erhaltenen Exportformaten auseinandersetzt. Eine im Programm bereits angelegte Schnittstelle könnte aber den Aufwand verringern, die Eingriffsstellen etwas vorstrukturieren und die Schwelle zur Gestaltung eigener Exportformate somit senken. Vermieden werden sollte jedenfalls, dass Anwender ihrerseits ihre Volltext-Anwendungen auf zufällige Eigenheiten der Ausgabeformate bestimmter Softwareprodukte abstimmen und damit den zukünftigen Spielraum eigener Entwicklungen unnötig beschränken.

4.4 Bereitstellung der OCR-Muster: Anlegen (Training) und Pflege von Musterbibliotheken

(a) BIT-Alpha

Training

Die Muster können entweder mit in einer bda-Konfigurationsdatei gespeichert sein oder in einer separaten Musterdatei im Format „.bit“.(s. Abb. 41 am Anfang des Kapitels). Nachdem die OCR einer Einzelseite gestartet wurde,[73] zeigt die OCR-Ansicht („View > Layer > OCR“), ob und – durch Farben unterschieden – mit welcher „Sicherheit“ BIT-Alpha die Zeichen erkannt hat. Das Anklicken eines beliebigen Zeichens öffnet das Trainingsfenster (Titelleiste „Verify Recognition“), wo der Zeichenwert des optischen Musters festgelegt oder geändert werden kann.

Mustern, denen kein Zeichenwert aus den beteiligten Alphabeten zugeordnet werden kann, kann ein eigener Code als „Special Value“ zugewiesen werden. Wegen der Möglichkeit, solche Sondercodes OCR-nah durch einen im „Sequencer“ (s. u.) festgelegten Zeichenwert zu ersetzen oder zu kombinieren, kann damit u. a. ein Training von Drucktypenteilen erfolgen, die in der Segmentierung gelegentlich anfallen (z. B. einzelne „Beine“ eines „n“ oder „m“).
Eine Kurzanleitung für einen einfachen Trainingsgang wird im Anhang 7.2.2 gegeben.

Maschinelle Beschleunigung bzw. Verstärkung des Trainings

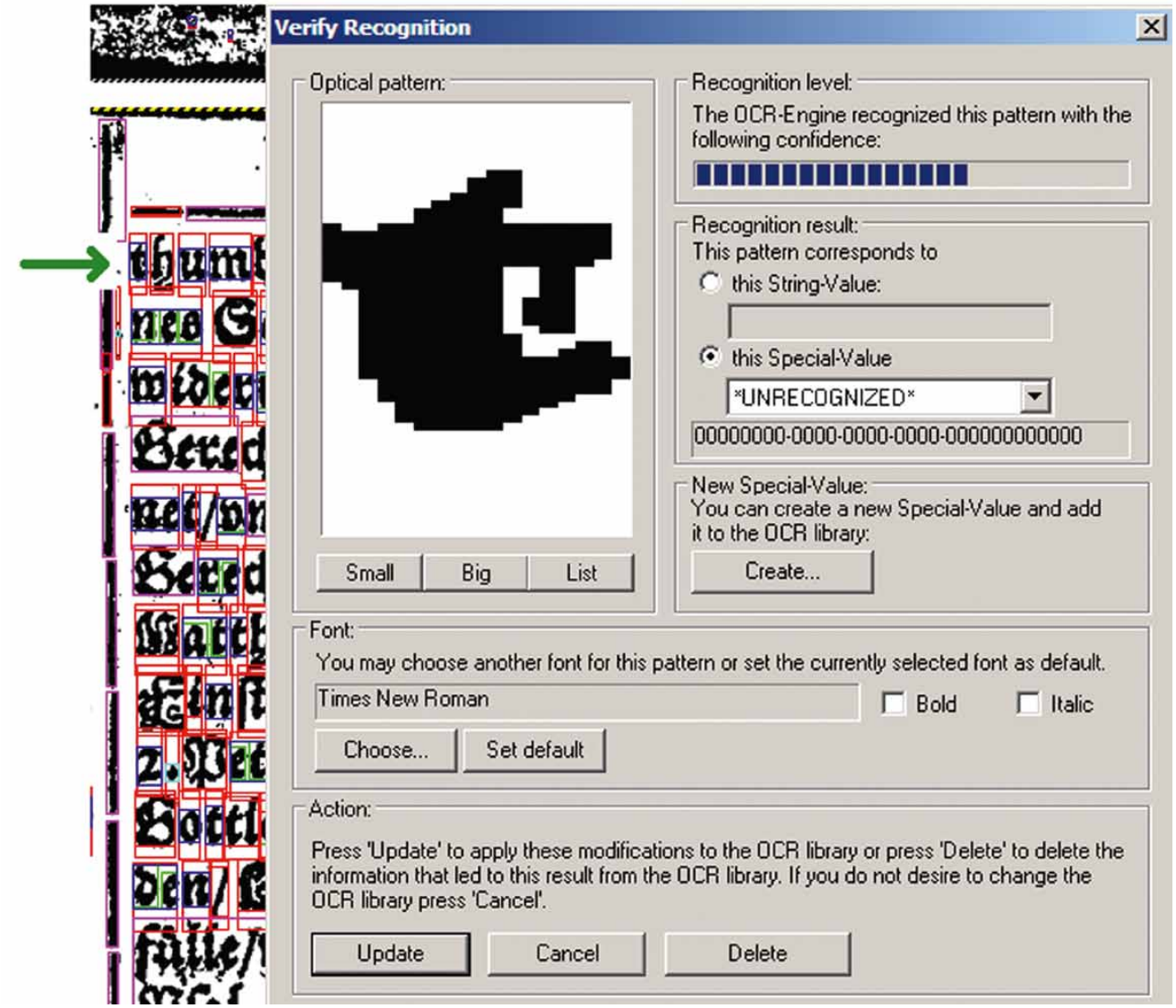
BIT-Alpha versucht mit zwei Verfahren, in manuellem Training begonnene Musterbestände zur automatischen Vergrößerung bzw. zur Verstärkung des Musterbestands zu nutzen. In beiden Fällen werden den manuell trainierten Mustern „ähnliche“ Muster automatisch hinzugefügt, was die richtige Erkennung von Drucktypen befördern *kann*, die dem manuell gelernten Zeichen nicht mehr ganz so ähnlich sind.

- „Learn Page“: Bei einer geöffneten Einzelseite können die als „fraglich“ erkannten Muster einer Seite (in der Ansicht „OCR“ durch nicht-blaue Farben gekennzeichnet), wenn sie vom Bearbeiter als korrekt eingeschätzt wurden, mit dem Befehl „OCR > Library > Learn Page“ der Musterbibliothek insgesamt auf einmal hinzugefügt werden. Dabei ist ein Konfidenzintervall einstellbar, für welche Erkennungswahrscheinlichkeiten das gelten soll.

- „Autolearn“: Etwas unmotiviert in das Menü der exportierenden Batchverarbeitung integriert findet sich die Möglichkeit, eine Anzahl Seiten ohne Export laufen zu lassen und dabei – wieder mit einem (unter „OCR > Library > Learning parameters“) einstellbaren Konfidenzintervall – die auftretenden optischen Muster des ganzen Stapels mit dem erkannten Zeichenwert automatisch der Musterbibliothek hinzuzufügen.[74] Damit können schnell große Mustermengen erreicht werden, deren Erkennungsleistung allerdings vor dem Einsatz kontrolliert werden sollte.
  In jedem Fall sollte der Zustand der Musterbibliothek vor dem „Autolearn“ gesichert werden, und es empfiehlt sich, durch vergleichende Tests das materialbezogen günstigste Konfidenzintervall zu ermitteln.

[73] Das Starten der OCR geschieht selbsttätig beim Laden eines Bildes („File > Load Image“) oder wenn die bda-Konfigurationsdatei ein gespeichertes Image enthält, dann schon beim Öffnen der bda-Datei („File > Open“).

[74] Achtung, im normalen exportierenden Batch-Lauf sollte „Autolearn“ dagegen deaktiviert werden, um die Musterbibliothek nicht ungewollt zu verändern!



46 Trainingsfenster in BIT-Alpha

Sequencer

Für Zeichenwerte, *special values* und Kombinationen aus diesen können im sogenannten Sequencer („OCR > Library > Sequencer“) feste unmittelbar nach der OCR vorzunehmende Ersetzungen definiert werden, die es beispielsweise ermöglichen, als *special value* gelernte Zeichenteile miteinander oder mit einem benachbarten Zeichen zusammenzusetzen oder Abbreviaturen zu interpretieren. Diese Ersetzungen dienen weniger der Korrektur echter OCR-Lesefehler als vielmehr der Behandlung erwartungsgemäß anfallender punktueller OCR-Ergebnisse, die lediglich einer regulären Umwandlung in einen Ziel-Zeichenwert bedürfen.

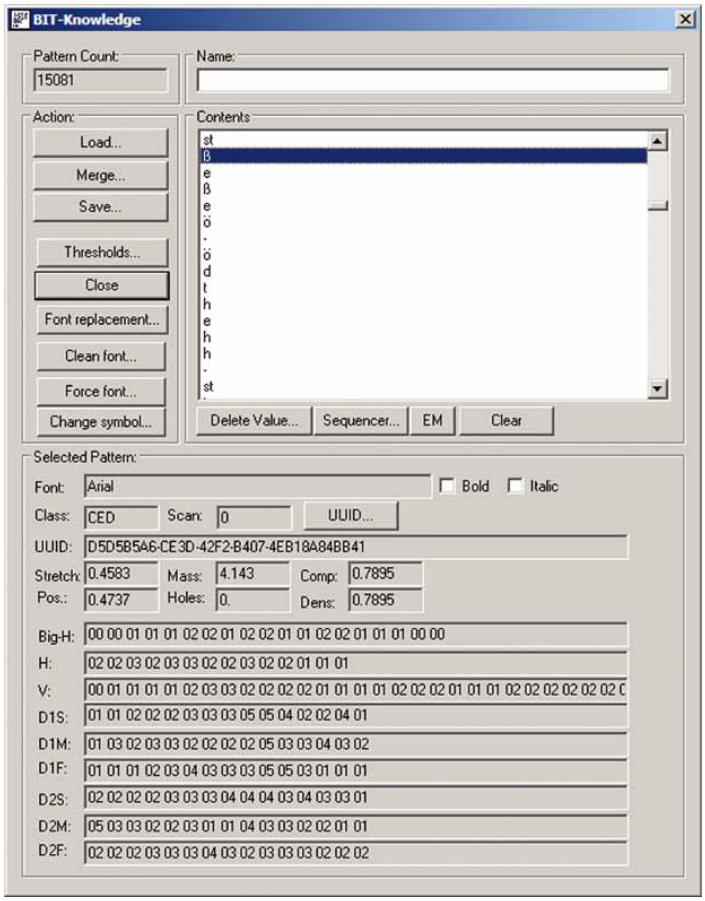
Speicherung, Bearbeitung, Zusammenführung

Muster, die im bit-Musterdateiformat vorliegen, können mit dem Mustereditor „BIT-Knowledge“ bearbeitet werden; möglich ist u. a. die Bearbeitung einzelner gespeicherter Muster (Entfernen, Zuweisung eines anderen Zeichens usw.). Hier steht allerdings keine bildliche Ansicht der Muster-Repräsentation mehr zur Verfügung, was den Umgang mit den von BIT-Knowledge angezeigten Muster-Einträgen sehr einschränkt.[75]

[75] Der häufigste Anwendungsfall war das Löschen von einzelnen falsch klassifizierenden Mustern, wie sie z. B. durch Tippfehler beim Training oder durch ein zu großzügiges automatisches Lernen angelegt werden können. Da man nur den Zeichenwert des zu löschenden Musters sieht, ist nicht ganz auszuschließen, dass hierbei vielleicht eine „gute“ Musterrepräsentation gelöscht wird; man löscht in BIT-Knowledge gewissermaßen nur noch „eines der Muster, welche für die falsche Klassifikation zu einem Zeichen x verantwortlich sind“. Für schon in BIT-Alpha bemerkte (oder dort wiederfindbare) Fehler ist die Situation etwas besser: hier kann man im Layer „OCR“ für ein konkretes Zeichensegment (anklicken) dasjenige Muster löschen, „das für diese falsche Klassifikation verantwortlich ist“. Es scheint in jedem Fall ratsam, nach dem Löschen eines Mustereintrags zu dessen Zeichenwert sicherheitshalber einige „gute“ (typische) neue Muster zu lernen.

In BIT-Alpha kann nicht mehr als eine bit-Musterdatei geladen werden, der Mustereditor „BIT-Knowledge“ dient deshalb auch zum Zusammenfügen verschiedener bit-Dateien in eine einzige, damit die Muster aus beiden genutzt werden können. Hierzu müssen die Musterbestände je selbst in einer bit-Musterdatei vorliegen;[76] dann kann zu einer ersten geladenen Musterdatei[77] eine weitere hinzugeladen[78] und die Gesamtmustermenge gemeinsam in einer bit-Musterdatei gespeichert werden.[79] Dieses Hinzumischen einer weiteren Musterdatei kann mit weiteren bit-Dateien wiederholt werden.

Achtung: Wenn außer Mustern auch Sequenzen in den beteiligten bit-Dateien gespeichert sind, werden beim Mischen nur die Sequenzen der zuerst geöffneten bit-Datei übernommen.



47 BIT-Know Mustereditor

[76] D. h. wenn sie bis dahin nur in einer bda-Konfigurationsdatei gespeichert waren, muss daraus erst eine bit-Musterdatei angelegt werden (in BIT-Alpha: „OCR > Library > Save“)

[77] In BIT-Knowledge: „Load ...“

[78] In BIT-Knowledge: „Merge ...“

[79] In BIT-Knowledge: „Save ...“

Grafische Benutzeroberfläche (GUI)

Jede Stelle auf der gelesenen Seite, z. B. in der Ansicht (Layer) „OCR“, ist per Maus frei erreichbar; man kann also wählen, ob man nur schwerpunktmäßig markante Zeichen trainieren oder den ganzen Text durchgehen möchte. Nach Anklicken eines Zeichenbereichs wird das Trainingsfenster „Verify Recognition“ aktiv. Darin wird das optische Muster binarisiert und in der der OCR-Engine vorliegenden Form noch einmal gezeigt, so dass beurteilt werden kann, ob diese Vorlage typisch genug ist und ihr ein Zeichenwert zugewiesen werden soll oder nicht. Statt eines Zeichenwerts kann einem Muster auch ein *special value* (selbstdefinierter Code) zugewiesen werden, auf den im Sequencer kontextabhängig zurückgegriffen werden kann. Sinnvoll ist das z. B. für separat segmentierte Buchstabenteile oder für mehrdeutige Zeichen wie das runde „r“, das in der etc.-Abbreviatur ausnahmsweise als „et“ aufzulösen ist.

Jedem Muster wird ein „Font“ zugewiesen, der zunächst eine Information zur Ausgabeformatierung ist. B.I.T. empfiehlt die Font-Zuweisung als Mittel zur unterschiedlichen Behandlung von Mustern verschiedener Schriftfamilien, d. h. Fraktur einen anderen Font zuzuweisen als Antiqua. Eine andere Art, beim Training die Schriftfamilie zu vermerken, gibt es nicht; insofern auch keine Möglichkeit, aus einer Musterdatei die Muster der Schriftfamilien separat zu extrahieren.

Die Anordnung der Bedienelemente im Fenster „Verify Recognition“ erfordert viel Aufmerksamkeit zur Vermeidung von Fehleingaben und teilweise umständliche Tastatur- und Mausaktionen; Hinweise zu einer möglicherweise günstigeren Ergonomie wurden dem Hersteller gegeben.

Per Standardeinstellung („OCR > Autorefresh“ aktiviert) wird nach jeder neuen Musterstringeingabe eine OCR-Analyse mit der neu kompilierten Musterbibliothek gestartet, wodurch die OCR-Ansicht sofort die nach dem Lernschritt neu bzw. anders erkannten Zeichen – wieder nach Erkennungssicherheit farbig abgestuft – anzeigt. Diese Neuanalyse benötigt je etwas Zeit; für ein zügiges Training kann „Autorefresh“ auch deaktiviert werden – dann folgt der Musterstringeingabe keine Verzögerung – und eine Neu-OCR lässt sich jederzeit per „OCR > Refresh“ manuell erzwingen.

Der Sequencer („OCR > Library > Sequencer“) wird in einer Eingabemaske „OCR-Sequence Manager“ bedient; auch hier wurden Hinweise zur Ergonomie an den Hersteller gegeben. In den getesteten Versionen von BIT-Alpha musste speziell beachtet werden, dass entgegen der Intuition eine neue Sequenz offenbar nicht erst bei Betätigung des OK-Buttons dieses Fensters angelegt und gespeichert wird, sondern bereits beim Editieren der Zeichenwerte und ohne Hinweis.

Die Oberfläche zur Musterbearbeitung in BIT-Knowledge eignet sich gut zum Bearbeiten bzw. Löschen einzelner Muster und zum Zusammenfügen zweier bit-Musterdateien, solange keine Konflikte wegen dort mitgespeicherter Sequenzen eintreten können, auf deren implizite Übernahmepriorität nicht hingewiesen wird. Eine effektive Bearbeitung von vielen Mustern auf einmal scheitert an der unübersichtlichen Auflistung (fehlende Sortierbarkeit, keine Mehrfachselektion).

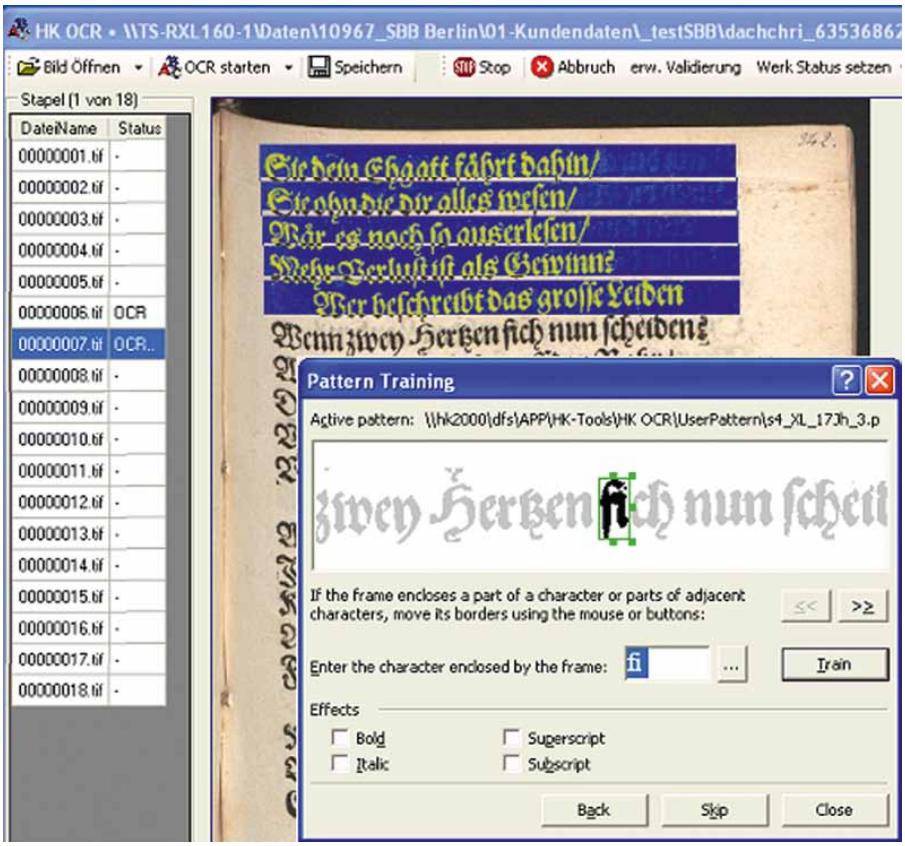
(b) HK-OCR / FREngine 9

Die FineReader-Engine bringt auch für Fraktur einen Grundbestand an Mustern (*built-in patterns*) mit; es muss also nur dann trainiert werden, wenn – wie bei Alten Drucken oder bei speziell gestalteten Lettern – die *built-in patterns* den Drucktypen der Vorlagen nicht gut genug entsprechen.

Training

Zum Training wird eine Seite wie zur OCR bereitgestellt („Bild öffnen > Ordner wählen > Ordner wählen“); vor dem Trainingslauf muss zunächst im Reiter „Einstellungen“, Feld „Erkennungsmuster“ (bei deaktivierter Checkbox „Musterdatei verwenden“) die Checkbox „Training“ aktiviert werden, und es muss eine ptn-Musterdatei erstellt bzw. – falls schon existent – ausgewählt werden (Button „..“ neben dem Textfeld mit dem Dateinamen der Musterdatei). Wie zur OCR können Sprachen bzw. Sprachengruppen und Schriftfamilien eingestellt werden. Dann kann der Trainingslauf wie ein OCR-Lauf gestartet werden („OCR starten > Aktuelles Bild lesen“).

Die recht problematische, von den FineReader-Bausteinen bestimmte Benutzerführung des Trainings erzwingt im Prinzip ein komplettes lineares Durchgehen der Seite vom ersten Zeichen zum letzten. In einem Fenster wird der Bildausschnitt um das aktuelle Segment herum jeweils binarisiert und vergrößert dargestellt, und dem markierten Muster kann ein Zeichenwert zugeordnet werden. Möglich sind nur Zeichenwerte innerhalb der für die Sprachengruppe eingestellten Zeichen – der Anwender muss also bereits vor dem Training alle im Bestand möglicherweise vorkommenden Sonderzeichen (Diakritika, Zeichen anderer Sprachen) kennen und bereitstellen. Nach Durchlaufen der Seite wird die Musterdatei gespeichert, und es kann eine weitere Seite aufgerufen werden.



48 Trainingsfenster in HK-OCR[80]

Speicherung, Bearbeitung, Zusammenführung

In einem Mustereditor-Fenster (Fenster „User Pattern“) können die in einer ptn-Datei gespeicherten Muster in der bitonalen optischen Form mit dem Zeichenwert angezeigt, gelöscht und wohl umdefiniert werden. In begrenztem Rahmen ist eine Mehrfachselektion zur Löschung mehrerer Muster auf einmal möglich. Nicht vorgesehen ist eine Zusammenführung verschiedener Musterdateien.

Grafische Benutzeroberfläche (GUI):

Speziell die von FineReader vorgegebenen Bildschirminteraktionen beschränken unnötig den Spielraum und die Effizienz beim Training und bei der Bearbeitung der Musterdateien. Entsprechende Hinweise wurden an den HK-OCR-Hersteller gegeben, der trotz Konsultationen mit ABBYY offenbar bisher keine Möglichkeit zur Bearbeitung dieser Bausteine bekommen hat. Auch eigene Nachfragen bei einer ABBYY-Regionalvertretung zeigten, dass die Benutzerfreundlichkeit der Trainingswerkzeuge nicht als aktuelle Aufgabe gesehen wird; verschiedenen während der Projektlaufzeit in Deutschland erreichbaren ABBYY-Vertretern schien die Möglichkeit eines (sinnvollen) anwenderseitigen Muster-Trainings nicht einmal bekannt zu sein.

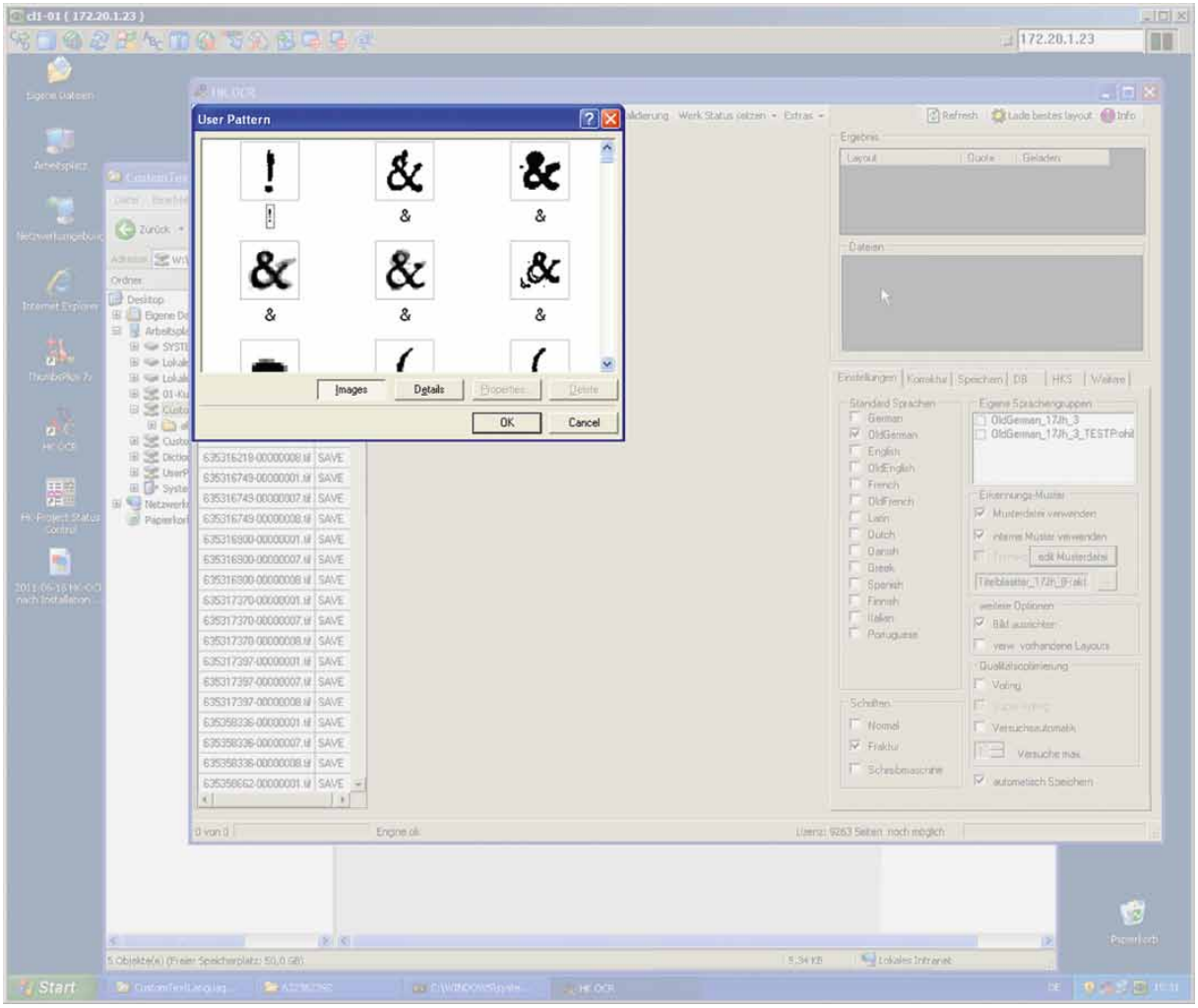
[80] Im Beispiel: Die OCR „erkannte“ die Ligatur „fi“; im Training ist Gelegenheit zur Eingabe der richtigen Zeichenfolge „si“. - Bei diesem eher schlechten Muster wird man auf ein Training aber vermutlich verzichten, um die Erkennung der „fi“-Ligatur nicht zu beschädigen.

In der Trainingssituation (Fenster „Pattern Training“ der FineReader-Engine) wird der Benutzer-Zeitaufwand durch eine ineffektive Führung durch die Bildseite immens in die Höhe getrieben. Die wichtigsten Nachteile sind:

- Es ist nicht möglich, eine Trainingssequenz an beliebigen Orten innerhalb der Seite zu beginnen und zu beenden. Die Navigation verläuft in Einzelschritten vorwärts. Man kann das Training nur am Anfang der Seite beginnen und muss daher, auch wenn man etwa nur am Ende der Seite ein paar Zeichen trainieren will, jedes Zeichen der ganzen Seite noch einmal bestätigen oder explizit überspringen (mit großem Zeitverlust und der Gefahr, beim „Durchklicken“ bereits gelernte Muster-Zeichen-Paare zu beschädigen).
- Ein Rückschreiten (zur Korrektur von bemerkten Fehleingaben) ist nur in Einzelschritten und nur innerhalb desselben Worts möglich.
- Oft werden dagegen automatisch Zeichen(gruppen) ohne Vorwarnung übersprungen, offenbar dann, wenn sie mit hohem Konfidenzwert gelesen wurden. Abgesehen davon, dass auch dort in Einzelfällen ein Korrektur- oder Trainingsbedarf bestehen kann, zwingt die ständige Unberechenbarkeit, ob nach einer Eingabe wirklich das nächstfolgende oder aber ein viel späteres Muster ins Trainingsfeld eingetragen wird, zu einer unnötigen Verlangsamung des Trainingsgangs. Bei zügiger Arbeitsweise bestätigt man per Enter-Taste oder Button leicht einen String, der auf das nach Schrittfolge erwartete Muster gepasst hätte, versehentlich für ein ganz anderes Muster. Wenn der Sprung dabei die Wortgrenze überschritten hat, ist dadurch ein Zurückschreiten zum eben eingegebenen Zeichen nicht mehr möglich, und die Fehleingabe wird unweigerlich in die Musterdatei übernommen. Eine Korrektur im Mustereditor ist zeitaufwendig und hängt von der Wiedererkennung des richtigen Zeichenbilds ab.
- Die im Fenster „Pattern Training“ angebotenen Tasten sind zumindest im deutschen Tastaturlayout nicht immer günstig, es kommt zu ungünstigen Maus-Tastatur-Wechseln und zu einer Erleichterung unerwünschter Aktionen gegenüber erwünschten.

Im Mustereditor („Edit Musterdatei“), ebenfalls einer FineReader-Komponente, fiel auf:

- Das angezeigte Fenster „User Pattern“ ist viel zu klein, es zeigt je nur 2 x 3 Muster in einer nicht änderbaren Reihenfolge und kann nicht vergrößert werden. Für einen Überblick fehlt die Möglichkeit, das Fenster so groß zu ziehen, dass wesentlich größere Mustergruppen miteinander optisch verglichen werden können.
- Die in „Edit Musterdatei“ vorgesehene Möglichkeit, gespeicherte Muster-Zeichen-Zuordnungen zu ändern, scheint noch nicht zu funktionieren. Während des Tests ist zwar das Löschen einzelner Muster gelungen, nicht aber das Ändern des zugewiesenen Strings.



49 HK-OCR Mustereditor

Desiderata, Kriterien für Alternativen

Modularität

Um die Stärken selbst trainierter auf konkrete Drucke bezogener Muster ausnutzen zu können, wird eine freie Kombinierbarkeit und Teilbarkeit von Musterbibliotheken benötigt. Hierzu wäre ein dokumentiertes, für den Nutzer verständliches Speicherformat für Muster gefragt, das u. a. Attribute der einzelnen Muster verwalten kann, nach denen Teilbestände selektiert werden können usw. Ideal wäre natürlich eine anwendungsübergreifende Austauschbarkeit trainierter Muster (s. Kap. 6.4).

Evaluation

Ehe überhaupt sinnvolle Evaluationswerkzeuge konzipiert werden können, müssen Musterbibliotheken in einer jederzeit auslesbaren und miteinander vergleichbaren Form vorliegen.

Benutzerschnittstelle

Hinsichtlich der graphischen Benutzerführung könnten beide Programme voneinander lernen. Unerlässlich sind die freie Navigation an eine beliebige Stelle sowie die freie Unterbrechung und unverzögerte Wiederaufnahme des Trainingsgangs.

Restriktionen im Alphabet der zulässigen Zeichenwerte sollte es nicht geben, da prinzipiell jede Druckvorlage, von der noch kein elektronisch durchsuchbarer Volltext vorliegt, etwas Unvorhergesehenes enthalten kann.

Der Mustereditor muss eine übersichtliche Ansicht der gespeicherten Muster bieten, d. h. sortierbar, in größenverstellbarem Fenster, mit Mehrfachauswahl und verständlich bezeichneten Aktionen.

4.5 Batch-spezifische Aspekte

Wie bereits angedeutet (s. Kap. 3.1.7), hängt die Qualität von Massen-OCR davon ab, ob die verwendeten Parameter, Musterbestände und Lexika so gut auf die Gruppe der gemeinsam zu verarbeitenden Bilder abgestimmt sind, wie es in der Einzelseitenbearbeitung möglich ist.
Die Software hat hier nur insofern Einfluss, als sie

- diese Gruppierung in irgendeiner Form unterstützen könnte, z. B. durch Kennziffern zur nachträglichen Selektion von Seiten, die gut oder schlecht auf einen OCR-Lauf angesprochen haben oder
- Konfigurationsparameter innerhalb eines Stapels zu wechseln in der Lage ist.

Außerdem wird die Flexibilität der Gruppenbildung davon beeinflusst, ob Quell-(Bild-)Dateien zur Stapelverarbeitung erst in ein gemeinsames Arbeitsverzeichnis kopiert werden müssen, oder auch wie leicht und sicher ggf. der Abbruch und die Weiterführung einer Stapel-Verarbeitung möglich sind usw.

(a) BIT-Alpha

Eine Vorab-Gruppierung wird nicht unterstützt, sie muss daher vom Anwender geleistet werden.

BIT-Alpha bietet keine für den Anwender erreichbare Auswertung von Kennziffern, die eine nachträgliche Gruppierung der Bilder in „fertige“ und „nachzuprozessierende“ erleichtern würden. Auch der XML-Export enthält keine diesbezüglich auswertbaren Attribute. Für den Ergebnistext könnte der Anwender selbst z. B. den Bezug zum Wörterbuch herstellen und eine Art Worterkennungsrate bestimmen.

Die in einem Batch-Lauf zu bearbeitenden Bilder müssen in ein und demselben Verzeichnis liegen; dann kann nach Laden der gewünschten bda-Konfigurationsdatei und den üblichen Einstellungen der Batch-Prozess per „File > Batch process ...“ konfiguriert (Exportformate festlegen usw.) und angestoßen werden. Zu beachten ist, dass das ebenfalls in diesem Dialog untergebrachte „Autolearn“ (s. Kap. 4.4) im normalen (Export-)Batchlauf deaktiviert sein muss, um die Musterbibliothek nicht ungewollt zu verändern.

Die gewählten Exportformate werden in Unterverzeichnisse eines wählbaren Exportverzeichnisses geschrieben. Wenn außer den sparsamen Text-, HTML- und XML-(ALTO)-Formaten auch das Bilder enthaltende PDF exportiert wird, ist auf ausreichend Speicherplatz im Dateisystem des Exportverzeichnisses zu achten.

Zur Fortsetzung einer abgebrochenen Stapelverarbeitung war es am praktischsten, je die bereits verarbeiteten Bilder aus dem Quellverzeichnis zu entfernen oder überhaupt neue Verzeichnisse zu benutzen.

(b) HK-OCR / FREngine 9

Eine Vorab-Gruppierung wird nicht unterstützt, sie muss daher vom Anwender geleistet werden.

HK-OCR legt bei jedem OCR-Lauf zu jedem *layout* auch Protokolldateien an, in denen statistische Kennziffern der aktuellen OCR festgehalten werden und die bei eingeschaltetem *voting* schon von HK-OCR selbst zur automatischen Auswahl des nach diesen Kennziffern „besten“ OCR-Versuchs genutzt werden. Eine genaue Dokumentation der Bedeutung der einzelnen Werte lag der Software nicht bei, sollte aber erhältlich sein. Sinnvolle Interpretierbarkeit vorausgesetzt, könnten sich diese Werte wegen des Textformats der Protokolle auch nach Stapelverarbeitungen maschinell auswerten lassen, ebenso die im Finereader-XML-Export (s. Screenshots einer FineReader-XML-Ausgabe im Anhang 7.1) enthaltenen Attribute zur Zeichenerkennungskonfidenz („charConfidence“) und zum Vorkommen des gelesenen Worts im Wörterbuch („wordFromDictionary“).[81]

Die in einem Batch-Lauf zu bearbeitenden Bilder müssen in ein und demselben Verzeichnis liegen, das zugleich das Ausgabeverzeichnis ist. Wegen der pro Originalbild entstehenden FRImage-Dateien, die stets etwas größer sind als die Originalbilddatei, muss der freie Speicherplatz im Dateisystem des Verzeichnisses noch einmal etwas größer sein als die Gesamtdateigröße der Bilder.

[81] Vgl. die Hinweise zur kritischen Anwendbarkeit solcher Kennzahlen in den Best-Practice-Guides des IMPACT-Projekts (ANDERSON 2010)

Der Status der Verarbeitung jedes Bilds aus dem Batch-Lauf wird in einer Liste in HK-OCR angezeigt und in einer XML-Datei im Ausgabeverzeichnis vermerkt. Solange diese Steuerdatei („_BildDateien.xml“) erhalten ist, kann eine abgebrochene Stapelverarbeitung leicht wieder aufgenommen werden und überspringt die bereits als verarbeitet und gespeichert vermerkten Bilder. Abbrüche traten vereinzelt spontan auf, möglicherweise wegen geringer Zugriffsverzögerungen auf ein Netzlaufwerk. Sicherheitshalber sollte nach jedem unterbrochenen Batch-Prozess die Vollzähligkeit der XML-Exporte überprüft werden; eventuell fehlende einzelne Seiten können dann leicht manuell nachprozessiert werden. Hierzu kann es notwendig werden, per Kontextmenü den in der Stapel-Übersicht von HK-OCR vermerkten „Status“ der jeweiligen Bilddatei zurückzusetzen.

Desiderata, Kriterien für Alternativen

Zu den im Folgenden angedeuteten Anforderungen werden im Abschnitt 6.4 konkrete Abhilfemöglichkeiten beschrieben.

Granularität

Batch-Verarbeitung ist aus sich heraus nicht davon abhängig, dass die zu verarbeitenden Bilder in ein und demselben Verzeichnis liegen. Außerdem müssen lange Batch-Läufe gelegentlich unterbrochen und dann nur für den Rest fortgesetzt werden, oder man will von vornherein nur bestimmte Bilder verarbeiten. Es gibt keinen nachvollziehbaren Grund, dass im Dateisystem erreichbare Quellbilder erst in ein Arbeitsverzeichnis gebracht werden müssen, und dass der Stapel vom Inhalt dieses Verzeichnisses bestimmt wird.

Es sollte zudem auch innerhalb eines Stapels möglich sein, für die Bilder wechselnde Parameter, Muster- und Wortbibliotheken vorzusehen. Die innerhalb üblicher Stapel zu beobachtenden erheblichen Qualitätsunterschiede der Fraktur-OCR rechtfertigen den Aufwand allemal, und mit etwas Glück geben bereits die Umfangsangaben im Katalog oder die Metadaten der Digitalisate für Seitenbereiche deren Schriftart und Sprache an.

Evaluation

Würde die Software geeignete statistische Indikatoren zur Abschätzung der Erkennungsgüte der einzelnen Seiten bereitstellen, dann könnten so ermittelte ungenügende Seiten nachträglich einer Neu-OCR mit anderen Mustern – oder anderen Wörterbüchern – zugeführt werden. Für Szenarien, in denen genügend Rechenzeit gegeben ist, sollte es möglich sein, heterogenes Material von vornherein parallel mit verschiedenen Musterdateien bzw. Wortlisten zu verarbeiten und aus den Ergebnissen das je geeignetste zu selektieren.

Benutzerschnittstelle

In beiden Programmen müssen Konfigurationen bisher auch dann per Bildschirminteraktion ausgewählt werden, wenn sie fertig in Parameterdateien vorliegen. Für eine Stapelverarbeitung mit vorab feststehenden Konfigurationen sollte aber neben der Bildschirminteraktion stets auch ein nicht-interaktives Anstoßen auf Kommandozeilen-, Batch-/Skript- oder Ticket-Ebene möglich sein (konventionelle Übergabe der Arbeitsverzeichnisse und der jeweiligen Parameter-, Muster- und Wörterbuchdateien).

4.6 Voraussetzungen und Randbedingungen des Softwareeinsatzes

Für beide Produkte wurden übliche Windows-PCs (Prozessoren Intel Core 2 Duo E8400 mit 3,0 GHz, Arbeitsspeicher 2GB, Betriebssystem Windows XP Professional SP 3) genutzt. Solange nur mit einer Instanz des jeweiligen Programms gearbeitet wird, haben diese Hardwarevoraussetzungen genügt; bestimmte Verarbeitungsphasen der OCR lasten den PC dabei zeitweise sehr aus, so dass von gleichzeitiger Arbeit in anderen Anwendungen abzuraten ist.

(a) BIT-Alpha

Lizenzmodell
Die Standalone-Anwendung BIT-Alpha (ebenso der Mustereditor BIT-Knowledge) wird als Anwendung lizenziert, hierzu ist eine Aktivierung unter Administratorrechten nötig.

Installations- und Betriebsvoraussetzungen
Zum Betrieb von BIT-Alpha genügt ein Einzelplatz-PC mit einem aktuellen Windows-Betriebssystem (ab XP). Eine möglichst gute Prozessorleistung (Mehrkernarchitektur) ist angeraten. Für jede Seite ist größenabhängig mit mehreren Minuten OCR-Analyse zu rechnen. Die Priorität des Programms für das Betriebssystem kann in der Benutzeroberfläche eingestellt werden, z. B. um sie herabzusetzen, wenn der PC gleichzeitig für anderes genutzt werden muss und längere OCR-Zeiten hingenommen werden können („File > Settings > System > Thread Priority“).

Angebot als Dienstleistung
B.I.T. bietet wahlweise die Komplett-OCR (vom Bild zum Exportformat) oder die Unterstützung anwenderseitiger OCR (Konfiguration der Binarisierungsparameter, Übernahme von Training usw.) auch als Dienstleistung an.

(b) HK-OCR / FREngine 9

Lizenzmodell
HK-OCR wurde als Anwendung von Herrmann & Kraemer zeitlich befristet lizenziert und ermöglicht gesondert die Neu-OCR bzw. die Korrektur („Validierung“) von Seiten, für die bereits eine FRImage-Datei im Arbeitsverzeichnis vorliegt. Zur erstmaligen OCR jedes Bilds (zugleich: Erstellung der FRImage-Datei) wird außerdem eine ABBYY-Volumenlizenz benötigt. Hierzu muss entweder auf dem Arbeitsrechner oder auf einem im Netzwerk erreichbaren Rechner der ABBYY-Lizenzmanager installiert sein und erworbene Seitenlizenzen enthalten.

Installations- und Betriebsvoraussetzungen
Sowohl zum Betrieb von HK-OCR als auch für den ABBYY-Lizenzmanager genügen Einzelplatz-PCs mit einem aktuellen Windows-Betriebssystem (ab XP). Die Installation und Aktivierung erfolgt jeweils unter Administratorrechten. Der ABBYY-Lizenzmanager kann entweder ebenfalls auf dem OCR-Rechner oder auf einem anderen per Netzwerk erreichbaren Rechner installiert werden und betreibt offenbar einen ständig laufenden Dienst. Der Lizenzmanager muss während der OCR ständig und nicht nur „einmal je Image“ erreichbar sein; auch bei nur sehr kurzzeitigem Aussetzen der Verbindung zum Lizenzmanager-Rechner wurden HK-OCR-Stapel ohne direkte Fortsetzungsmöglichkeit abgebrochen. Für jede Seite ist größenabhängig mit mehreren Minuten OCR-Analyse bzw. Exporterstellung zu rechnen.

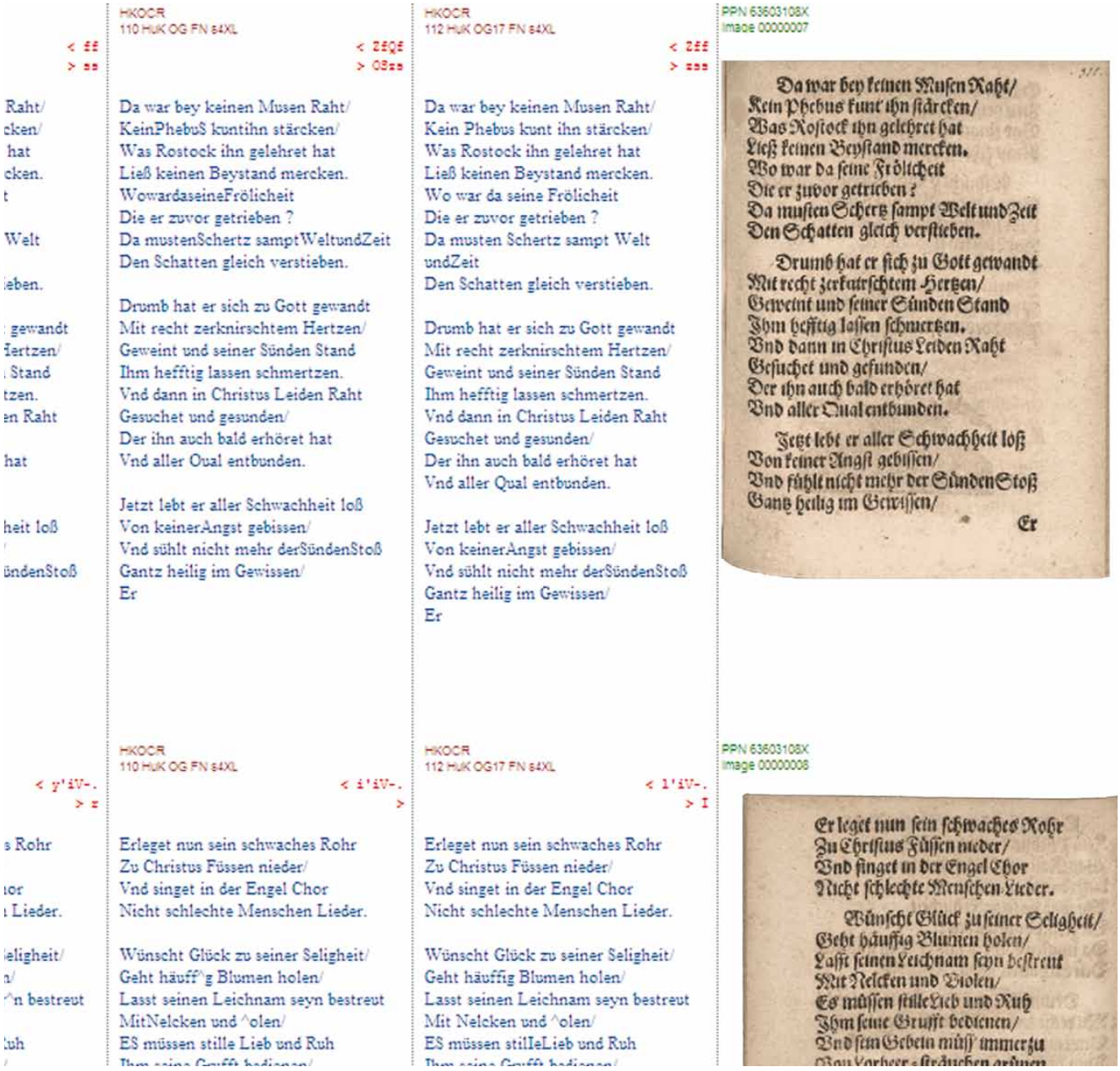
Angebot als Dienstleistung
Herrmann & Kraemer bietet die OCR auch als Dienstleistung an, ggf. auch in einem integrierten Workflow mit weiteren Komponenten.

5 Praktischer Softwaretest und -vergleich II: Optimierungsversuche durch Sortierung, Binarisierungsparameter, Training und Lexika

Methodik

Zur Beurteilung der in jeder Software variierbaren OCR-Faktoren waren viele parallele OCR-Läufe erforderlich, die sich jeweils in möglichst wenigen Bedingungen unterscheiden durften.

Als Material verwendet wurden insbesondere eine Marge aus sehr homogenen Gedichtzeilen (Druckort: Königsberg) und eine Marge, in der die zwei nächst kleineren Schriftgrade bei verwandtem Typenbild überwogen. Für bestimmte Musterbibliotheken wurden darüber hinaus bewusst die mit Schriftvarianten überladenen Titelblätter herangezogen, um möglichst schnell eine große Vielfalt zu erreichen (Vgl. Kap. 2.6, Abb. 36).



50 Tabellarischer Vergleich der OCR-Läufe; längere rote Zeichenlisten bedeuten mehr Erkennungsfehler

Für beide Schriftgruppen wurden Stichproben festgelegt, die manuell transkribiert wurden und als Referenz zur skriptgesteuerten Ermittlung von OCR-Fehlern der verschiedenen Testläufe dienten. Das Training erfolgte konsequent auf Seiten, die nicht den Stichproben angehörten.[82] Verglichen wurden die unter verschiedenen Bedingungen entstandenen OCR-Ergebnistexte einerseits durch tabellarische Gegenüberstellung der Volltext-Resultate, andererseits durch die Liste der „fehlenden“ und die Liste der „fälschlich gelesenen“ Zeichen einer Seite.

Die Längen dieser Listen dienten als einfache Maßzahlen zum relativen Vergleich verschiedener OCR-Läufe, d. h. um festzustellen, wann eine Konfigurationsänderung einen Fortschritt erbrachte und wann nicht. Als realistische „Fehlerrate“ für die Zeichen-Ebene sind diese Zahlen nicht zu interpretieren.[83]

Wegen der vielen Einflussdimensionen konnten bei weitem nicht alle interessanten Parameterkombinationen getestet werden; oft musste in einer Art Wege-Auswahl frühzeitig entschieden werden, welche günstige Parameteränderung weiterverfolgt wird und welche ungünstige nicht.

[82] Zu beachten bleibt, dass in OCR-Prozessen immer wieder eine so große Schwankung der Erkennungsrate beobachtet wird, dass jeder Schluss von der Stichprobe auf die Gesamtheit auch hinterfragt werden kann. Die Statistik fordert, dass für solche Inferenzen entweder eine echte Zufallsauswahl oder eine „repräsentative“ (Quoten-)Stichprobe vorliegen muss; beides ist für Images gescannter Buchseiten schwer herzustellen.

Zur Visualisierung der unübersichtlichen Ergebnismengen wurden die verschieden konfigurierten Batchläufe in einem Ableitungsdiagramm dargestellt, in dem die gezählten Fehleranteile nicht nur genannt, sondern auch als Farbwert zwischen Rot und Grün dargestellt wurden. So konnten die aussichtsreicheren Zweige der Versuchsanordnung schnell gefunden und auf weniger aussichtsreiche Konfigurationsgruppen früh verzichtet werden (vgl. Abb. 54 u. 58).

(a) BIT-Alpha

Gruppierung

Wegen der Möglichkeit, verschiedene Musterdateien zu mischen, ist ein zunächst separates Training relativ reiner Zeichenbestände in BIT-Alpha unproblematisch, wenn die Muster später auch kombiniert genutzt werden sollen. Etwas einschränkend wirkt das Zusammenspiel der Muster mit den eingegeben Sequenzen: verschiedene zu kombinierende Musterdateien sollten auf ähnliche Sequenzen zurückgreifen, um nach dem „Merging“ ihre Erkennungsleistung zu behalten.

Die trainierten Muster einer Schriftgröße haben bis zu einem gewissen Grad auch Zeichen anderer Schriftgrößen richtig klassifiziert, d. h. es gibt eine gewisse Toleranz gegenüber Größenabweichungen.

Binarisierungs- und Segmentierungsparameter

Wegen der Komplexität der vorzunehmenden Einstellungen wurden die bda-Dateien überwiegend vom Dienstleister konfiguriert und bereitgestellt. Zusätzlich wurden auch im Projekt einige bda-Dateien experimentell abgewandelt oder von Grund auf selbst eingerichtet, um die Wirkung bestimmter Parameter kennenzulernen. Einige Hinweise hierzu finden sich im Anhang 7.3; ein erläuterndes Handbuch bleibt einzufordern.

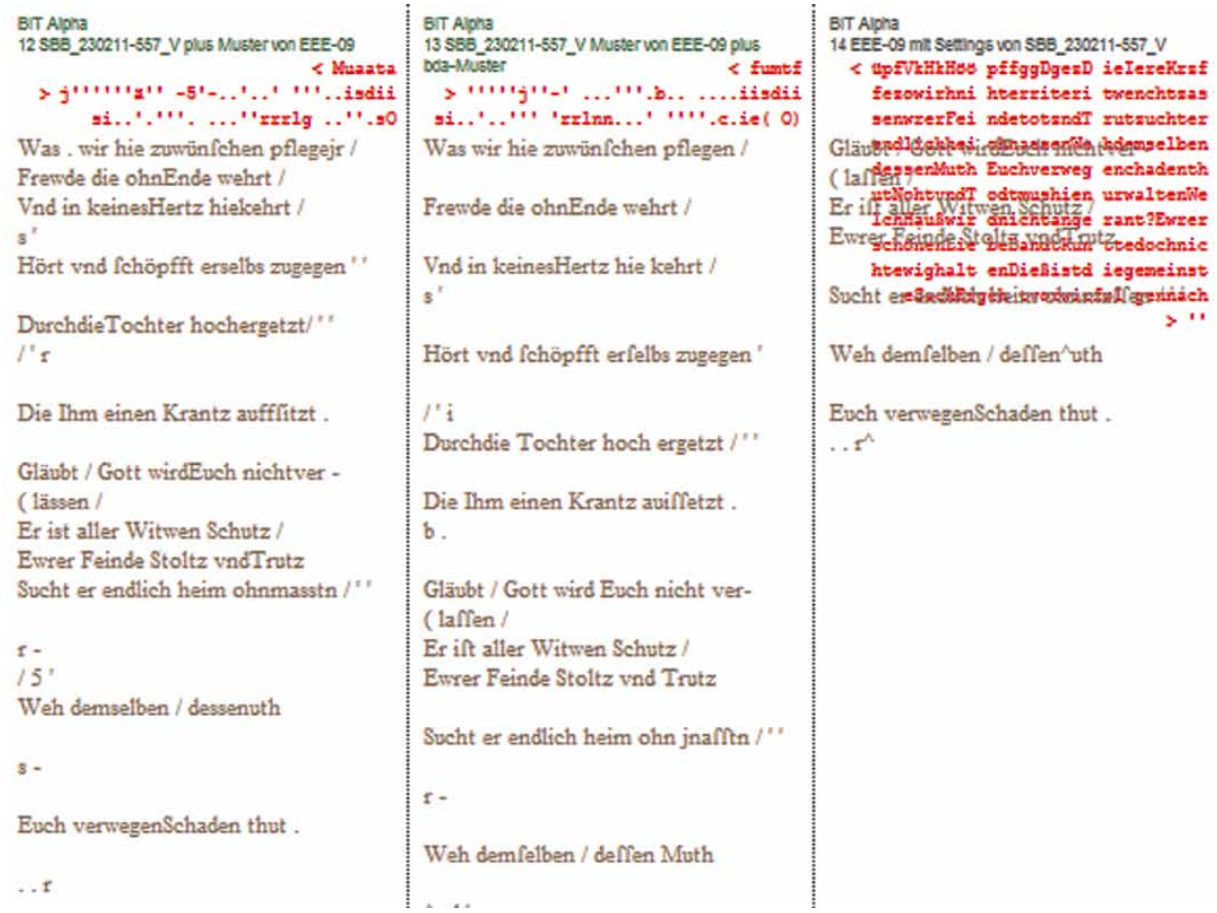
Durch vergleichende OCR-Läufe verschiedener bda-Dateien auf denselben Seiten mit gleichem Trainingsstand konnten anhand der Fehlerzählung die günstigeren bda-Konfigurationen ausgewählt werden, z. B. wurde so der zur Marge passende Binarisierungsalgorithmus gewählt (hier: „Niblack“).

Gelegentliche sprunghafte Unterschiede in der „Fehlerrate“ derselben Stichprobe konnten auch auf den Ausfall ganzer Absätze hindeuten, wenn bestimmte Segmentierungsparameter auf einzelnen Seiten einen Textblock fälschlich als Bildregion klassifizierten (genauere Hinweise hierzu im Anhang 7.3). Erkennbar war das am ehesten visuell in einer tabellarischen Textübersicht (auffällige Lücken, lange Fehlerlisten) oder durch einen maschinellen Vergleich der Textlänge mit anderen OCR-Läufen derselben Seite (beides wurde von der Software selbst leider nicht bereitgestellt; vgl. Abb. 51).

Training

Solange man der Benutzerführung im Training folgt und schwerpunktmäßig die nicht erkannten sowie die farbig hervorgehobenen besonders unsicher erkannten Zeichen trainiert, ergibt sich eine automatische Verbesserung der Erkennungsleistung mit fortschreitendem Training von selbst. Im Zweifelsfall kann per manuellem oder automatischem *refresh* der Anzeige nach jedem Trainingsschritt zumindest für den Bereich der aktuell geöffneten Seite sofort kontrolliert werden, ob mit dem Lernfortschritt für das trainierte Zeichen unerwünschte Verschlechterungen der Erkennung anderer Zeichen eingetreten sind (farbige Markierungen ändern sich; vgl. Abb 52).

[83] Auf standardisierte, extern vergleichbare statistische Maße wurde verzichtet, da sie softwaremäßig nicht unmittelbar bereitstanden und ihre aufwendige Einbeziehung in den Software-Test der Fragestellung des Projekts nicht entsprochen hätte. Der wie beschrieben gezählte Anteil „fehlender“ Zeichen kann zumindest als echte Untergrenze der Fehlerrate auf Zeichen-Ebene gelten. Andere Fehlerarten blieben ausgeklammert: z. B. erscheinen Buchstabenvertauschungen („un“ statt „nu“) in diesen Listen nicht; ebenso wurden Weißraum-Fehler wie fehlende oder falsche Spatien bewusst aus der Zählung ausgenommen – sie hätten wegen der sehr unterschiedlichen Verhältnisse im Drucksatz und entsprechenden Abgrenzungsproblemen (ab wann sollen nicht erkannte Wortzwischenräume als OCR-Fehler gelten?) einen unverhältnismäßig hohen und unberechenbar schwankenden Anteil eingenommen und die wichtigeren Differenzen in der echten Zeichenerkennungsleistung vermutlich überdeckt. Die Liste „überschüssiger“ Zeichen kann hingegen Unterschiedliches bedeuten: echte Lesefehler sind es dann, wenn sie falsche Lesarten tatsächlich vorhandener Druckzeichen darstellen; daneben erscheinen hier aber auch Artefakte von fälschlich als Zeichen interpretierten Bildelementen, Durchdruckstellen der Rückseite usw., die eine Nutzbarkeit der so entstandenen Volltexte nicht im selben Maße beeinträchtigen und daher als Fehler schwächer gewichtet werden können.



51 Auswirkung falscher Segmentierung auf Fehlerrate

52 Farbliche Darstellung unterschiedlich „sicher“ erkannter Zeichen

Spezieller Evaluierungsbedarf besteht für das in BIT-Alpha mögliche automatische Lernen, bei dem Muster, die (noch) nicht im Trainingsbestand sind, aber einen Zeichenwert mit gewisser „Wahrscheinlichkeit“ klassifizieren, den trainierten Mustern in einem Batch-Lauf (zweckmäßigerweise ohne Export) automatisch hinzugefügt werden. Ob diese Erweiterung der Musterbibliothek real zu einer besseren oder schlechteren OCR-Leistung führt, hängt von der Passung der automatisch hinzugekommenen Muster zur konkreten Vorlage ab und muss an Originalstichproben überprüft werden. Durch die Einstellung des Konfidenzintervalls („OCR > Library > Learning parameters“) kann gesteuert werden, ob das automatische Lernen die Musterbibliothek eher „vorsichtig“ oder eher „großzügig“ erweitert.

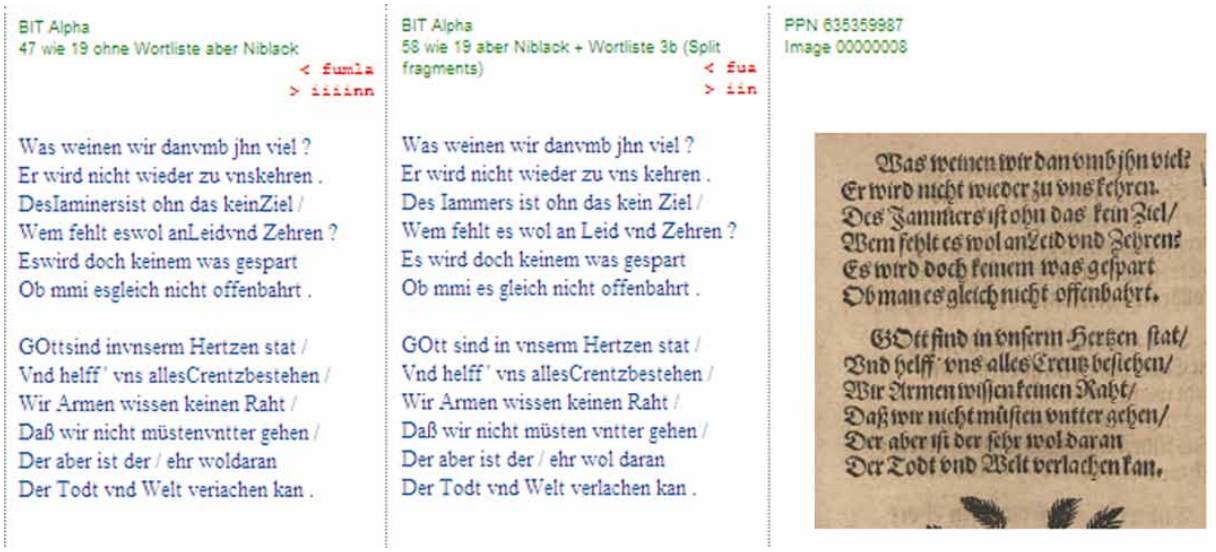
Auf diese Weise konnten einige gegenüber der nur manuell trainierten Mustermenge „besser“ erkennende bda-Dateien erstellt werden.

Wörterbuch

Nachdem die Möglichkeiten der Mustererkennung ausgeschöpft sind, kann eine sorgfältig angelegte lexikalische Korrektur noch einen merklichen Fortschritt bringen. Wieder ist das Zusammenspiel mehrerer Faktoren in BIT-Alpha sehr komplex und erfordert eine gründliche Vorbereitung mit mehrfachem Vergleichstest anhand einer Originalstichprobe. Aufeinander abgestimmt werden müssen:

- eine gut zur Lexik passende Wortliste, die einerseits möglichst vollständig inklusive aller flektierten Formen und vorkommenden Schreibvarianten sein, andererseits möglichst wenig irrelevanten Wortschatz enthalten soll. Es kann versucht werden, einmalige bzw. sehr selten vorkommende Wortformen zu sichten und ggf. auszuschließen. Ob getrennte Wortteile einzeln eingelesen werden können oder zusammengesetzt werden müssen, ist abhängig von den Trennungseinstellungen in der Konfiguration der Lexikalischen Korrektur zu entscheiden. Jedes Vorgehen hat Vor- und Nachteile.
- die Distanz, bis zu der eine Ersetzung überhaupt stattfinden soll (Entscheidung, ob generell eher viel oder wenig ersetzt werden soll);
- die einzelnen Ersetzungskoeffizienten (Erleichterung oder Erschwernis der konkreten Zeichenpaare, die ausgetauscht werden dürfen).

Hier liegt ein hohes Potential an lexikalischer und linguistischer Adaption an das Textmaterial. Die entstehenden Konfigurationen (in lx2-Dateien) können gespeichert werden und wären dann für Material der jeweiligen Sprache und Epoche wiederverwendbar.



53 Einfluss des Lexikons. Rot über dem Text die Listen der fehlenden (<) und überzähligen (>) Zeichen

Der Aufwand ist allerdings hoch. Im Testprojekt konnten die Möglichkeiten nicht vollständig ausgeschöpft werden; bewährt hat sich die Erstellung von Wörterbüchern aus vergleichbarer Literatur (hier besonders: Personalschriften, Gelegenheitsgedichte und biographisches Material des 17. Jahrhunderts) sowie die Einpflege spezieller Thesauri (hier: zeitspezifische Orts-, Berufs- und Krankheitsbezeichnungen) und bereits vorliegender Katalogdaten (Titel- und Personenfelder) der zu lesenden Schriften selbst, wodurch auch sehr spezifische passende Wortformen ins Wörterbuch gelangten. Die Pflege und Nachbesserung des Wortbestands wird stark behindert durch das binäre, nicht frei editierbare Format der lx2-Dateien; dringend wäre eine freie Bearbeitung der Wortliste in einem offenen Textformat erforderlich, in dem jederzeit Such- und Ersetzungsoperationen, Umsortierungen, Löschung und Austausch größerer Blöcke usw. möglich sind.

Zur Beurteilung des lexikalischen Korrekturschritts sollte neben der Fehleranzahl auch die Art der Fehler betrachtet werden. Wenn etwa eine ursprünglich optisch nahe liegende Verwechslung unter Wörterbucheinfluss zu einem optisch unähnlichen neuen Fehler ersetzt wird, dann wäre trotz gleich gebliebener

Fehlerquote die ursprüngliche Version besser nutzbar als die neue. Insofern können quantitative Maße die Fehler lexikalischer Korrektur nur zum Teil abbilden und gelegentlich sogar in der Vergleichsaussage falsch liegen. Fällt eine falsche Korrektur gar in einen plausiblen Kontext (gesehen z. B.: Vorlage „bei sich" > OCR „bei stch" > Korrektur „bei steh"), dann wird auch dem menschlichen Leser, der die fehlerbelassene Version wohl im Sinne der Vorlage richtig interpretiert hätte, eine falsche Lesart aufgedrängt.

(b) HK-OCR

Gruppierung

Eine erste, auf einer sehr homogenen Schriftgruppe trainierte Mustermenge erkannte die speziell zu dieser Schriftgröße passenden Seiten sehr gut und bereits Seiten des nächst niedrigeren Schriftgrads deutlich schlechter. Die Muster in den ptn-Dateien dürfen demnach als hoch selektiv gelten, was in passenden Einsatzszenarien ein Vorteil sein kann.

Eine Anlage separater später kombinierbarer Musterbibliotheken für bestimmte Schrifttypen und -größen läge daher für HK-OCR bzw. FineReader besonders nahe. Wegen der fehlenden Möglichkeit zum Mischen verschiedener Musterbestände aber erfordert die Mehrfachverwendung eine sorgfältige Planung bzw. erhöht den Aufwand. Ohne weiteres lässt sich z. B. ein Musterbestand für Schrift X trainieren, zunächst in einer X.ptn-Datei speichern, und anschließend könnte für den Einsatz an gemischten Vorlagen der Schriftgruppen X+Y in einer Kopie „XY.ptn" dieser Musterdatei das Training mit Schriftart Y fortgesetzt werden, um eine kombinierte Bibliothek aus beidem zu erhalten. Eine reine Y-Mustersammlung ist daraus dann aber nicht extrahierbar, wie umgekehrt ein ggf. rein vorliegender Y-Musterbestand nicht der X-Bibliothek hinzugefügt werden kann. Wenn Muster in verschiedenen Mischungen benötigt werden, kann durch diese Beschränkung ein mehrfaches Training bestimmter Vorlagen nötig werden.

Leicht möglich ist andererseits die Kombination der gewählten Anwender-Muster mit den eingebauten FineReader-Mustern, so dass es sich stets lohnt, die drei Alternativen (a) Anwender-Muster, (b) FineReader-Muster, (c) beides zusammen an Originalstichproben miteinander zu vergleichen.

Binarisierungsparameter, Training, Wörterbuch

Wegen der in HK-OCR überschaubaren Konfigurationsmöglichkeiten konnten einige Parametervariationen auf verschiedenen Trainingsstufen systematisch paarweise unter Konstanthaltung der anderen Bedingungen verglichen werden.

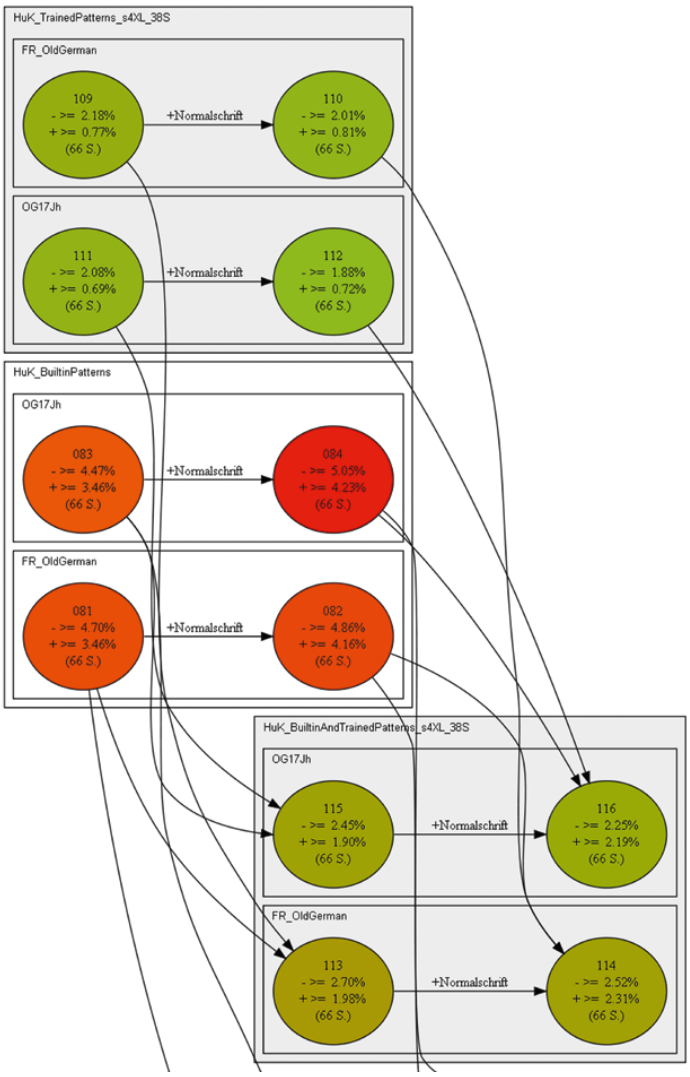
– *Muster (‚Lohnt sich Training und/oder die Kombination von eingebauten mit trainierten Mustern?')*

Für die Laborsituation der homogenen ersten Marge konnte sehr schnell die Wirksamkeit des Trainings nachgewiesen werden.

Bereits nach einem Training von 37 Seiten aus Gelegenheitsgedichten Simon Dachs war die Erkennung der Stichprobe mit den selbst trainierten Mustern wesentlich besser als die Erkennung mit den FineReader-Mustern allein (die Anzahl fehlender Zeichen fiel auf weniger als die Hälfte, die Anzahl überschüssiger Zeichen ca. auf ein Fünftel)[84] und auch merklich besser als die Erkennung bei Kombination der selbst trainierten mit den FineReader-Mustern (der Anteil fehlender Zeichen sank um ein Fünftel, die Anzahl überschüssiger auf unter die Hälfte). Dieser Effekt war unabhängig davon, ob die FineReader-Sprache „OldGerman" oder das Anwenderwörterbuch, und ob in den OCR-Optionen allein Fraktur oder die Kombination von Fraktur und Antiqua aktiviert war (vgl. Abb. 54 u. 55).

Für heterogene Muster ist der Fortschritt erwartungsgemäß langsamer, hier wurde nach Training von sechzig Seiten Titelblättern ein Zustand erreicht, in dem der Anteil fehlender Zeichen auch für die OCR der obigen homogenen Gedicht-Seiten bei Verwendung nur selbst trainierter Muster fast an die FineReader-Rate herankam und der Anteil überschüssiger Zeichen bereits deutlich geringer war als bei den FineReader-Mustern. Schon weit vorher, ab ca. zehn Seiten Training, war die Erkennung mit den selbst trainierten Mustern zwar noch deutlich schlechter als die FineReader-Rate, aber bereits hier brachte die

[84] Auch wenn die Fehlerzählung nicht für sich als Gütemaß gelten kann, dürfen die Unterschiede der so gezählten Fehlerfälle wohl als Hinweis auf analoge Bewegungen anderer Fehlermaße interpretiert werden.



54 Graphischer Vergleich aufeinander folgender OCR-Läufe mit verschiedenen Musterdateien

Hinzuwahl der selbst trainierten Muster zu den FineReader-Mustern eine stabile Verbesserung.

Für spätere, im 18. Jahrhundert erschienene Schriften des SBB-Funeralschriftenbestands ändert sich das Bild. Erste Tests deuten an, dass für das späte 17. und beginnende 18. Jahrhundert eine Kombination aus den trainierten und den FineReader-Mustern häufig das bessere Ergebnis erzielt und dass im späteren 18. Jahrhundert häufig die FineReader-Muster auch allein zumindest besser abschneiden können als die trainierten Muster des 17. Jahrhunderts aus dem Projekt. Da speziell zu den Schriften des 18. Jahrhunderts noch keine Proben trainiert wurden, konnte kein echter Vergleich mit solchen stattfinden. Eingehendere Tests wären sicher nicht aussichtslos. Solange allerdings die additive Verwendung speziell trainierter Musterbestände ungeklärt ist, wird der Aufwand ihrer Erstellung wegen der eingeschränkten Nachnutzbarkeit selten vertretbar sein.

Muster
'Lohnt sich Training und/oder die Kombination von eingebauten mit trainierten Mustern?'

Builtin-Patterns		s4XL-Trained Patte		s4XL-Trained + Bui		TitlBl-Trained Patte		TitlBl-Trained + Builtin	
081 HuK OG	4,70	109 HuK OG	2,18	113 HuK OG	2,70	085 HuK OG	12,35	089 HuK OG	5,40
082 HuK OG	4,86	110 HuK OG	2,01	114 HuK OG	2,52	086 HuK OG	14,69	090 HuK OG	5,37
083 HuK OG	4,47	111 HuK OG	2,08	115 HuK OG	2,45	087 HuK OG	12,10	091 HuK OG	4,53
084 HuK OG	5,05	112 HuK OG	1,88	116 HuK OG	2,25	088 HuK OG	14,45	092 HuK OG	4,39
						093 HuK OG	7,19	097 HuK OG	4,24
						094 HuK OG	7,06	098 HuK OG	4,13
						095 HuK OG	7,04	099 HuK OG	4,05
						096 HuK OG	6,95	100 HuK OG	3,91
						117 HuK OG	4,14	145 HuK OG	3,71
						118 HuK OG	4,06	146 HuK OG	3,55
						119 HuK OG	4,14	147 HuK OG	3,55
						120 HuK OG	4,02	148 HuK OG	3,31

55 Vergleich aufeinander bezogener OCR-Läufe mit verschiedenen Musterdateien; die Zahlen zeigen die durchschnittliche Rate fehlender Zeichen pro Seite an

– *Nur-Fraktur- vs. Fraktur-plus-Antiqua-Training (‚Lohnt es sich, in Fraktur-Musterdateien Antiqua-Muster mitzutrainieren?‘)*

Bei der Gelegenheit des Trainings von Titelblättern mit ihrer Vielfalt an Schrifttypen wurde getestet, ob ein Mittrainieren der enthaltenen Antiqua-Zeichen die Erkennung reiner Frakturseiten beeinträchtigt. Wenn ja, würde man darauf zumindest in einer Musterbibliothek verzichten; wenn nein, könnte von Anfang an eine Reserve von Antiqua-Zeichen unbedenklich mitgeführt werden, unabhängig davon, wie viel Antiqua-Einsprengsel der Text tatsächlich enthält.

Wider Erwarten zeigte sich auch auf den Nur-Fraktur-Seiten stets eine leichte Verbesserung der Frakturerkennung, wenn ein gewisser Anteil (im Versuch: ca. fünf Prozent) Antiqua-Zeichen mittrainiert wurde. Eine Erklärung fällt schwer; immerhin bleibt die pragmatische Aussage, hier nicht streng separieren zu müssen (vgl. Abb. 56).

Nur-Fraktur- vs. Fraktur-plus-Antiqua-Training

'Lohnt es sich, in Fraktur-Musterdateien Antiqua-Muster mitzutrainieren?'

Nur Fraktur		Fraktur + Antiqua	
093 HuK OG	7,19	101 HuK OG	7,13
094 HuK OG	7,06	102 HuK OG	7,05
095 HuK OG	7,04	103 HuK OG	6,96
096 HuK OG	6,95	104 HuK OG	6,92
097 HuK OG	4,24	105 HuK OG	4,21
098 HuK OG	4,13	106 HuK OG	4,09
099 HuK OG	4,05	107 HuK OG	4,02
100 HuK OG	3,91	108 HuK OG	3,82

56 Zusätzlich trainierte Antiquamuster und durchschnittliche Rate fehlender Zeichen

– *Schriftfamilien-Checkboxen (‚Lohnt sich die Zuwahl von (eingebauter) Antiqua-Erkennung?‘)*

Ob die in der HK-OCR-Konfiguration mögliche Zuwahl von „Normalschrift“ einen zusätzlichen Musterbestand in die Erkennung einbezieht oder andere Parameter ändert, war nicht dokumentiert. Der Vergleich der Ergebnisse zwischen Nur-Fraktur- und Fraktur-plus-Antiqua-Läufen ergab, dass nur für die FineReader-Builtin-Patterns allein und für noch fast untrainierte eigene Muster durch Zuwahl der Antiqua eine Verschlechterung eintrat, in allen Fällen fortgeschrittenen eigenen Trainings und der Kombination selbst trainierter mit den FineReader-Mustern dagegen eine Verbesserung der Erkennung auch in reinen Fraktur-Seiten. Ohne nähere Kenntnis der Funktion der Normalschrift-Auswahl fällt eine Interpretation dieses Befunds schwer. Zu vermerken ist die Empfehlung, auch für reinen Frakturtext die Normalschrift-Checkbox möglicherweise nutzen zu können (vgl. Abb. 57).

Schriftfamilien-Checkboxen

'Lohnt sich die Zuwahl von (eingebauter) Antiqua-Erkennung?'

Fraktur		Fraktur + "Normal"	
081 HuK OG	4,70	082 HuK OG	4,86
083 HuK OG	4,47	084 HuK OG	5,05
085 HuK OG	12,35	086 HuK OG	14,69
087 HuK OG	12,10	088 HuK OG	14,45
089 HuK OG	5,40	090 HuK OG	5,37
091 HuK OG	4,53	092 HuK OG	4,39
093 HuK OG	7,19	094 HuK OG	7,06
095 HuK OG	7,04	096 HuK OG	6,95
097 HuK OG	4,24	098 HuK OG	4,13
099 HuK OG	4,05	100 HuK OG	3,91
109 HuK OG	2,18	110 HuK OG	2,01
111 HuK OG	2,08	**112 HuK OG**	**1,88**
113 HuK OG	2,70	114 HuK OG	2,52
115 HuK OG	2,45	116 HuK OG	2,25
117 HuK OG	4,14	118 HuK OG	4,06
119 HuK OG	4,14	120 HuK OG	4,02
145 HuK OG	3,71	146 HuK OG	3,55
147 HuK OG	3,55	148 HuK OG	3,31

57 Zuwahl mitgelieferter Antiquamuster u. durchschnittliche Rate fehlender Zeichen

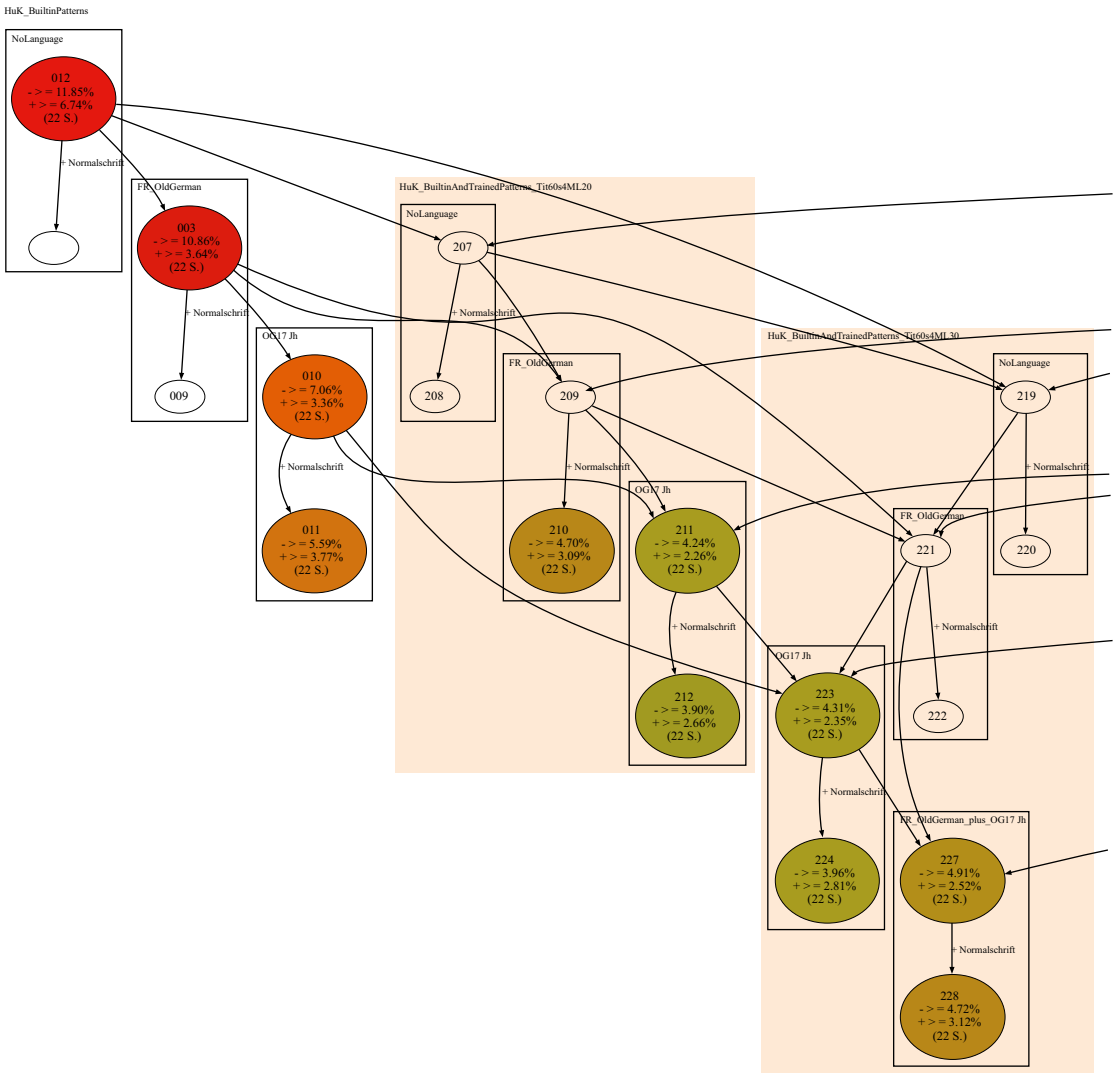
– *Lexikon (‚Lohnt sich die eigene Wortliste?‘):*

In allen ausgewerteten Situationen der ersten Schriftgruppe war eine deutliche Überlegenheit des Einsatzes von altdeutschen Wortlisten gegenüber modernem Wortschatz generell, und fast immer eine leichte Überlegenheit des auf die Funeralschriften angepasst erstellten Lexikons gegenüber dem FineReader-OldGerman festzustellen – obwohl das im Projekt verwendete Wörterbuch bei weitem noch nicht zu Ende optimiert worden war.

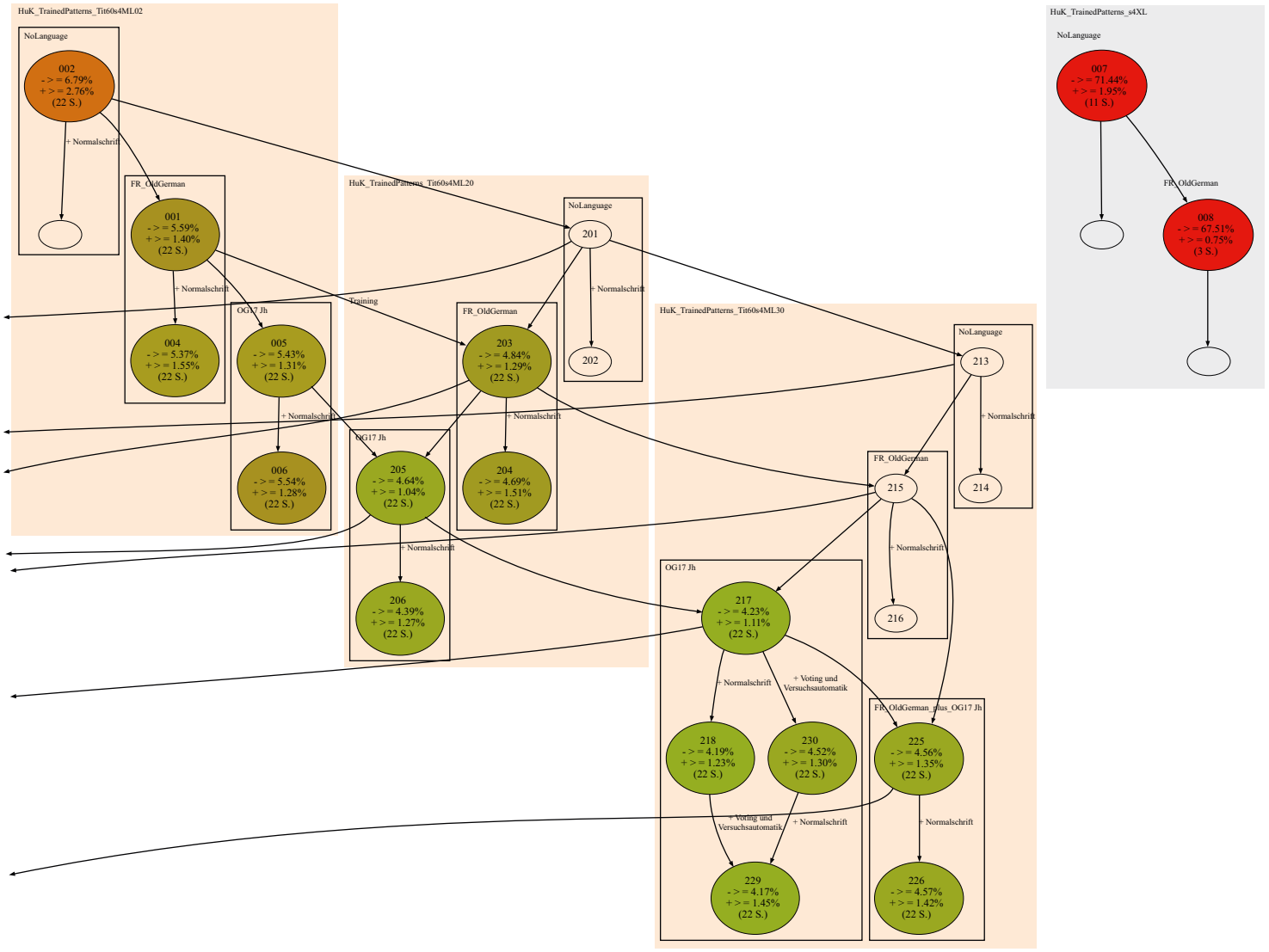
Für die zweite Schriftgruppe (die zugleich aus über Lyrik hinausgehendem Material bestand) fiel

der Unterschied noch deutlicher aus. Hier wurde zusätzlich zum Vergleich der FineReader-Sprache „OldGerman“ mit der Anwender-Sprachengruppe auch die in HK-OCR mögliche Kombination beider einbezogen. Sie erreichte die Werte der Anwenderwortbibliothek ebenfalls nicht und blieb meist auf derselben Erkennungsgüte wie die FineReader-Sprache allein. Zu interpretieren ist das wohl so, dass im Fall der Kombination die FineReader-Wortliste klar dominiert: zum Vorteil der Erkennung, wenn sie die passenderen Einträge enthält, sonst zu ihrem Nachteil.

Sowohl die in die Sprach-Definitionen von FineReader eingehenden Wortlisten als auch die Anwenderlexika haben einen starken Einfluss auf die OCR-Leseergebnisse, ohne dass der Anwender hier viel steuern (insbesondere: dämpfen) kann.[85] Auch hier lohnt es sich also, viel Sorgfalt auf die Erstellung des Wörterbuchs zu verwenden, selbst wenn HK-OCR bzw. FineReader bisher keine Feinsteuerung der Ersetzungen erlaubt.



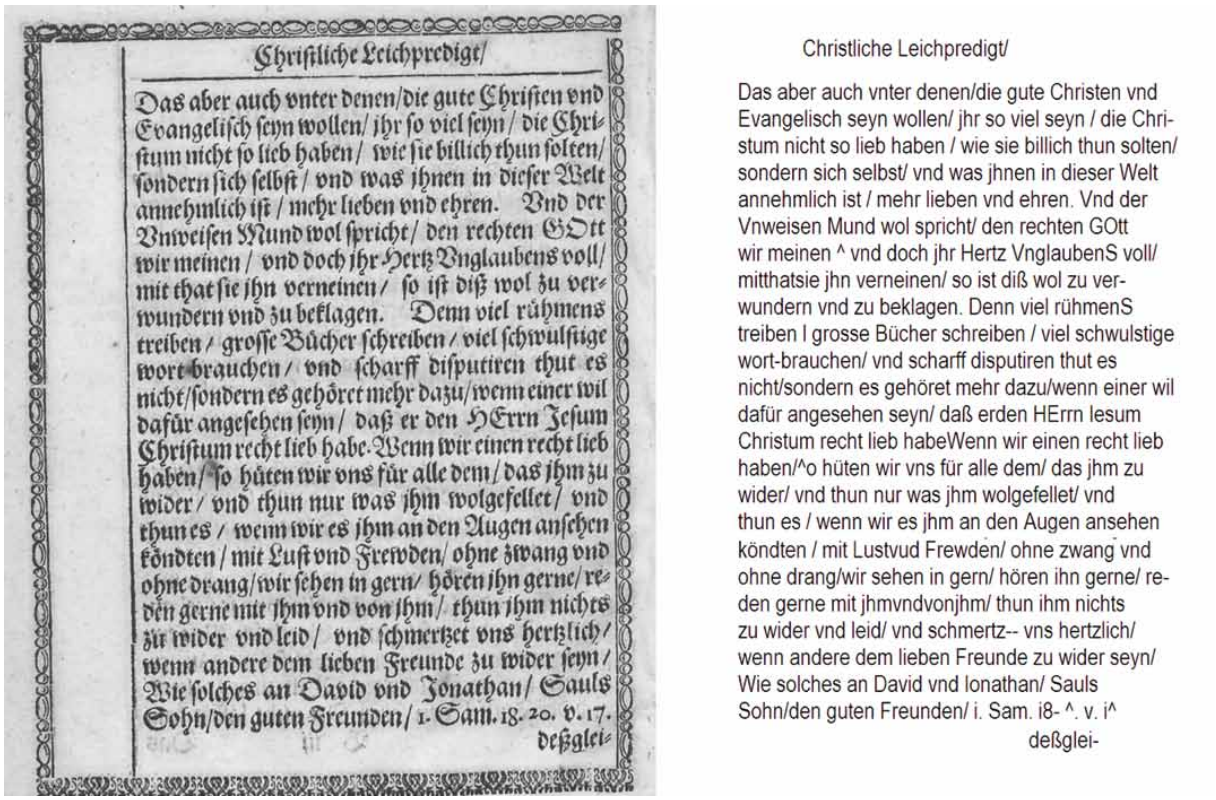
[85] Gesehen wurde z. B. eine offenbar lexikongetriebene Hyperkorrektur von zunächst (OCR mit FineReader-Mustern ohne gewählter Sprache:) „dieNen“ zu (FineReader-Muster mit gewählter Sprache:) „vierten“, die nach eigenem Mustertraining bei Verwendung desselben Lexikons richtig als „dienen“ gelesen wurde.



58 Vergleiche in Schriftgruppe 2: Trainingsfortschritt, Musterwahl, Wortbestandswahl und -kombination usw.

6 Zusammenfassung der Ergebnisse
6.1 Einflussfaktoren auf die Erkennungsgüte
6.1.1 Nachweis des Einflusses von Vorsortierung

Für beide getesteten Softwareprodukte gilt: Seiten, deren Typenvorrat einer passenden trainierten Musterbibliothek zugeordnet werden kann, können auch schon nach mäßigem Training und bei durchschnittlichen sonstigen Randbedingungen eine hohe Erkennungsgüte erreichen:



Christliche Leichpredigt/

Das aber auch vnter denen/die gute Christen vnd Evangelisch seyn wollen/ jhr so viel seyn / die Christum nicht so lieb haben / wie sie billich thun solten/ sondern sich selbst/ vnd was jhnen in dieser Welt annehmlich ist / mehr lieben vnd ehren. Vnd der Vnweisen Mund wol spricht/ den rechten GOtt wir meinen ^ vnd doch jhr Hertz VnglaubenS voll/ mitthatsie jhn verneinen/ so ist diß wol zu verwundern vnd zu beklagen. Denn viel rühmenS treiben I grosse Bücher schreiben / viel schwulstige wort-brauchen/ vnd scharff disputiren thut es nicht/sondern es gehöret mehr dazu/wenn einer wil dafür angesehen seyn/ daß erden HErrn Iesum Christum recht lieb habeWenn wir einen recht lieb haben/^o hüten wir vns für alle dem/ das jhm zu wider/ vnd thun nur was jhm wolgefellet/ vnd thun es / wenn wir es jhm an den Augen ansehen köndten / mit Lustvud Frewden/ ohne zwang vnd ohne drang/wir sehen in gern/ hören ihn gerne/ reden gerne mit jhmvndvonjhm/ thun ihm nichts zu wider vnd leid/ vnd schmertz-- vns hertzlich/ wenn andere dem lieben Freunde zu wider seyn/ Wie solches an David vnd Ionathan/ Sauls Sohn/den guten Freunden/ i. Sam. i8- ^. v. i^

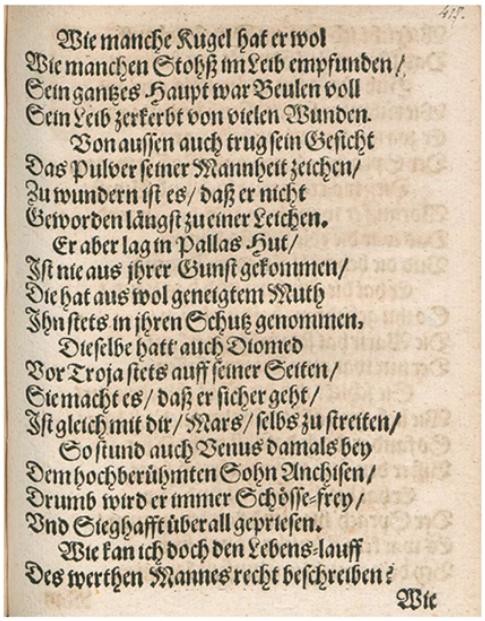
deßglei-

59 Gute Zuordnung von Drucktypen zu passender Musterbibliothek - bearbeitet mit HK-OCR

Durch eine entsprechende Gruppierung des Ausgangsmaterials können Stapelverarbeitungen aussichtsreich vorbereitet werden. Ohne Vorsortierung kann diese Erkennungsrate bei weitem nicht erwartet werden, und entsteht die Notwendigkeit, die Spreu erst nach der OCR vom Weizen zu trennen. Dies trifft auch für große Teile der im Projekt verarbeiteten Bände zu, eben weil die Vorsortierung nur in Versuchsmargen (insgesamt ca. 5.000 Seiten) konsequent genug war.

6.1.2 Nachweis des Einflusses von Binarisierungsparametern

Dass Binarisierungs- und Segmentierungsparameter entscheidend für die Möglichkeit einer sinnvollen Musteridentifizierung sind, ist trivial. Zwar ist bemerkenswert, mit wie wenig nutzerseitiger Parametrierung die FineReader-Engine eine *im Durchschnitt* optimale Zeichensegmentierung erreicht, dennoch gehen hier Möglichkeiten zur Nachjustierung verloren. Zu den im Gegenbeispiel BIT-Alpha reichhaltig verfügbaren Optionen und Parametern wird im Anhang versucht, eine vorsichtige Orientierung über sinnvolle Einstellungen und Wertebereiche zu geben (s. Anh. 7.3); verbindliche Information wäre allerdings vom Hersteller anzufordern. An folgendem Beispiel kann verfolgt werden, wie sich die Anzahl fehlender richtiger Zeichen unter dem Gebrauch verschiedener Binarisierungsparameter verändert.

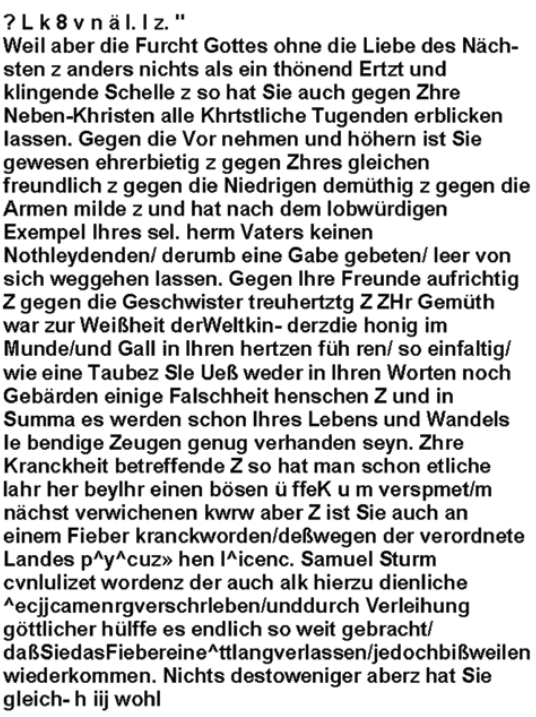
Beispiel B.I.T. Alpha – halbtrainiert, ohne Wörterbuch, Binarisierungsparameter I	Leichenpredigt 1649	Beispiel B.I.T. Alpha – halbtrainiert, ohne Wörterbuch, Binarisierungsparameter II
BIT Alpha17 SBB_230211-557_V bereinigt Muster von EEE-09 plus bda-Muster Fehlende Zeichen: 8 mmSoltsr Überflüssige Zeichen: 11 'ffnnnieni e Wie manche Kugel hat erwoi Wie manchen Stohß iniLeib empfunden / Sein gantzes Haupt war Beulen voll Sein Leib zerkerbt von vielen Wunden . Vonauſſen auch trug ſein Geſicht n^ Das Pulver ſeiner Mannheit zeichen / Zuwundern iſt es / daß er nicht Geworden längſt zu einer Leichen . Er aber lag in Pallas Hut / Iſt nieaus jhrer Gunſt gekennmen / Die hat auswol geneigtem Muth Ihn ſtets in jhren Schutz genommen . Dieſelbe hatt ' auch Diomed VorTroja ſtets aufffeiner Seiten / ^iemacht es / daß er ſichergeht / Iſtgleich mit dir / Mars / ſelbszu ffeeiten/ So ſtund auchVenus damalsbey Dem hochberühmten Sohn Anchiſen / Drumb wird er immer Schöſſe - frey / Vnd Sieghafft überall geprieſen . Wie kan ich doch den Lebens - lauff Des werthen Mannesrecht beſchreiben ? Wie	 Wie manche Kugel hat er wol Wie manchen Stohß im Leib empfunden / Sein gantzes Haupt war Beulen voll Sein Leib zerkerbt von vielen Wunden. Von auſſen auch trug ſein Geſicht Das Pulver ſeiner Mannheit zeichen/ Zu wundern iſt es/ daß er nicht Geworden längſt zu einer Leichen. Er aber lag in Pallas Hut/ Iſt nie aus jhrer Gunſt gekommen/ Die hat aus wol geneigtem Muth Ihn ſtets in jhren Schutz genommen. Dieſelbe hatt' auch Diomed Vor Troja ſtets auff ſeiner Seiten/ Sie macht es/ daß er ſicher geht/ Iſt gleich mit dir/ Mars/ ſelbs zu ſtreiten/ So ſtund auch Venus damals bey Dem hochberühmten Sohn Anchiſen/ Drumb wird er immer Schöſſe-frey/ Vnd Sieghafft überall geprieſen. Wie kan ich doch den Lebens-lauff Des werthen Mannes recht beſchreiben? Wie	BIT Alpha59 wie 27 ohne Wortliste aber Niblack Fehlende Zeichen: 3 ene Überflüssige Zeichen: 10 '..oliccii Wie manche Kugel hat erwol Wie manchen Stohß imLeib empfuliden / Sein gantzes Haupt warBeulen voll Sein Leib zerkerbt von vielen Wunden . Vonauſſen auch trug sein Gesicht n^ Das Pulver seiner Mannheit zeichen / Zuwundern ist es / daß er nicht Geworden längst zu einer Leichen . Er aber lag in Pallas Hut / Ist nieaus jhrer Gunst gekommen / Die hat aus wol geneigtem Muth Ihn stets in jhren Schutz genommen . Diesclbe hatt ' auch Diomed VorTroja stets auffseiner Seiten / Siemacht es / daß er sichergeht / o . . Istgleich mit dir / Mars / sclbszu streiten / So stund auchVenus damalsbey Dem hochberühmten Sohn Anchisen / Drumb wird er immer Schöſſe - frey / Vnd Sieghafft überall gepriesen . Wie kan ich doch den Lebens - lauff Des werthen Maiinesrecht

60 Texterkennung nach verschiedenen Binarisierungsverfahren in BIT-Alpha

6.1.3 Nachweis des Trainingseffekts

BIT-Alpha, das ohne Muster ausgeliefert wird, beruht auf der Wirksamkeit des Trainings passender Muster, so dass der hier während der Versuche mit zunehmendem Training beobachtete Erkennungsfortschritt – bzw. die Differenzen bei Verwendung verschiedener Musterbibliotheken – selbstverständlich ist.

Interessanter war die Feststellung, wie sehr sich ein materialbezogenes Training auch bei der mit guter Frakturmuster-Grundversorgung ausgestatteten FineReader-Engine auswirkte. Das zeigen nicht nur die im Text beschriebenen Fehlerzählungen auf den vorliegenden Ground-Truth-Stichproben, sondern auch die in den Layout-Protokollen der FineReader-Engine vermerkten Erkennungsraten (offenbar Anteile der im Wörterbuch enthaltenen Wörter an den insgesamt erkannten). In den Bänden des 17. Jahrhunderts, die sowohl mit Finereader-Mustern als auch mit selbst trainierten Mustern vergleichend prozessiert wurden, wurden für die eigene Musterdatei – trotz der begrenzten Trainingsintensität von 90 Seiten – oft als um zwei bis fünf Prozentpunkte höhere Raten vermerkt als im Lauf mit den FineReader-Frakturmustern.



? L k 8 v n ä l. l z. "
Weil aber die Furcht Gottes ohne die Liebe des Näch-
sten z anders nichts als ein thönend Ertzt und
klingende Schelle z so hat Sie auch gegen Zhre
Neben-Khristen alle Khrtstliche Tugenden erblicken
lassen. Gegen die Vor nehmen und höhern ist Sie
gewesen ehrerbietig z gegen Zhres gleichen
freundlich z gegen die Niedrigen demüthig z gegen die
Armen milde z und hat nach dem lobwürdigen
Exempel Ihres sel. herm Vaters keinen
Nothleydenden/ derumb eine Gabe gebeten/ leer von
sich weggehen lassen. Gegen Ihre Freunde aufrichtig
Z gegen die Geschwister treuhertztg Z ZHr Gemüth
war zur Weißheit derWeltkin- derzdie honig im
Munde/und Gall in Ihren hertzen füh ren/ so einfaltig/
wie eine Taubez Sle Ueß weder in Ihren Worten noch
Gebärden einige Falschheit henschen Z und in
Summa es werden schon Ihres Lebens und Wandels
le bendige Zeugen genug verhanden seyn. Zhre
Kranckheit betreffende Z so hat man schon etliche
lahr her beylhr einen bösen ü ffeK u m verspmet/m
nächst verwichenen kwrw aber Z ist Sie auch an
einem Fieber kranckworden/deßwegen der verordnete
Landes p^y^cuz» hen l^icenc. Samuel Sturm
cvnlulizet wordenz der auch alk hierzu dienliche
^ecjjcamenrgverschrleben/unddurch Verleihung
göttlicher hülffe es endlich so weit gebracht/
daßSiedasFiebereine^ttlangverlassen/jedochbißweilen
wiederkommen. Nichts destoweniger aberz hat Sie
gleich- h iij wohl

Gattungsspezifische Wortliste + FineReader-Muster

PERSONALIA. ^
Weil aber die Furcht Gottes ohne die Liebe desNäch^
sten /^ anders nichts als ein thönend Ertzt und
klingende S chelle z so hat Sie auch gegen Ihre Neben-
Christen alle Christliche Tugeuden erblicken lassen.
Gegen die Vor nehmen und Höhern ist Sie gewesen
ehrerbietig /^ gegen Ihres gleichen freundlich /^ gegen
die Niedrigen demüthig ^ gegen die Armen milde /^
und hat nach dem lobwürdigen Exempel Ihres sel.
Herrn Vaters keinen Nothleydenden/ derumb eine Gabe
gebeten/ leer von steh weggehen lassen., Gegen Ihre
Freunde aufrichtig /^ gegen die Gefchwister
treuhertzig/ Ihr Gemüth war zur Weißheit derWellkm-
der/die Honig im Munde/und Gall in Ihren Hertzen füh-.
ren/ fo einfaltig/ wie eine Taube/ Sie ließ weder in Ihren
Worten noch Gebärden einige Falschheit herrschen /^
und in Summa es werden schon Ihres Lebens und
Wandels le bendige Zeugen genug verhanden seyn.
Ihre Kranckheit betreffende /^ so hat man schon etliche
lahr her beylhr einen bösen A ssect u m verspüret/in
nächst verwichenen Martio aber/ ist Sie auch au einem
Fieber kranckworden/deßwegenderverordnere Landes
Physicus, Herr Licent. Samuel Sturm consuliret
worden/ der auch al^e hierzu dienliche
^edica^eneaverschrieben/ unddurch Verleihung
göttlicher Hülffe es endlich so weit gebracht/
daßSiedasFiebereinezeitlangverlassen/jedochbißweilen
wiederkommen. Nichts destoweniger aber/ hat Sie
gleich^ H ii^ wohi

Gattungsspezifische Wortliste + anwendertrainierte Muster

61 Fortschritt in OCR-Erkennung (hier mit HK-OCR) durch Training

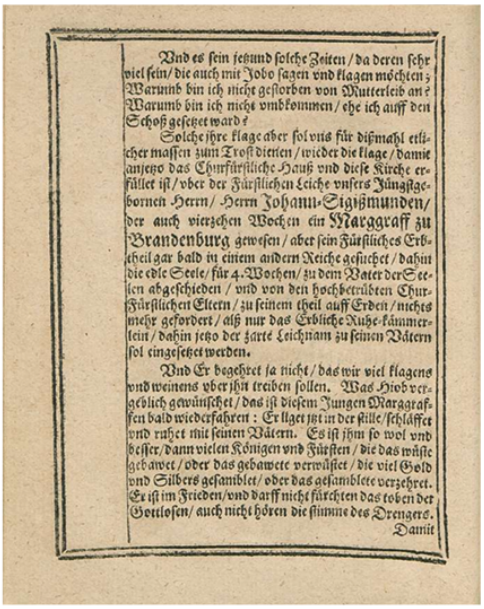
6.1.4 Nachweis des Wörterbucheffekts

Durch gezielte Gegenüberstellung von OCR-Ergebnissen, die mit bzw. ohne Hinzunahme von Wortlisten entstanden, lässt sich bei beiden Softwareprodukten zeigen, dass sich der Einsatz von Wortbibliotheken positiv auf die Ergebnisse auswirkt. Dieser Effekt beruht in BIT-Alpha auf einer vom Anwender detailliert steuerbaren lexikalischen Korrektur nach der OCR. Bei HK-OCR dagegen beeinflusst die Wortbibliothek durch Definition so genannter „Sprachen“ und „Sprachengruppen“ der FineReader-Engine offenbar schon die Texterkennung. Der maßgebliche Einfluss der Wortliste zeigt sich vor allem in der Erkennung der Wortgrenzen. Vergleicht man das Beispiel in den verschiedenen Bearbeitungsstufen wird vor allem durch die korrigierten Wortgrenzen der Text flüssiger lesbar. Es fällt ebenso auf, dass durch Einsatz der Wortliste[86] neue Fehler entstehen können: „dieNen“ wird fälschlich zu „vierten“.

Die im Projekt erstellte gattungsspezifische Wortliste bestand in der OCR-Optimierungsphase aus ca. 70.000 Einträgen. Wie eingangs beschrieben, enthält sie vor allem Wortschatz aus Leichenpredigten. Dieser Wortschatz entstand durch Transkription von Referenzseiten und speist sich zusätzlich aus Titelmaterial aller über den GBV nachgewiesenen Leichenpredigten. Zudem ergänzt um Gedichte von Simon Dach und einige wenige zeitgenössische Texte wurden vor allem die Thesauri der Forschungsstelle für Personalschriften Marburg THEPRO und THELO sowie eine Liste historischer Krankheitsbezeichnungen des Vereins für Computergenealogie[87] herangezogen. Eine weitere Optimierung in der Testphase selbst konnte nicht vorgenommen werden.

Wörterbücher können deutlich genauer erstellt werden, wenn das Ausgangsmaterial regional, zeitlich und inhaltlich auf den Verwendungszweck abgestimmt im Zuge einer fehlerfreien Erfassung entstanden ist. Ein weiterer Weg erschließt sich über die Hinzunahme aller möglichen Flexionsformen bzw. Rechtschreibvarianten der vorkommenden Wörter (s. Kap. 3.1.5 und 4.3.4).

[86] Die Ersetzung von Wörtern über das Wörterbuch in HK-OCR ist nicht steuerbar.
[87] VEREIN FÜR COMPUTERGENEALOGIE

Beispiel HK-OCR/FR9 – ohne Training, ohne „Sprache“	Leichenpredigt 1625	Beispiel HK-OCR/FR9 – ohne Training mit selbst def.„Sprache“
HKOCR012 NoLang F Builtins Fehlende Zeichen: 108 ö?ß?üßüüü Lülßüüüüßä Läjjül-jzä äjönnnüene üeeweeewee ewüeeelnle eleeebleee zererFreen nrnürendbe nderlenhnh hörendeede DrenerDa Überflüssige Zeichen: 108 !JOTN!ZNNc MccccN,ccu cukccukcck cWWHcucuMc ffccZffcuM VkvfccvZvv Svmtmkvttm ZvvffctMkc tcskttsftu cctSistimm sSsZ)mik Vndes fein feftundsolcheZsiten/daderen schr vie! fein/ die auch mit Jobs sagen vnd klagm mochten ; Warumb bin ich nichkgsstorben von Mutterletb ani Warumb bin ich nicht vmbkommen/ ehe ich auff den Schostgesetzet wards iOolchejhrc klagcaber folvns fur distmahl ttli- cher massen zum Trost dieNen / tvicdcr die kiagc / vamit anjctzo das ChnrfursilichcHaust vnd diese Kirche er- fu!let ist/vber der Fursilichenieiche vnserslungstgc- bornett Herm/ HermZohanN- StgHMttNden/ der such vierzchen Wochen ein Marggraff zu Brandenburg	 Vnd es fein jetzund solche Zeiten / da deren sehr viel sein/ die auch mit Jobo sagen vnd klagen möchten ; Warumb bin ich nicht gestorben von Mutterleib an? Warumb bin ich nicht vmbkommen / ehe ich auff den Schoß gesetzet ward? Solche ihre klage aber sol vns für dißmahl etlicher massen zum Trost dienen / wieder die klage / damit anjetzo das Churfürstliche Hauß vnd diese Kirche erfüllet ist / vber der Fürstlichen Leiche vnsers Jüngstgebornen Herrn / Herrn Johann-Sigißmunden / der auch vierzehen Wochen ein Marggraff zu Brandenburg gewesen / aber sein Fürstliches Erbtheil gar bald in einem andern Reiche gesuchet / dahin die edle Seele / für 4. Wochen / zu dem Vater der Seelen abgeschieden / vnd von den hochbetrübten Churfürstlichen Eltern / zu seinem theil auff Erden / nichts mehr gefordert / alß nur das Erbliche Ruhe-kämmerlein / dahin jetzo der zarte Leichnam zu seinen Vätern sol eingesetzet werden. Vnd Er begehret ja nicht / das wir viel klagens vnd weinens vber jhn treiben sollen. Was Hiob vergeblich gewünschet / das ist diesem Jungen Marggrafen bald wiederfahren: Er liget jtzt in der stille / schläffet vnd ruhet mit seinen Vätern. Es ist jhm so wol vnd besser / dann vielen Königen vnd Fürsten / die das wüste gebawet / oder das gebawete verwüstet / die viel Gold vnd Silbers gesamblet / oder das gesamblete verzehret. Er ist im Frieden / vnd darff nicht fürchten das toben der Gottlosen / auch nicht hören die stimme des Drengers. Damit	Beispiel HK-OCR/FR9 – ohne Training mit selbst def.„Sprache“ HKOCR010 OG17 F Builtins Fehlende Zeichen: 32 ??-–--L --äSFeeloe ueeedeener Dt Überflüssige Zeichen: 43 ZOTZZZZZZZ Z,ZZZMkcZc cMZZZZzZZv cfcfZccssZ)ik Vnd es fein jetzund solche Zsiten Z da deren schr viel sein/ die auch mit lobo sagen vnd klagen möchten ; Warumb bin ich nicht gestorben von Mutterletb ane Warumb bin ich nicht vmbkommen / ehe ich auff den Schoß gesetzet ward s iOolchejhrc klage aber sol vns für dißmahl etli cher massen zum Trost vierten Z wieder die klage z damit anjetzo das Churfürstliche Hauß vnd diese Kirche er füllet ist/vber der Fürstlichenieiche vnsers lüngstge bornen HermZ Herrn lohann-Sigißmunden/ der auch vierzehen Wochen ein Marggraff zu Brandenburg ...

62 Fortschritt in OCR-Erkennung (hier mit HK-OCR) durch Wortliste

6.2 Im Projekt erreichte Erkennungsgüte

Eine wie im Beitrag von Thomas Stäcker (vgl. 8.7) dargestellte statistisch abgesicherte Fehleranalyse wurde bisher nicht durchgeführt. Gleichwohl wurden Stichproben aus den verschiedenen Projektphasen untersucht. Diese zum Teil erst aus der Zeit nach der Optimierung entstandenen Daten sollen hier kurz vorgestellt werden, um Eindrücke anhand von kompletten Beispielen zu erlauben. Die Fehlerzählung erfolgte analog zu der in der Herzog August Bibliothek Wolfenbüttel durchgeführten Methode (Worttrennungen, fehlende bzw. überschüssige Spatien fanden keine Berücksichtigung).

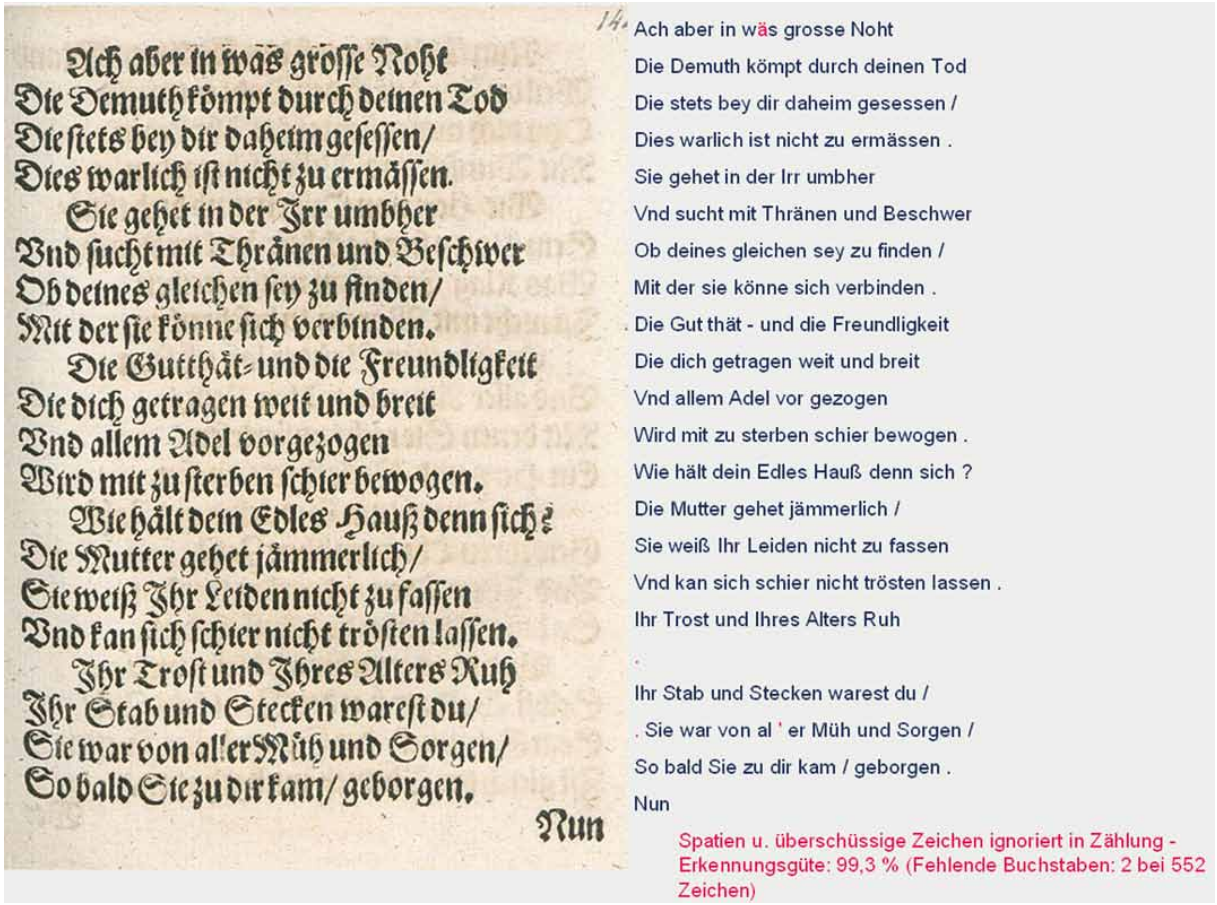
Wie bereits geschildert (s. Kap. 5) wurde in der Optimierungsphase nur in sehr bescheidenem Maße trainiert, so dass grundsätzlich starke Probleme bei der Erkennung von in Marginalien gesetzten Texten auftraten, was zum einen an der sehr kleinen und oft schwer lesbaren Schrift, zum anderen an der mangelnden Segmentierung lag. Auch die Erkennung von Ziffern und Zeichen in fremden (als den im Trainingsmaterial vorkommenden) Alphabeten muss daher als unbefriedigend bezeichnet werden. Die folgenden Fälle sind zu unterscheiden:

1. Marge:

Homogene Texte von Simon Dach in einer Schrift. Hierzu wurden vorab 50 Seiten Text trainiert.
BIT-Alpha (vgl. Abb. 63):
Von sieben wahllos herausgesuchten Seiten wiesen fünf eine Erkennungsgüte von 97,7 % und höher auf; in zwei Fällen wurden über 99 % erreicht.

2. Marge:

Weniger homogene Texte von Simon Dach mit wenigen unterschiedlichen Schriften. Hierzu wurden vorab 60 Titelblätter und 30 Seiten Text trainiert.
HK-OCR:
Hier wurden nur drei Seiten stichprobenartig untersucht. Die Ergebnisse lagen bei 93,6 % bis 96,4 %.

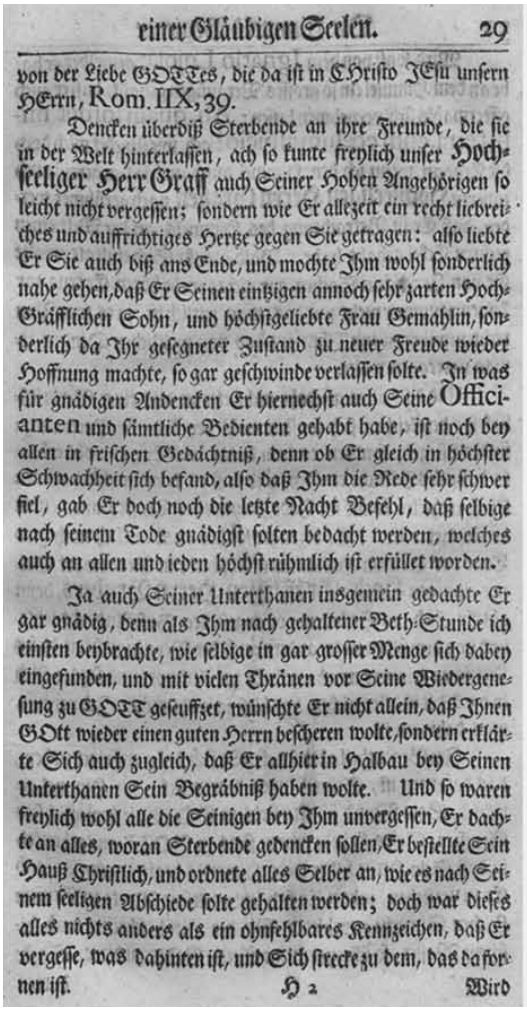


63 Beispiel BIT-Alpha; 1. Optimierungslauf: 40 in: Yi 851-3, S. 2

3. Marge (Massenlauf):

Nicht vorselektierte Texte Funeralschriften vornehmlich aus dem 17. Jh. Kein weiteres Training hat stattgefunden; die vorab erstellten Musterdateien der 2. Marge wurden für das gesamte Material angewendet.
BIT-Alpha (vgl. Abb. 64):
Von acht einzelnen Seiten wiesen nur vier eine Erkennungsgüte über 90 % auf (91,8 % bis 95,8 %).
HK-OCR (vgl. Abb. 65):
Hier wurden nur sechs Seiten stichprobenartig untersucht. Die Ergebnisse lagen bei fünf Seiten deutlich über 94 % (88 % bis 97,9 %).

Eine Hochrechnung erlauben diese willkürlich erhobenen Daten kaum. Es gilt außerdem zu berücksichtigen, dass im so genannten Massenlauf keine weiteren Selektionen mehr vorgenommen werden konnten, was sicherlich bei weiterem Training eigens getrennter Gruppen zu Qualitätssteigerungen geführt hätte. Speziell für BIT-Alpha ist diese Art der Prozessierung nicht die angezeigte, da wie bereits oben geschildert, die Software in besonderem Maße ein Training erfordert und nicht auf einen großen Fundus vorhandener Muster zurückgegriffen werden kann.



einer Gläubigen Seelen. 29

von der Liebe GOTTes, die da ist in CHristo JEsu unsern HErrn, Rom. IIX, 39.

Dencken überdiß Sterbende an ihre Freunde, die sie in der Welt hinterlassen, ach so kunte freylich unser Hochseeliger Herr Graff auch Seiner Hohen Angehörigen so leicht nicht vergessen; sondern wie Er allezeit ein recht liebreiches und auffrichtiges Hertze gegen Sie getragen: also liebte Er Sie auch biß ans Ende, und mochte Ihm wohl sonderlich nahe gehen, daß Er Seinen eintzigen annoch sehr zarten Hoch-Gräfflichen Sohn, und höchstgeliebte Frau Gemahlin, sonderlich da Ihr gesegneter Zustand zu neuer Freude wieder Hoffnung machte, so gar geschwinde verlassen solte. In was für gnädigen Andencken Er hiernechst auch Seine Officianten und sämtliche Bedienten gehabt habe, ist noch bey allen in frischen Gedächtniß, denn ob Er gleich in höchster Schwachheit sich befand, also daß Ihm die Rede sehr schwer fiel, gab Er doch noch die letzte Nacht Befehl, daß selbige nach seinem Tode gnädigst solten bedacht werden, welches auch an allen und ieden höchst rühmlich ist erfüllet worden.

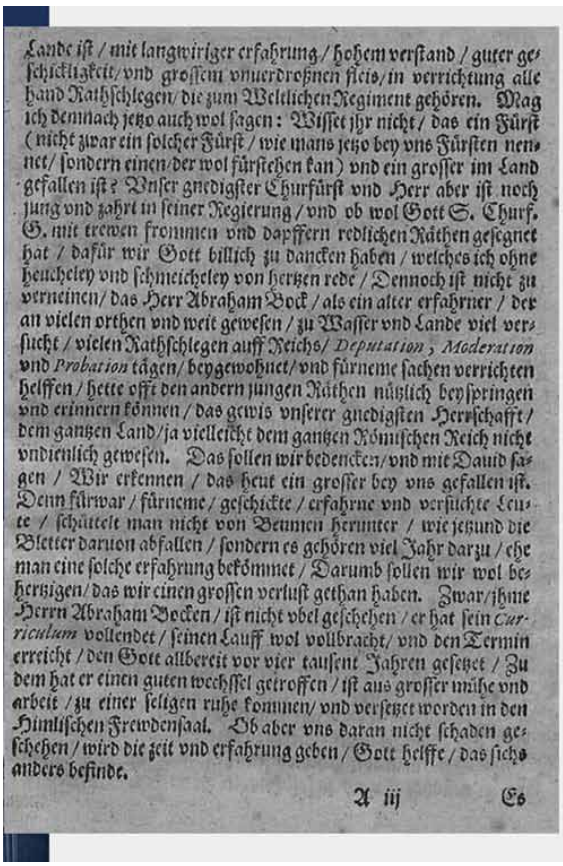
Ja auch Seiner Unterthanen insgemein gedachte Er gar gnädig, denn als Ihm nach gehaltener Beth-Stunde ich einsten beybrachte, wie selbige in gar grosser Menge sich dabey eingefunden, und mit vielen Thränen vor Seine Wiedergenesung zu GOTT geseufzet, wünschte Er nicht allein, daß Ihnen GOtt wieder einen guten Herrn bescheren wolte, sondern erklärte Sich auch zugleich, daß Er allhier in Halbau bey Seinen Unterthanen Sein Begräbniß haben wolte. Und so waren freylich wohl alle die Seinigen bey Ihm unvergessen, Er dachte an alles, woran Sterbende gedencken sollen, Er bestellte Sein Hauß Christlich, und ordnete alles Selber an, wie es nach Seinem seeligen Abschiede solte gehalten werden; doch war dieses alles nichts anders als ein ohnfehlbares Kennzeichen, daß Er vergesse, was dahinten ist, und Sich strecke zu dem, das da vornen ist.

H 2 Wird

ey
einerGlänbigenSeeJen .
.
oi , n aer giebc q ; OeTes , die da ist in icHrisio : 3Esii unsern HEern , Rom . IIX , g9 .
Dencken überdiß Sterbende an ihre Freunde , die sie in der Welt hinterlassen , ach so kunte freylich unser Hoch - seeliger HerrGrtff auch Sciner Hohen Angehörigen so leicht nichtverqessen ; sondern wie Er allezeit ein recht liebrei - ches undauffriehtiges Herl ; e gegen Sie getragen : also liebte Er Sie auch biß ansEnde , und mochtelhm wohl sonderlich nahe gehen , daß Et Seinen eintzigen annoch sehr zarten Hoch - Gräfflichen Sohn , und höchstgeliebte Frau Gemahlin , soni derlich da Ihr gesegneter Zustand zu neuer Freude wieder Hoffnung machte , so gar geschivindeverlassensolte . Initias für gnädigen Aindencken Er hiernechst auch Seine Offici - anten und sämtliche Bedicnten gehabt habe , ist noch bey allen in frischen E) eddchtniß , denn ob Er gleich in höchster Schwachhcitsich befand , also daß Ihm die Redc sehrschwet fiel , gab Er doch noch die letzte Nacht Befehl , daß selbige nach seinem Todc gnädigst solten bedacht werden , weiches auch an allen und iedele höchstrühmlich ist erfülletworden - Ja auch Seiner Linterthanen insgemeiAt gedachte Et gar gnädig , denn als Ihm nach gehaltenerBeth - Stunde ich einsten beybrachte , wie selbige in gar grofferMenge sich dabey eingefunden , und mit vielen Thränen vor Seine Wiedergene - sung zuGOTT gesenfftet , wunschte Er nichtallein , daß Ihnen GOtt wieder einen guten Herrn bescheren wolte , sondern erklär - te Sich auch zugleich , daß Er allhierin Halbau bey Seinen Unterthanen Sein Begräbniß haben wolte . Und so waren freylich wohl alle die Seinigen bey Ihm undergessen , Er dach - tean alles , woran Sterbende gedencken sollen , Er besteslteSein Hauß Christlich , und ordnete alles Selber an , wie es nach Sei –

Spatien u. überschüssige Zeichen ignoriert in Zählung - Erkennungsgüte: 97,2 % (Fehlende Buchstaben: 47 bei 1664 Zeichen)

64 Beispiel BIT-Alpha; Massenlauf: Ee 705-1071, S. 29



Lande ist / mit langwiriger erfahrung / hohem verstand / guter geschickligkeit/vnd grossem vnuerdroßnen fleis/ in verrichtung alle hand Ralhschlegen/ die zum Weltlichen Regiment gehören. Mag ich demnach jetzo auch wol sagen: Wisset jhr nicht/ das ein Fürst (nicht zwar ein solcher Fürst / wie mans jetzo bey vns Fürsten nennet/ sondern einen/der wol fürstehen kan) vnd ein grosser im Land gefallen ist? Vnser gnedigster Churfürst vnd Herr aber ist noch jung vnd zahrt in seiner Regierung /vnd ob wol Gott S. Churf. G.mit trewen frommen vnd dapffern redlichen Räthen gesegnet hat / dafür wir Gott billich zu dancken haben /welches ich ohne heucheley vnd schmeicheley von hertzen rede /^ Dennoch ist nicht zu verneinen/ das Herr Abraham Bock /als ein alter erfahrner / der an vielen orthen vnd weit gewesen /^ zu Wasser vnd Lande viel versucht / vielen Rathschlegen auff Reichs/ *Deputation* , */^oderat/on* vnd *Brobation* lägen/ beygewohnet/ vnd fürneme sachen verrichten helffen / heue offt den andern jungen Räthen nützlich beyspringen vnd ei innern können /das gewis vnserer gnedigsten Herrschafft/ dem gantzen Land/ja vielleicht dcm gantzen Römischen Reich nicht vndienlich gewesen. Das sollen wirbedenckcn/vnd mit ^auid sagen / Wir erkennen / das heut ein grosser bey vns gefallen ist. Denn^fürwar /fürneme/ geschickte/ erfahrne vnd versuchte Lente / schüttelt man iiicht voit Beunien herunter / wie jetzund die Bletter damon abfallen / sondern es gehören viel Jahr darzu / ehe man eine solche erfahrung bekömmet/ Darumb sol.en wir wol behertzigen/ das wir einen grossen verlust gethan haben. Zwar/jhme Herrn Abraham Bocken / ist nicht vbel geschehen / er hat sein *Cur^ricu/um* vollendet/ seinen Lauff wol vollbracht/ vnd den Termin erreicht /den Gott allbereit vor vier tausent Jahren gcsctzet / Zu dem hat er einen guten wcchssel getroffen / ist aus grosser mühe vnd arbeit /zu einer seligen ruhe kommen/ vnd ver setzet worden inden Himlischen Frewdensaal. Ob aber vns daran nicht schaden geschehen / wird die zeit vnd erfahrung geben / Gott helffe / das sichs anders befinde.

A iij Es

^ softwareseitig nicht erkanntes Zeichen; Spatien u. überschüssige Zeichen ignoriert in Zählung - Erkennungsgüte: 98,3 % (Fehlende Buchstaben: 27 bei 1791 Zeichen)

65 Beispiel HK-OCR; Massenlauf: Ee 705-106, S. 7

6.3 Hervorzuhebende Vor- und Nachteile der getesteten Software für die OCR deutschsprachiger Alter Drucke

6.3.1 *Good practice*

(a) BIT-Alpha

BIT-Alpha zeichnet sich aus durch:

- eine reiche Parametervielfalt in Binarisierung und Segmentierung, wodurch auf Layout- und Schrifttypeneigenschaften detailliert eingegangen werden kann;
- die Möglichkeit, auch Schrifttypen-Teile (Schaft, Fuß usw.) als eigene Muster auf „Special characters“ zu trainieren, um die Teile später mittels Sequencer zusammenzusetzen (so können bestimmte häufige OCR-Fehler wie brüchige „n“, „m“, „u“ repariert werden);
- die mit dem Hilfsprogramm BIT-Knowledge gegebene Möglichkeit zur Mischung mehrerer Musterbibliotheken;
- eine fein steuerbare Wortkorrektur durch gewichtete Zeichenersetzungskoeffizienten.

(b) HK-OCR/FREngine 9

HK-OCR besitzt:

- eine aus dem Stand (ohne langwierige Justierungen) gegebene sehr gute Segmentierung auf Zeichenniveau und
- eine gut bedienbare Benutzeroberfläche für Wortkorrekturen mit Berücksichtigung der Korrekturen in der Koordinatenberechnung des XML-Exports.

6.3.2 Gegenwärtige Nachteile

(a) BIT-Alpha

Die wichtigsten Nachteile von BIT-Alpha sind darin zu sehen, dass

- unnötige „Compound“-Formate der Parameter- und Hilfsdateien (s. Kap. 4.1) zur manchmal unerwünschten Zusammenspeicherung von nicht zusammengehörigen Informationen führen und
- zu den zahlreichen Optionen bzgl. Parameteränderungen eine gründliche Dokumentation und eine übersichtliche Vorschau der Auswirkungen solcher Justierungen fehlte.

(b) HK-OCR/FREngine 9

Als für die getesteten OCR-Aufgaben ungünstig erwiesen sich folgende Eigenschaften von HK-OCR, die offenbar auf Komponenten der FineReader-Engine beruhen:

- die stark beschränkte Navigation im Trainingsteil (s. Kap. 4.4);
- die unpraktische Bedienung des Mustereditor-Fensters;
- die fehlende Kombinierbarkeit trainierter Musterbestände sowie
- die Erschwernis durch die Doppel-Lizenzierung einerseits der Software beim Hersteller, andererseits der von ABBYY zu erwerbenden und zu installierenden Einzelseitenlizenzen.

6.4 Anforderungen an Software-Weiterentwicklung

Aus Sicht eines an realistischen Produktionsbedingungen für OCR interessierten Anwenders haben einige Produkteigenschaften gefehlt, die für eine Stapel verarbeitende Software eigentlich bewährt sind und die auch mit dem Programmdesign beider Produkte ohne weiteres vereinbar wären.

Ein Anliegen dieser Studie ist es daher, zur Diskussion über nahe liegende Anforderungen bei der Weiterentwicklung von für historische Drucke geeigneten OCR-Tools beizutragen und eine aktive Nachfrage nach wesentlichen Produktmerkmalen anzuregen, die den Herstellern eine adressatengerechte Produktentwicklung ermöglicht.

Die folgenden, für die getesteten Programme BIT-Alpha und HK-OCR den Herstellern sinngemäß mitgeteilten (und in deren Software meist ohne weiteres integrierbar scheinenden) Anregungen können zugleich als allgemeiner Kriterienkatalog für die Auswahl bzw. Entwicklung von OCR-Produkten mit herangezogen werden.

Aufrufsituation im Betriebssystem

Könnten dem jeweiligen Programm als konventionelle Aufrufparameter schon beim Start alle Angaben (Namen der Konfigurations-, Muster- und Wörterbuch-/Sprachdateien, ebenso Bildverzeichnisse bzw. Bilddateinamen) übergeben werden, sinngemäß etwa:

```
OCRAPP.exe -conffile xx -patternfile yy -lexiconfile zz-outputdir xx/xx image1.tiff
OCRAPP.exe -conffile yy -patternfile zz -lexiconfile xx-outputdir xx/yy image2.tiff
...
```

Dann könnten Batchprozesse viel flexibler gestaltet werden, indem z. B. im selben Stapel zu bestimmten Bilder(gruppe)n bestimmte Konfigurationen oder Muster-/Wortbibliotheken übergeben werden. Auch für Beschränkungen, die aus der gegenwärtig „nur aus einem gemeinsamen Bildverzeichnis heraus“ möglichen Batch-Verarbeitung resultieren, wäre die explizite Angabe der zu prozessierenden Images hilfreich. Dadurch könnten die Images an ihrem Speicherort verbleiben, ohne dass der Bearbeiter ein eigentlich unmotiviertes (und gelegentlich wegen fehlender Rechte oder aus Speicherplatzgründen gar nicht mögliches) temporäres Hin-und-Her-Schieben von Quelldateien organisieren muss. Demzufolge wäre ein Anstoßen beliebig kleinteiliger Test- und Produktionsläufe mit sehr unterschiedlichen Bedingungen kein zeitlicher Faktor mehr.

Zudem wären die OCR-Anwendungen mit dieser konventionellen Aufrufsyntax so auch in beliebige Workflow-Systeme integrierbar, an denen Anwenderinstitutionen aus unterschiedlichen Gründen festhalten wollen. Die Entscheidung zwischen bibliothekseigenem Workflow und der Nutzung von Produkt XY wäre dann nicht mehr nötig.

Konfigurationsspeichermodell; Formate; Zusammenspiel von Parameter-, Muster- und weiteren Hilfsdateien

Da für eine an einzelne Vorlagen angepasste OCR die eingesetzten Binarisierungsparameter-, Muster-, Ersetzungsregel-, Sprach- und Lexikon-Dateien optimal ausgewählt und kombiniert werden müssen, sollten sie am besten:

- unabhängig voneinander einzeln abgespeichert werden;
- soweit möglich jederzeit auch außerhalb des OCR-Prozesses passiv lesbar sein (so für Protokoll- und Vergleichszwecke, z. B. wenn Nutzer ihre längerfristig Muster-, Parameter- und Regelsammlungen anlegen und pflegen);
- und je möglichst komfortabel editierbar sein.

Im Ergebnis sollte die gezielte Änderung oder Löschung bestimmter Parameter/Muster/Ersetzungsregeln/Wortlisteneinträge jederzeit möglich sein, ohne eine OCR-Analyse anzustoßen und ohne Auswirkungen auf nicht-betroffene Parameter/Muster/Regeln. Im Sinne „atomarer" Datenablage sollte für jede Steuerinformation nur ein eindeutiger Speicherort vorgesehen sein.

Für alle diese Hilfsdateien (außer den Mustern selbst) gilt, dass eine Speicherung in einem textartigen, les- und editierbaren Format (einfache Listen, Schlüssel-Wert-Paare wie in *.ini- oder *.config-Dateien, notfalls XML) möglich sein sollte und folgende Vorteile hätte:

- Es wäre erstmals ein durchgehendes *monitoring* und *logging* der im konkreten OCR-Lauf wirksamen Parameter, Sequenzen, Wortlisten oder Ersetzungskoeffizienten möglich – eine Voraussetzung für sinnvolle langfristige Weiterpflege durch die Anwender.
- Parameterlisten, Sequenzen, Wortlisten, lexikalische Ersetzungen usw. könnten bequem und wesentlich flexibler ergänzt, korrigiert oder in ausgewählten Teilen gelöscht werden.
- Zur Übertragung bewährter Parameter oder Regeln in eine andere Konfiguration ist bisher der manuelle Neueintrag jedes einzelnen Werts in zeitaufwendigen Dialogen der graphischen Benutzerschnittstelle nötig. Aus textartigen Hilfsdateien aber wäre eine selektive und dennoch kompakt „blockweise" Übertragung von Parametersätzen von einer Konfigurationsdatei in eine andere möglich.[88]

Insbesondere ist kein Grund erkennbar, ausgerechnet auch die vom Anwender selbst in Textform oder durch Tastatureingaben bereitgestellten Wortlisten, Listen erlaubter/verbotener Buchstaben, Ersetzungsregeln usw. in proprietären, vom Anwender nicht mehr lesbaren Binärformaten zu verwalten. Die sachgerechte Pflege dieser Bestände gestaltet sich dadurch unbegründet schwierig bis unmöglich.

Aber auch für die Speicherung der Muster selbst wäre ein dokumentiertes und anwendungsübergreifendes Format anzustreben, das die freie Bearbeitung und nach Möglichkeit sogar den Austausch von Mustern zwischen verschiedenen OCR-Anwendungen erlaubt. Die Formate von Open-Source-OCR-Anwendungen wurden nicht untersucht, könnten aber vermutlich zur Formulierung entsprechender Anforderungen herangezogen werden. Bei geeigneter Serialisierung (Aufeinanderfolge von Datensätzen) wären Kombination und Durchsuchbarkeit solcher Musterdateien trivial.

Benutzeroberfläche und Ergonomie

Während des Pilotprojekts ist viel Zeit mit ineffektiven Bedienungsabläufen verbraucht worden. Das betrifft nicht nur die OCR-bezogenen Softwareeigenschaften, zu denen im folgenden Anregungen genannt werden, sondern offenbar geraten längst bewährte und etabliert geglaubte Merkmale ergonomischen Programm-Designs in Neuentwicklungen gelegentlich in Vergessenheit. Umso wichtiger ist es, dass Anwender, die durch Aufträge an Softwarehäuser Einfluss auf die Programmgestaltung nehmen können, sich rechtzeitig über die Mindestanforderungen klar sind, die für eine Arbeitsweise mit vertretbarem Personalaufwand unverzichtbar sind.

Da die Wirtschaftlichkeit von maschineller OCR eng mit der für Training und Konfiguration aufzuwendenden Arbeitszeit zusammenhängt, sind im Anhang (s. Anh. 7.4) einige solcher allgemeiner Mindestanforderungen genannt, auf die geachtet werden sollte.

Die Zusammenarbeit mit beiden Dienstleistern hat gezeigt, dass Hinweise durchaus willkommen sind – schließlich sind Produkte, die einer schnellen technischen Entwicklung folgen müssen, praktisch immer „in Entwicklung". Bei konsequenter Nachfrage und rechtzeitiger Übernahme gewünschter Eigenschaften z. B. in ein Pflichtenheft würden diese Merkmale sicher nicht fehlen.

[88] Speziell die Stärke der B.I.T.-Programme, unterschiedliche Kombinationen von Binarisierungsparametern, Mustern, Sequenzen, Wortlisten, Lexikalischen Ersetzungsregeln usw. überhaupt zu ermöglichen, könnte so noch viel konsequenter eingesetzt und produktiv gemacht werden.

OCR-spezifisch wären folgende Anliegen zu nennen:[89]

Zeitsparende Orientierung über aktuell geltende Eigenschaften, Parameter usw. und ihre Auswirkungen

Wenn eine reiche Konfiguration von Binarisierungs- und Segmentierungsparametern angeboten wird, dann wird wegen ihrer Vieldimensionalität eine Beurteilung der Wirkung einzelner Änderungen fast unüberschaubar und zwingt bei der Konfiguration eines OCR-Stapels zur Beschränkung auf erfahrungsgestützte Vermutungen, ohne diese auch nur stichprobenartig mit vertretbarem Aufwand verifizieren zu können.

Eine Vorschau auf zu erwartende Ergebnisse könnte leicht gewonnen werden, wenn die Programme

- wie oben beschrieben alle nötigen Parameter auch in der Kommandozeile akzeptieren würden und
- die bisher nur in der Benutzeroberfläche erreichbaren Vorschau-Ansichten als Bilddateien exportieren könnten.

Denn dann könnte durch Batch-Testläufe wie

```
OCRAPP.exe -conffile xx image1.tiff --printlayer=BlockLayout
> image1-confxx-blocklayout.png
OCRAPP.exe -conffile yy image2.tiff --printlayer=CharLayout
> image2-confyy-blocklayout.png
OCRAPP.exe -conffile yy image2.tiff --printlayer=OCR
> image2-confyy-charlayout.png
```

usw.

sehr schnell (jedenfalls ohne von Bearbeitern aufzuwendende Zeit) ein repräsentativer Überblick über die konfigurationsbezogene Segmentierungsgüte und Layouterkennung einer größeren Bildmenge gewonnen und z. B. in einer automatisch erstellten Tabelle visuell vergleichend dargestellt werden.[90] Vermutlich würde es durch solch eine „Viele-Screenshots-Ansicht“ überhaupt erst möglich, die Auswirkung einer konkreten Parameteränderung auf einen vorliegenden Gesamt-Stapel halbwegs kontrolliert abzuschätzen. Bisher entscheidet man über die Anpassung eines Parameters anhand einer eher sehr kleinen Menge gesehener und einzeln analysierter Bilder (dabei mit Minuten Wartezeit je Bild) und muss sich für die Prognose auf die Intuition verlassen – mit der Gefahr, bei der „Rettung“ einer Seite die Erkennung vieler anderer zu beschädigen.

Interaktivität, frühe Evaluation der Einzelverarbeitungsschritte

Für alle Stufen gilt: Wo adaptive Selbstkonfiguration und automatische Korrektur nicht stattfinden oder eher heuristisch, d. h. nur möglicherweise optimal arbeiten, könnte Software mit unmittelbaren Warnmeldungen und Hinweisen auf sinnvolle interaktive Eingriffe die schnelle Findung der geeigneten Einstellungen und Konfigurationen unterstützen.

Speziell die Segmentierung sollte auch in der Stapelverarbeitung justierbar präsentiert werden, um auf druckspezifische Schwierigkeiten bei der Layouterkennung reagieren zu können. Gebraucht würden einerseits „halbfertige“ Konfigurationsvorlagen, wo der Anwender bestimmte Abstandsmaße, Eingangstexte und ähnliche Kriterien festlegen kann, anhand derer bestimmten Blöcken *für den aktuellen Stapel* eine bestimmte Textstruktureinheit zugewiesen wird,[91] andererseits eine am Bildschirm gut geführte Möglichkeit zur schnellen graphischen Verschiebung von Blockgrenzen sowohl für Einzelseiten als auch auf Beispielseiten mit Geltung für einen Stapel. Voraussetzung wäre hierfür u. a. eine Möglichkeit, Images eines Stapels auf dem Bildschirm in Schichten übereinander darzustellen und so Gruppen mit Layoutgemeinsamkeiten (einfachster Fall: linke Seiten vs. rechte Seiten) bilden zu können.

[89] Vgl. außerdem die einzelschrittbezogen genannten Desiderata in den Abschnitten des Kapitels 4.

[90] Hingewiesen sei auch auf Vorarbeiten im IMPACT-Projekt zu Evaluierungen bereits jeder Teilstufe des OCR-Prozesses. Das 2012 als Nachfolgeeinrichtung von IMPACT entstandene Kompetenzzentrum sollte als Ansprechpartner in Anspruch genommen werden.

[91] Unter den Konfigurationsparametern von BIT-Alpha gibt es etliche, die hierfür geeignet wären.

Alternative: Vorsorgliche Parallelverarbeitung mit späterer Evaluation

Eine in OCR-Algorithmen ansatzweise bereits anzutreffende Parallelverarbeitung[92] könnte als Vorbild dienen, bei der OCR inhomogener Vorlagen generell unterschiedliche Lesegänge durchzuführen und deren Ergebnisse für einen anschließenden Vergleich bereitzustellen.

Insoweit in solchen parallelen OCR-Outputs dieselben Teilblock- (Absatz- usw.) Segmentierungen erreicht und ausgezeichnet wurden, sollte hier sogar ein über die Selektion des besseren Gesamtseiten-Ergebnisses hinausgehender Mehrwert erreichbar sein, indem trotz seitenweiser OCR die Evaluation auf Blockebene ansetzen könnte und sich z. B. für einen Absatz der Output aus dem OCR-Lauf mit Konfiguration A, für einen anderen Absatz der Output aus dem OCR-Lauf mit Konfiguration B als passender erweist.

Öffnung der Blackboxes

Transparenz und Konfigurierbarkeit der wirkenden Faktoren

Alle sinnvoll variierbaren Faktoren sollten für den Anwender konfigurierbar sein und müssen in jedem Fall – auch nachträglich und auch dann, wenn die Einstellung automatisch erfolgte – visuell dargestellt und protokolliert werden können. Erfolge an einem OCR-Stapel sind nur nachnutzbar, wenn sich die Umstände reproduzieren lassen.

Modularität

Wo immer ein Anwenderinteresse an der Ausgabe von Teilergebnissen oder an der Einspeisung anderen Materials vorliegt, sollte die geschlossene Programmarchitektur geöffnet und eine voraussetzungslose Nutzung auch von Einzelkomponenten ermöglicht werden.

Es gibt beispielsweise keine Notwendigkeit, die gute Binarisierungsleistung einer Software nur im Zusammenhang mit dem OCR-Lauf in derselben Software zu erlauben.

Die Funktion der Korrekturkomponente sollte nur dann an das Vorhandensein der Original-Bilddateien geknüpft sein, wenn tatsächlich neue OCR-Analysen notwendig sind. Soweit hingegen nur die textartigen OCR-Leseergebnisse verarbeitet werden, sollten diese – ohne erneute OCR-Analyse – eingespeist werden können; idealerweise auch Material, das aus anderen Quellen als der OCR des Programms selbst kommt. Gerade wenn ein Korrekturmodul spezielle Stärken hat (HK-OCR seine gut bedienbare „Validierungs"-Oberfläche, BIT-Alpha seine gut justierbare automatische Nachkorrektur anhand gewichteter Zeichenersetzungsregeln), wäre sein Einsatz auf beliebigem Input wünschenswert.

Die jederzeitige Ausgabe von Zwischenergebnissen wird auch schon zur Evaluation von Teilschritt-Konfigurationen benötigt.

6.5 Schlussfolgerungen und Tipps; Ausblick

Informationsverluste als „Leitfaden"

Der vermutlich ergiebigste Hinweis für die OCR Alter Drucke dürfte sein, dass nur eine adäquat vorbereitete OCR Aussicht auf zufrieden stellende Ergebnisse bietet und dass hierzu eine gründliche Kenntnis des Materials notwendig ist.

Je nach Verwendungszweck[93] können unterschiedliche Qualitätsanforderungen bestehen, zu deren Überprüfung zunächst von der gewählten OCR-Software unabhängige Kriterien und Werkzeuge gefunden werden sollten. Zur Mindestausstattung von OCR-Projekten gehören daher Standardwerkzeuge für Textextraktion und einfache statistische Auswertungen; hierfür können die in allen Betriebssystemen gängigen Kommandozeilenfilter und eine Skriptsprache mit komfortabler Textverarbeitung wie Perl bereits genügen.

[92] Zu nennen ist hier z. B. in HK-OCR das mehrfache Lesen unter unterschiedlichen Bedingungen („Layouts") mit anschließender manueller oder automatischer Auswahl des „besten" Ergebnisses, s. Kap.4.3.3"

[93] Eine Übersicht der wichtigsten Verwendungsziele gibt z. B. Ralf Stockmann (SUB Göttingen) in einem Vortrag „Was tun mit den Ergebnissen der OCR?" im Rahmen des IMPACT-Projekts; darunter u. a. eine nach steigender „Sichtbarkeit" geordnete Aufzählung der Bereitstellungsarten: „Versteckt in Suchindex" - „Versteckt, aber Image-Highlighting der Fundstelle" - „Volltext als Layer hinter dem Image (etwa in PDF gebunden)" - „Volltext sichtbar über / neben dem Image" - „Nur Volltext sichtbar" - „Volltext als Download" - „Volltext für Harvester verfügbar (TEI Datei in OAI)", s. STOCKMANN 2010

Neben der „absoluten“ Qualitätsbeurteilung (zur Entscheidung, ob und wann sich ein OCR-Projekt lohnt) sind innerhalb der Arbeit häufig relative Entscheidungen zu treffen, z. B. in welche Richtung eine Konfiguration geändert werden soll, wofür es nicht auf die Perfektion von Fehlerkriterien ankommt, sondern nur darauf, ob die Tendenz einer Verbesserung oder Verschlechterung der Erkennungsrate abgelesen werden kann. Vergleiche müssen beliebig oft (und jeweils schon für Veränderungen nur eines Faktors) vorgenommen werden können, um aus den Ergebnissen die richtigen Schlüsse zu ziehen.

Für die Suche nach geeigneter Software oder nach Dienstleistern können dann die im Kapitel 3 genannten typischen Fehlerquellen der OCR als Entscheidungshilfe dienen. Interessenten sollten sich klar werden, welche der absehbaren Informationsverluste auch im eigenen konkreten Material auftreten werden und welche möglicherweise unproblematisch sind. Daraus folgernd könnten sie dann die Software oder den Dienstleister danach auswählen, wie sie mit den betreffenden Fehlerquellen umgeht.

Außerhalb der eigentlichen OCR-Leistungsfähigkeit werden Fragen der Nachhaltigkeit und der Bindung an Lizenzen und Formate eine Rolle spielen wie:

- Korrespondieren von Import- und Exportformaten mit den eigenen Geschäftsgängen;
- Wiederholbarkeit der OCR- und Korrekturläufe;
- Wiederverwendbarkeit von Konfigurationsdateien, Musterbibliotheken und Korrekturlexika;
- Auswirkungen einer eventuellen Softwareinstallation auf betriebliche Abläufe;
  usw.

Nicht zuletzt lohnt es sich, stets die denkbaren Auftragsmodi anhand der konkreten OCR-Aufgabe gegeneinander abzuwägen:

- Arbeit in eigener Regie durch Erwerb bzw. Lizenzierung der Software;
- Beauftragung der Gesamt-OCR als Dienstleistung (hier sollte eine Zielerkennungsgüte vertraglich festgelegt werden; s. auch Kap. 8.8);
- Arbeitsteilige Zusammenarbeit: z. B. mit eigener Produktion, eigenem Training, aber Beauftragung der Konfiguration schwieriger Parameter[94] – oder umgekehrt: externe Beauftragung von Stapelverarbeitungen, die der Anwender vorher selbst konfiguriert hat.[95]

6.5.1 Tipps für Anwender der getesteten Software: Wann wäre welches der getesteten Produkte geeignet?

Um sich *von der Software ausgehend* ein Bild zu machen, werden die Kapitel 4 und 5 empfohlen, wo in den einzelnen Abschnitten auf die Merkmale von HK-OCR und BIT-Alpha eingegangen wird.

Zur *Nachnutzbarkeit* von nutzereigenen Muster- und Wortbibliotheken gilt für beide Produkte ähnlich, dass die erstellten Bibliotheken prinzipiell aufbewahrt und wiederverwendet werden können. Selbstverständlich muss langfristig auf die Kompatibilität etwaiger Nachfolgeversionen der jeweiligen Programme geachtet werden. Eine wirklich flexible Nachnutzbarkeit für nicht vorhergeplante Fälle wird wie beschrieben durch die proprietären, vom Nutzer nur bedingt editierbaren Speicherformate eingeschränkt.

Wenn man *vom Einsatzbedarf* ausgehen möchte, könnten aus den Versuchen im Pilotprojekt folgende Empfehlungen abgeleitet werden:

Wenn eigenes Training erfolgen soll:

- BIT-Alpha eher für Szenarien, in denen
  - der zu erkennende Zeichenvorrat (insbesondere Diakritika, griechische, hebräische Zeichen u. ä.) im voraus nicht vollständig bekannt ist oder
  - es im Training darauf ankommt, frei und mit häufigen Sprüngen navigieren zu können.

[94] Nachteil dieser Variante wäre, dass unerwünschte Softwareeigenschaften wie eine kompliziert dargebotene oder schlecht dokumentierte Konfiguration honoriert würden.

[95] Ein möglicher Vorteil: die professionelle Materialkenntnis kann optimal eingesetzt werden.

- HK-OCR eher für Szenarien, in denen
  - der zu erkennende Zeichenvorrat (insbesondere Diakritika, griechische, hebräische Zeichen u. ä.) im voraus vollständig bekannt ist und in den Spracheinstellungen angegeben werden kann oder
  - ein Training überwiegend ohne Sprünge, d. h. fortlaufend über die ganze Seite erfolgen kann.

Für eine OCR-Prozessierung von Anfang bis Ende:

- BIT-Alpha eher für Szenarien, in denen
  - der Anwender vorlagenbedingt Einfluss auf Binarisierung und Segmentierung nehmen möchte oder muss und Zeit für die Konfiguration hat (Alternative: Konfiguration in Form einer bda-Datei beim Softwarhersteller bestellen; hier sollten Qualitätskriterien vereinbart werden);
  - eine ausreichend große ungefähr passende Mustermenge trainiert werden kann oder vorliegt;
  - ein die Vorlage weitgehend abdeckendes Lexikon zur Verfügung steht und eine automatische, gewichtet einstellbare lexikalische Korrektur bevorzugt wird.
    (Als Alternative können Konfiguration und/oder Training und/oder Stapelverarbeitung (mit/ohne Wortkorrektur) beim Softwarehersteller als Dienstleistung bei B.I.T. in Auftrag gegeben werden; hier empfiehlt sich, Zielmaße für die zu liefernde Erkennungsgüte zu vereinbaren.)

- HK-OCR eher für Szenarien, in denen
  - das Bildmaterial keine speziellen Schwierigkeiten für Binarisierung oder Segmentierung aufweist, die eine individuelle Änderung von Parametern erfordern würden;
  - eine sehr gut zur Vorlage passende Mustermenge trainiert werden kann oder vorliegt und die Stapelverarbeitungen sehr gut nach homogener Sprache, Schriftfamilie und -größe portioniert werden können;
  - eine manuelle lexikalische Korrektur bevorzugt wird bzw. auf eine automatische lexikalische Korrektur entweder verzichtet werden kann oder diese nachgelagert – dann ohne Mitpflege der Wortkoordinaten – vorgenommen wird.
    (Die Übernahme von Trainingsleistungen wurde zum Projektzeitpunkt nicht von der Herstellerfirma angeboten. Wenn der Anwender aber Musterdateien und Lexika bereitstellen kann, könnte auch hier die Verarbeitung als Dienstleistung beim Softwarehersteller Herrmann & Kraemer vereinbart werden.)

Nicht zum Untersuchungsgegenstand gehörte die Eignung für jüngere Frakturschriften, es soll aber wenigstens erwähnt werden, dass für zunehmende Nähe zum 19. Jahrhundert die in der FineReader-Engine bereits enthaltenen Fraktur-Muster immer besser passen und ein eigenes Training hierfür meist unnötig wäre.

Bei beiden Produkten bleibt es Aufgabe des Anwenders, die jeweiligen XML-Exporte in ein Zielformat seiner Volltextpräsentation oder in einen Suchindex zur Volltextsuche zu überführen.

Für eine schrittweise, modulare OCR-Verarbeitung

Wenn Eingriffsmöglichkeiten bzw. Einspeisung und Auswertung vor und nach jedem Zwischenschritt erwünscht sind, sind beide Produkte derzeit nicht darauf vorbereitet. Theoretisch mögliche Kombinations- und Anschlussmöglichkeiten bestehen daher nur am Anfang und Ende der jeweiligen Verarbeitungskette.

Eine wirklich offene und an jedem Verarbeitungsschritt ihre Teilergebnisse liefernde Verarbeitungskette scheint allerdings auch in anderen Produkten nicht Standard zu sein; möglicherweise sollte das 2012 gegründete IMPACT-Kompetenzzentrum hier in Anspruch genommen werden, um einschlägige Anforderungen zu bündeln.

6.5.2 Tipps für generelle Planungen von OCR Alter Drucke

Modularität des Workflows

Jedes OCR-Projekt ist bestimmten Eigenheiten und Rahmenbedingungen unterworfen, die wesentlich im zu erschließenden Material selbst begründet sind. Gleichwohl soll versucht werden, einige allgemeine Hinweise zu formulieren. Wie bereits eingangs erwähnt, lässt sich der Workflow auch dank internationaler Projekte und ihrer Ergebnisse zunehmend modularisieren und standardisieren. Wesentliches Ziel ist dabei, sich zukünftig für das jeweilige Projekt einen geeigneten Workflow aus verschiedenen Komponenten zusammenzustellen und somit die für die Aufgabenstellung besten Resultate zu erhalten. Im hier behandelten Projekt bestand diese Möglichkeit nicht bzw. konnte nur sehr begrenzt durch Parametrierung der jeweiligen Software Einfluss genommen werden. Auf die dadurch entstandenen Nachteile wurde wiederholt hingewiesen. Die Softwaretests zeigen, dass sich beide Softwareprodukte durch Stärken und Schwächen in verschiedenen Bereichen auszeichnen. Bestünde die Möglichkeit, die einzelnen starken Module für eine spezielle Anwendung zu kombinieren und auch weitere freie Module in den eigenen Workflow zu integrieren, ergäben sich neue Chancen zur Ergebnisoptimierung. Die Einbeziehung weiterer Software war jedoch nicht Gegenstand dieses Pilotprojekts. In der Fachdiskussion spielen das Workflowmanagement und seine technische Unterstützung eine wachsende Rolle.[96]

Materialkenntnis

Wichtige Kriterien für die Planung von OCR leiten sich zuerst aus der Sichtung und Kenntnis der Drucke selbst ab. Sowohl ihr Erhaltungszustand, typographische Besonderheiten als auch inhaltliche Bezüge bilden Themen, mit denen eine Beschäftigung im Vorfeld lohnt. Der Erhaltungszustand des Drucks sowie die Qualität seiner Digitalisierung erfordern geeignete Aktivitäten für eine optimale Binarisierung. Wie auch im Projekt gezeigt werden konnte, beeinflussen verschiedene Binarisierungsmethoden direkt die OCR-Erkennung. Das Wissen über Inhalt und Darstellung des zu erkennenden Wortmaterials ist wesentlich für die Einbeziehung möglichst passender Musterdateien und Wörterbücher. Neben der regionalen und zeitlichen Einordnung der Drucke ist daher die Kenntnis der überwiegend verwendeten Sprache (auch der Fachsprache) sowie der vorkommenden Schriftenvielfalt von Vorteil. Besteht die Möglichkeit, in einem Projekt geringfügige Anpassungen der Konfigurationsparameter vorzunehmen, können bereits wenige Vorsortierungen das Ergebnis verbessern. Eine grobe Einteilung nach Druckorten (-gebieten) bzw. Erscheinungszeiträumen ist hierbei anzuraten; gegebenenfalls lohnt auch eine Systematisierung nach äußeren Merkmalen oder Gattungen.

Einbeziehung der OCR-Ergebnisse

Eine zusätzliche Optimierungsvariante ist es, die Erkennungsgüte der erstellten Dateien zu analysieren. Hierfür bieten sich verschiedene Methoden an – wichtig ist letztlich die Vergleichbarkeit der Ergebnisse innerhalb eines Projekts. Heranzuziehen sind durchaus auch Werte, die durch die OCR selbst erstellt und den Wörtern oder Zeichen als Information (z. B. Konfidenzwerte) mitgegeben werden. All diese Informationen können Indikatoren für eine nachträgliche Selektion sein, um Material zurückzustellen oder einer gesonderten Behandlung zu unterziehen. Beispielsweise ist folgendes Szenario vorstellbar: die Erkennungsgüte einer Seite ist extrem schlecht. Nach Blick auf das Image wird festgestellt, dass die Seite im Buch kopfüber eingebunden war, wodurch die Zeichenerkennung fast unmöglich wurde. Diesen Fehler könnte man folglich unkompliziert beheben. Ein anderes Beispiel ist die gezielte Analyse der Wörterbuchnutzung: Eine signifikant geringe Wörterbuchnutzung (so die Software derartige Auswertungen zulässt) kann anzeigen, dass sehr wahrscheinlich ein Werk in einer fremden Sprache vorliegt. „Fremd“ bedeutet in diesem Fall: die Sprache gehört nicht zu denen, die im zeitgleich bearbeiteten Material überwiegend vertreten sind und mit deren Wörterbuch aktuell gearbeitet wird. Weitere Einsatzfelder von Evaluationsergebnissen verschiedenster Qualität sind aufwandsabhängig denkbar.

[96] Ein interessanter Vortrag zum Thema Workflowdesign: SCHLARB 2011.

Sekundäre Materialien

Hinsichtlich der Sortierung wird offenbar, wie wichtig es bei Alten Drucken schon in der vorbereitenden Phase sein kann, die vorhandenen bibliographischen Beschreibungen zu analysieren. Existieren in den oftmals konvertierten Titelaufnahmen gar Informationen über die Schriftart („In Fraktur gedruckt“ oder dergleichen), gibt es eine Sprachbezeichnung? Kann diese Sprachbezeichnung eventuell auf einfache Weise generiert oder übernommen werden? Lässt sich mit diesem Titelmaterial möglicherweise bereits ein Wortschatz für ein Wörterbuch finden? Gibt es vergleichbare Texte, die als Referenzmaterialien verwendet werden können?

Arbeitsteilung

Eine andere wesentliche Frage ist die der Nutzung von Dienstleistungen als Alternative zum Inhouse-Einsatz von OCR-Software. Die Beantwortung hängt von vielen Faktoren ab. Nicht zuletzt wird jedoch die technische und personelle Ausstattung in der jeweiligen Einrichtung maßgeblich sein. Nicht zu unterschätzen (und das belegen die Erfahrungen im Projekt) sind Rechenzeiten und Speicherbedarf.[97]

Bibliotheksspezifische Schlussfolgerungen

Es eröffnen sich vielfältige Möglichkeiten zur Gestaltung eigener Arbeitsabläufe. Mit jedem weiteren Workflowschritt entstehen neue Fragen und Ideen. Mithin ergeben sich für die Funeralschriften/Alten Drucke der Staatsbibliothek zu Berlin weitere Testszenarien, von denen nur einige angerissen werden sollen:

Ist es realisierbar, den OCR-Service der VZG[98] unter Nutzung des Abbyy Recognition Servers über das Ticket-System mit den via HK-OCR erstellten Muster- und Wörterbuchdateien zu „füttern“? Bei erfolgreichem Einsatz könnten auf diese Weise weitere Funeralschriften einer Massen-OCR-Erkennung unterzogen werden. Gleichzeitig wäre zu testen, ob diese Muster und Parameter auch für inhaltlich vielfältigeres Material (jedoch möglichst zeitlich vergleichbares) zu besseren Ergebnissen führen als die Standardeinstellungen.

Grundsätzlich ist für die Staatsbibliothek weiterhin zu untersuchen, ob es für einzelne Workflowschritte andere Softwarelösungen gibt, mit denen bessere Ergebnisse zu erzielen sind. In diesem Zusammenhang könnte auch die Opensource-Software Tesseract[99] Gegenstand vergleichender Betrachtungen sein. Weitere Ansatzpunkte bilden die vielfältigen Bemühungen im von der EU geförderten IMPACT-Projekt, Phasen der OCR-Vorbereitung und OCR-Nachbearbeitung zur Optimierung zu nutzen.[100] Beispielhaft wurden inzwischen Evaluationswerkzeuge (auch einzelschrittbezogen) entwickelt, deren Anwendung vergleichbare Ergebnisse verschiedener Herkunft ermöglichen soll. Der intensive Kontakt und die Zusammenarbeit mit verwandten Projekten ist für den Erfahrungsaustausch von unschätzbarem Wert (Vgl. eine Auflistung von Projekten im Anhang 7.5).

[97] Die Prozessierung eines Images (Quartformat) mit einem Standardbürocomputer dauerte durchschnittlich bis zu 10 Minuten; eine Alto-Datei ist ca. 70 KB, eine FR-XML ca. 500 KB, eine FR-Image-Datei ca. 5-7.000 KB groß.

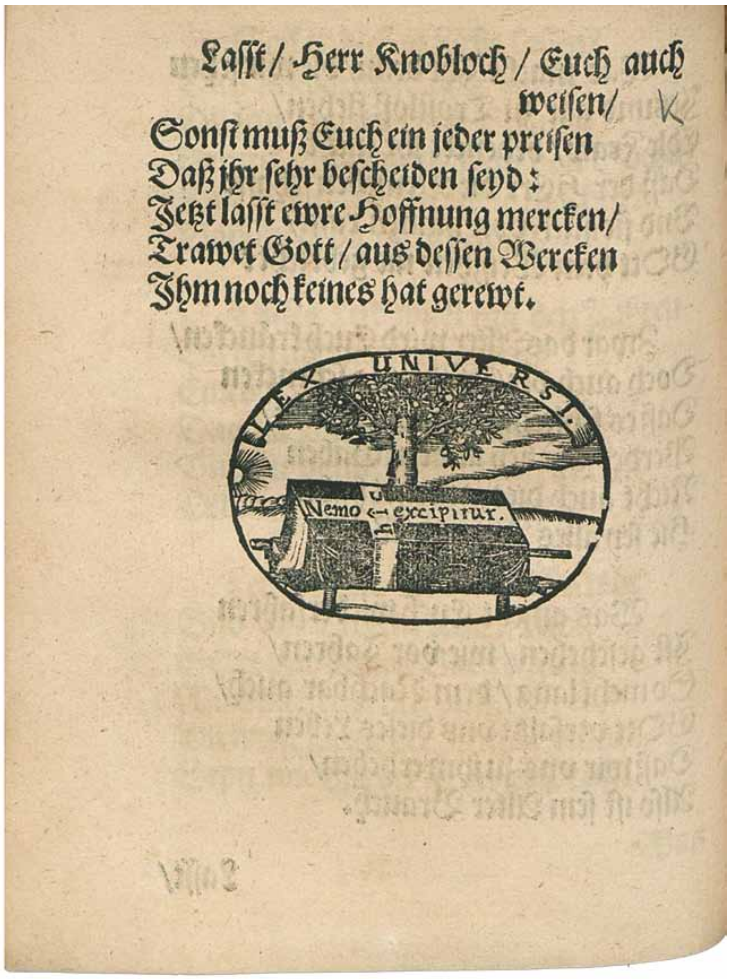
[98] http://www.gbv.de/wikis/cls/OCR-Service_der_VZG [Stand: 02/2013]

[99] http://code.google.com/p/tesseract-ocr/ [Stand: 02/2013]

[100] Zu nennen ist hier die Entwicklung des IMPACT Interoperability Framework.

7 Anhänge
7.1 Beispiele Exportformat

Zur Demonstration der XML-Ausgabeformate wurde eine sehr kurze, nur aus sechs Zeilen bestehende Seite gewählt:


Lasst / Herr Knobloch / Euch auch
weisen/
Sonst muß Euch ein jeder preisen
Daß ihr sehr bescheiden seyd:
Jetzt lasst ewre Hoffnung mercken/
Trawet Gott / aus dessen Wercken
Ihm noch keines hat gerewt.
SIGILLVM UNIVERSITATIS
Nemo excipitur.

66 Faksimile – Beispielseite

(a) BIT-Alpha:
ALTO

```
<?xml version="1.0" encoding="utf-8"?>
<alto xmlns="http://schema.ccs-gmbh.com/ALTO" xmlns:a="http://schema.ccs-gmbh.com/ALTO" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <a:Description>
    <a:MeasurementUnit>mm10</a:MeasurementUnit>
    <a:sourceImageInformation>
      <a:fileName>F:\Batch SBB\dachxlag_635359391_orig\00000003.tif</a:fileName>
    </a:sourceImageInformation>
    <a:OCRProcessing ID="ID_BIT_1609ec32-7185-4b66-ac85-b6949a62bf81">
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:42:47</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Color Enhancement</a:processingStepDescription>
        <a:processingStepSettings>ContrastR=0.5 ContrastG=0.5 ContrastB=0.5 GammaR=1.0 GammaG=1.0 GammaB=1.0 LuminanceR=0.5
        LuminanceG=0.5 LuminanceB=0.5</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:42:47</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Rotation</a:processingStepDescription>
        <a:processingStepSettings>Type=None Margins={Left=0.5 Right=0.5 Top=0.800000011920929 Bottom=0.5}</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:42:47</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Binarisation</a:processingStepDescription>
        <a:processingStepSettings>SourceBPP=24 Algorithm=Intensity based algorithm</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:42:50</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Cleaning</a:processingStepDescription>
        <a:processingStepSettings>ContrastR=0.5 ContrastG=0.5 ContrastB=0.5 GammaR=1.0 GammaG=1.0 GammaB=1.0 LuminanceR=0.5
        LuminanceG=0.5 LuminanceB=0.5</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:42:52</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Remove Dots</a:processingStepDescription>
        <a:processingStepSettings>ContrastR=0.5 ContrastG=0.5 ContrastB=0.5 GammaR=1.0 GammaG=1.0 GammaB=1.0 LuminanceR=0.5 LuminanceG=0
        LuminanceB=0.5</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:42:57</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Blackborder elimination</a:processingStepDescription>
        <a:processingStepSettings>ContrastR=0.5 ContrastG=0.5 ContrastB=0.5 GammaR=1.0 GammaG=1.0 GammaB=1.0 LuminanceR=0.5 LuminanceG=0
        LuminanceB=0.5</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:43:03</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Detection of horizonzal lines</a:processingStepDescription>
        <a:processingStepSettings>ForceBitmap=true MaxThickness=0.5 MinThickness=0.0 MaxWhiteLength=0.1 MinBlackLength=0.9
        MinTotalLength=2.0 Margins={Left=0.5 Right=0.5 Top=0.800000011920929 Bottom=0.5}</a:processingStepSettings>
        <a:processingSoftware>
          <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
          <a:softwareName>BIT-Alpha</a:softwareName>
          <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
        </a:processingSoftware>
      </a:preProcessingStep>
      <a:preProcessingStep>
        <a:processingDateTime>2011-06-29T19:43:04</a:processingDateTime>
        <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
        <a:processingStepDescription>Detection of vertical lines</a:processingStepDescription>
        <a:processingStepSettings>ForceBitmap=true MaxThickness=0.1 MinThickness=0.0 MaxWhiteLength=0.0 MinBlackLength=0.9
        MinTotalLength=5.0 Margins={Left=0.5 Right=0.5 Top=0.800000011920929 Bottom=0.5}</a:processingStepSettings>
        <a:processingSoftware>
```

67 BIT-Alpha Export ALTO

```
        <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
        <a:softwareName>BIT-Alpha</a:softwareName>
        <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
      </a:processingSoftware>
    </a:preProcessingStep>
    <a:preProcessingStep>
      <a:processingDateTime>2011-06-29T19:43:05</a:processingDateTime>
      <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
      <a:processingStepDescription>Segmentation</a:processingStepDescription>
      <a:processingStepSettings>MinHeight=0.1 MinWidth=0.1 MinVertDist=2.0 MinHorDist=2.0 Margins={Left=0.5 Right=0.5
      Top=0.800000011920929 Bottom=0.5}</a:processingStepSettings>
      <a:processingSoftware>
        <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
        <a:softwareName>BIT-Alpha</a:softwareName>
        <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
      </a:processingSoftware>
    </a:preProcessingStep>
    <a:preProcessingStep>
      <a:processingDateTime>2011-06-29T19:43:06</a:processingDateTime>
      <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
      <a:processingStepDescription>Region identification</a:processingStepDescription>
      <a:processingStepSettings>Default=Binary ColorRegions={Detect=false} PaletteRegions={Detect=false} BinaryRegions={Detect=false}
      </a:processingStepSettings>
      <a:processingSoftware>
        <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
        <a:softwareName>BIT-Alpha</a:softwareName>
        <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
      </a:processingSoftware>
    </a:preProcessingStep>
    <a:ocrProcessingStep>
      <a:processingDateTime>2011-06-29T19:43:16</a:processingDateTime>
      <a:processingAgency>Staatsbibliothek zu Berlin - PK</a:processingAgency>
      <a:processingStepDescription>Optical Character Recognition</a:processingStepDescription>
      <a:processingStepSettings>not implemented.</a:processingStepSettings>
      <a:processingSoftware>
        <a:softwareCreator>B.I.T. Bureau Ingénieur Tomasi</a:softwareCreator>
        <a:softwareName>BIT-Alpha</a:softwareName>
        <a:softwareVersion>2.0.38.595 (Rel. 38)</a:softwareVersion>
      </a:processingSoftware>
    </a:ocrProcessingStep>
  </a:OCRProcessing>
</a:Description>
<a:Layout>
  <a:Page ID="ID_BIT_41462251-f383-4bc1-a3f6-f2301a5327d1" HEIGHT="1939" WIDTH="1424" PHYSICAL_IMG_NR="0" QUALITY="OK" PROCESSING=
  "ID_BIT_1609ec32-7185-4b66-ac85-b6949a62bf81">
    <a:PrintSpace ID="ID_BIT_70278dba-eec8-44ea-b094-d85119a7733b" HEIGHT="1808.87" WIDTH="1323.25" HPOS="50.0" VPOS="80.0">
      <a:TextBlock ID="ID_BIT_59d0acb9-1bbc-4ef6-9c2d-471d558705a5" HEIGHT="505" WIDTH="991" HPOS="350" VPOS="149">
        <a:TextLine ID="ID_BIT_c6c8b85f-920d-4b89-b1ad-30b4e24e6770" HEIGHT="83.82" WIDTH="892.39" HPOS="99.06" VPOS="6.77">
          <a:String HEIGHT="79.59" WIDTH="117.69" VPOS="6.77" HPOS="99.06" CONTENT="Lasse"/>
          <a:SP WIDTH="123.61" VPOS="6.77" HPOS="99.06"/>
          <a:String HEIGHT="79.59" WIDTH="30.48" VPOS="6.77" HPOS="222.67" CONTENT="/"/>
          <a:SP WIDTH="38.1" VPOS="6.77" HPOS="222.67"/>
          <a:String HEIGHT="79.59" WIDTH="137.16" VPOS="6.77" HPOS="260.77" CONTENT="Herr"/>
          <a:SP WIDTH="157.48" VPOS="1.27" HPOS="260.77"/>
          <a:String HEIGHT="72.81" WIDTH="245.53" VPOS="4.23" HPOS="418.25" CONTENT="Knobloch"/>
          <a:SP WIDTH="261.62" VPOS="5.5" HPOS="418.25"/>
          <a:String HEIGHT="70.27" WIDTH="28.79" VPOS="6.77" HPOS="679.87" CONTENT="/"/>
          <a:SP WIDTH="46.57" VPOS="6.77" HPOS="679.87"/>
          <a:String HEIGHT="70.27" WIDTH="130.39" VPOS="6.77" HPOS="726.44" CONTENT="Euch"/>
          <a:SP WIDTH="157.48" VPOS="5.93" HPOS="726.44"/>
          <a:String HEIGHT="71.12" WIDTH="107.53" VPOS="5.08" HPOS="883.92" CONTENT="auch"/>
        </a:TextLine>
        <a:TextLine ID="ID_BIT_7e62def2-46be-48f6-98ee-d1146ca1efc2" HEIGHT="2.54" WIDTH="3.39" HPOS="159.17" VPOS="73.66">
          <a:String HEIGHT="2.54" WIDTH="3.39" VPOS="73.66" HPOS="159.17" CONTENT="."/>
        </a:TextLine>
        <a:TextLine ID="ID_BIT_49fadae8-6170-4e08-a73b-95f294a1fb91" HEIGHT="77.05" WIDTH="303.95" HPOS="687.49" VPOS="80.43">
          <a:String HEIGHT="71.97" WIDTH="158.33" VPOS="80.43" HPOS="687.49" CONTENT="weisen"/>
          <a:SP WIDTH="159.17" VPOS="80.43" HPOS="687.49"/>
          <a:String HEIGHT="71.97" WIDTH="24.55" VPOS="80.43" HPOS="846.67" CONTENT="/"/>
          <a:SP WIDTH="103.29" VPOS="86.78" HPOS="846.67"/>
          <a:String HEIGHT="64.35" WIDTH="41.49" VPOS="93.13" HPOS="949.96" CONTENT="."/>
        </a:TextLine>
        <a:TextLine ID="ID_BIT_126ea2a4-ba77-44ec-b9b2-ff9421760f70" HEIGHT="80.43" WIDTH="820.42" HPOS="3.39" VPOS="137.16">
          <a:String HEIGHT="73.66" WIDTH="160.87" VPOS="137.16" HPOS="3.39" CONTENT="Sonst"/>
          <a:SP WIDTH="193.04" VPOS="137.16" HPOS="3.39"/>
          <a:String HEIGHT="73.66" WIDTH="96.52" VPOS="137.16" HPOS="196.43" CONTENT="muß"/>
          <a:SP WIDTH="108.37" VPOS="141.82" HPOS="196.43"/>
          <a:String HEIGHT="67.73" WIDTH="128.69" VPOS="146.47" HPOS="304.8" CONTENT="Euch"/>
          <a:SP WIDTH="139.7" VPOS="148.17" HPOS="304.8"/>
          <a:String HEIGHT="65.19" WIDTH="69.43" VPOS="149.86" HPOS="444.5" CONTENT="ein"/>
          <a:SP WIDTH="82.97" VPOS="150.28" HPOS="444.5"/>
          <a:String HEIGHT="66.04" WIDTH="115.99" VPOS="150.71" HPOS="527.47" CONTENT="jeder"/>
          <a:SP WIDTH="128.69" VPOS="151.55" HPOS="527.47"/>
          <a:String HEIGHT="65.19" WIDTH="167.64" VPOS="152.4" HPOS="656.17" CONTENT="peissen"/>
        </a:TextLine>
        <a:TextLine ID="ID_BIT_7fad1c16-25cb-4ae5-b4d9-fed34033a0ee" HEIGHT="71.97" WIDTH="724.75" HPOS="0.0" VPOS="214.21">
          <a:String HEIGHT="69.43" WIDTH="86.36" VPOS="214.21" HPOS="0.0" CONTENT="Daß"/>
          <a:SP WIDTH="115.99" VPOS="214.21" HPOS="0.0"/>
          <a:String HEIGHT="69.43" WIDTH="86.36" VPOS="214.21" HPOS="115.99" CONTENT="jhr"/>
          <a:SP WIDTH="101.6" VPOS="216.75" HPOS="115.99"/>
          <a:String HEIGHT="66.04" WIDTH="92.29" VPOS="219.29" HPOS="217.59" CONTENT="sehr"/>
          <a:SP WIDTH="105.83" VPOS="219.71" HPOS="217.59"/>
          <a:String HEIGHT="66.04" WIDTH="248.92" VPOS="220.13" HPOS="323.43" CONTENT="bescheiden"/>
          <a:SP WIDTH="268.39" VPOS="220.13" HPOS="323.43"/>
          <a:String HEIGHT="66.04" WIDTH="100.75" VPOS="220.13" HPOS="591.82" CONTENT="seyd"/>
          <a:SP WIDTH="114.3" VPOS="219.71" HPOS="591.82"/>
          <a:String HEIGHT="65.19" WIDTH="18.63" VPOS="219.29" HPOS="706.12" CONTENT=":"/>
        </a:TextLine>
```

68 BIT-Alpha Export ALTO (Fortsetzung)

```
          <a:TextLine ID="ID_BIT_4abe08d2-e8f9-433a-bbba-9a0880e2e487" HEIGHT="77.89" WIDTH="874.61" HPOS="0.0" VPOS="284.48">
            <a:String HEIGHT="66.89" WIDTH="124.46" VPOS="284.48" HPOS="0.0" CONTENT="Jetzt"/>
            <a:SP WIDTH="135.47" VPOS="285.75" HPOS="0.0"/>
            <a:String HEIGHT="66.04" WIDTH="99.06" VPOS="287.02" HPOS="135.47" CONTENT="lasse"/>
            <a:SP WIDTH="112.61" VPOS="287.87" HPOS="135.47"/>
            <a:String HEIGHT="71.12" WIDTH="117.69" VPOS="288.71" HPOS="248.07" CONTENT="ewre"/>
            <a:SP WIDTH="147.32" VPOS="288.71" HPOS="248.07"/>
            <a:String HEIGHT="71.12" WIDTH="236.22" VPOS="288.71" HPOS="395.39" CONTENT="Hoffnung"/>
            <a:SP WIDTH="251.46" VPOS="292.1" HPOS="395.39"/>
            <a:String HEIGHT="66.89" WIDTH="199.81" VPOS="295.49" HPOS="646.85" CONTENT="mercken"/>
            <a:SP WIDTH="201.51" VPOS="295.49" HPOS="646.85"/>
            <a:String HEIGHT="66.89" WIDTH="26.25" VPOS="295.49" HPOS="848.36" CONTENT="/"/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_fb094242-77e3-4695-aa8e-917cbfb9440f" HEIGHT="79.59" WIDTH="868.68" HPOS="0.0" VPOS="354.75">
            <a:String HEIGHT="69.43" WIDTH="198.97" VPOS="354.75" HPOS="0.0" CONTENT="Trawet"/>
            <a:SP WIDTH="212.51" VPOS="357.29" HPOS="0.0"/>
            <a:String HEIGHT="67.73" WIDTH="132.08" VPOS="359.83" HPOS="212.51" CONTENT="Gott"/>
            <a:SP WIDTH="140.55" VPOS="359.83" HPOS="212.51"/>
            <a:String HEIGHT="67.73" WIDTH="23.71" VPOS="359.83" HPOS="353.06" CONTENT="/"/>
            <a:SP WIDTH="33.02" VPOS="361.53" HPOS="353.06"/>
            <a:String HEIGHT="66.04" WIDTH="99.06" VPOS="363.22" HPOS="386.08" CONTENT="aus"/>
            <a:SP WIDTH="110.07" VPOS="364.07" HPOS="386.08"/>
            <a:String HEIGHT="66.89" WIDTH="140.55" VPOS="364.91" HPOS="496.15" CONTENT="dessen"/>
            <a:SP WIDTH="151.55" VPOS="366.18" HPOS="496.15"/>
            <a:String HEIGHT="66.89" WIDTH="220.98" VPOS="367.45" HPOS="647.7" CONTENT="Wercken"/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_72020365-be74-4556-8710-51632c34990d" HEIGHT="77.05" WIDTH="717.13" HPOS="0.0" VPOS="428.41">
            <a:String HEIGHT="71.12" WIDTH="132.08" VPOS="428.41" HPOS="0.0" CONTENT="Ihm"/>
            <a:SP WIDTH="141.39" VPOS="429.68" HPOS="0.0"/>
            <a:String HEIGHT="71.12" WIDTH="117.69" VPOS="430.95" HPOS="141.39" CONTENT="noch"/>
            <a:SP WIDTH="127.85" VPOS="432.22" HPOS="141.39"/>
            <a:String HEIGHT="71.12" WIDTH="151.55" VPOS="433.49" HPOS="269.24" CONTENT="keines"/>
            <a:SP WIDTH="162.56" VPOS="434.34" HPOS="269.24"/>
            <a:String HEIGHT="69.43" WIDTH="83.82" VPOS="435.19" HPOS="431.8" CONTENT="hat"/>
            <a:SP WIDTH="93.13" VPOS="435.19" HPOS="431.8"/>
            <a:String HEIGHT="70.27" WIDTH="171.87" VPOS="435.19" HPOS="524.93" CONTENT="gerewt"/>
            <a:SP WIDTH="175.26" VPOS="435.19" HPOS="524.93"/>
            <a:String HEIGHT="70.27" WIDTH="16.93" VPOS="435.19" HPOS="700.19" CONTENT="."/>
          </a:TextLine>
        </a:TextBlock>
        <a:TextBlock ID="ID_BIT_31e5d647-63a5-4880-9b45-ead04f5b660e" HEIGHT="502" WIDTH="717" HPOS="488" VPOS="723"/>
        <a:TextBlock ID="ID_BIT_37402e39-2b3d-4b14-b757-c80891a5f62b" HEIGHT="292" WIDTH="168" HPOS="498" VPOS="1435">
          <a:TextLine ID="ID_BIT_2f4716bb-6e35-4e20-9c1d-262e1e93926d" HEIGHT="25.4" WIDTH="13.55" HPOS="0.0" VPOS="233.68">
            <a:String HEIGHT="25.4" WIDTH="13.55" VPOS="233.68" HPOS="0.0" CONTENT="."/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_ca23d635-5c2f-4ced-9e6b-0aa38d6c4d71" HEIGHT="2.54" WIDTH="3.39" HPOS="8.47" VPOS="277.71">
            <a:String HEIGHT="2.54" WIDTH="3.39" VPOS="277.71" HPOS="8.47" CONTENT="."/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_5836981f-c2f8-43ec-bbdd-589b6433557d" HEIGHT="7.62" WIDTH="9.31" HPOS="5.93" VPOS="284.48">
            <a:String HEIGHT="66.89" WIDTH="199.81" VPOS="295.49" HPOS="646.85" CONTENT="mercken"/>
            <a:SP WIDTH="201.51" VPOS="295.49" HPOS="646.85"/>
            <a:String HEIGHT="66.89" WIDTH="26.25" VPOS="295.49" HPOS="848.36" CONTENT="/"/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_fb094242-77e3-4695-aa8e-917cbfb9440f" HEIGHT="79.59" WIDTH="868.68" HPOS="0.0" VPOS="354.75">
            <a:String HEIGHT="69.43" WIDTH="198.97" VPOS="354.75" HPOS="0.0" CONTENT="Trawet"/>
            <a:SP WIDTH="212.51" VPOS="357.29" HPOS="0.0"/>
            <a:String HEIGHT="67.73" WIDTH="132.08" VPOS="359.83" HPOS="212.51" CONTENT="Gott"/>
            <a:SP WIDTH="140.55" VPOS="359.83" HPOS="212.51"/>
            <a:String HEIGHT="67.73" WIDTH="23.71" VPOS="359.83" HPOS="353.06" CONTENT="/"/>
            <a:SP WIDTH="33.02" VPOS="361.53" HPOS="353.06"/>
            <a:String HEIGHT="66.04" WIDTH="99.06" VPOS="363.22" HPOS="386.08" CONTENT="aus"/>
            <a:SP WIDTH="110.07" VPOS="364.07" HPOS="386.08"/>
            <a:String HEIGHT="66.89" WIDTH="140.55" VPOS="364.91" HPOS="496.15" CONTENT="dessen"/>
            <a:SP WIDTH="151.55" VPOS="366.18" HPOS="496.15"/>
            <a:String HEIGHT="66.89" WIDTH="220.98" VPOS="367.45" HPOS="647.7" CONTENT="Wercken"/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_72020365-be74-4556-8710-51632c34990d" HEIGHT="77.05" WIDTH="717.13" HPOS="0.0" VPOS="428.41">
            <a:String HEIGHT="71.12" WIDTH="132.08" VPOS="428.41" HPOS="0.0" CONTENT="Ihm"/>
            <a:SP WIDTH="141.39" VPOS="429.68" HPOS="0.0"/>
            <a:String HEIGHT="71.12" WIDTH="117.69" VPOS="430.95" HPOS="141.39" CONTENT="noch"/>
            <a:SP WIDTH="127.85" VPOS="432.22" HPOS="141.39"/>
            <a:String HEIGHT="71.12" WIDTH="151.55" VPOS="433.49" HPOS="269.24" CONTENT="keines"/>
            <a:SP WIDTH="162.56" VPOS="434.34" HPOS="269.24"/>
            <a:String HEIGHT="69.43" WIDTH="83.82" VPOS="435.19" HPOS="431.8" CONTENT="hat"/>
            <a:SP WIDTH="93.13" VPOS="435.19" HPOS="431.8"/>
            <a:String HEIGHT="70.27" WIDTH="171.87" VPOS="435.19" HPOS="524.93" CONTENT="gerewt"/>
            <a:SP WIDTH="175.26" VPOS="435.19" HPOS="524.93"/>
            <a:String HEIGHT="70.27" WIDTH="16.93" VPOS="435.19" HPOS="700.19" CONTENT="."/>
          </a:TextLine>
        </a:TextBlock>
        <a:TextBlock ID="ID_BIT_31e5d647-63a5-4880-9b45-ead04f5b660e" HEIGHT="502" WIDTH="717" HPOS="488" VPOS="723"/>
        <a:TextBlock ID="ID_BIT_37402e39-2b3d-4b14-b757-c80891a5f62b" HEIGHT="292" WIDTH="168" HPOS="498" VPOS="1435">
          <a:TextLine ID="ID_BIT_2f4716bb-6e35-4e20-9c1d-262e1e93926d" HEIGHT="25.4" WIDTH="13.55" HPOS="0.0" VPOS="233.68">
            <a:String HEIGHT="25.4" WIDTH="13.55" VPOS="233.68" HPOS="0.0" CONTENT="."/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_ca23d635-5c2f-4ced-9e6b-0aa38d6c4d71" HEIGHT="2.54" WIDTH="3.39" HPOS="8.47" VPOS="277.71">
            <a:String HEIGHT="2.54" WIDTH="3.39" VPOS="277.71" HPOS="8.47" CONTENT="."/>
          </a:TextLine>
          <a:TextLine ID="ID_BIT_5836981f-c2f8-43ec-bbdd-589b6433557d" HEIGHT="7.62" WIDTH="9.31" HPOS="5.93" VPOS="284.48">
            <a:String HEIGHT="7.62" WIDTH="9.31" VPOS="284.48" HPOS="5.93" CONTENT="."/>
          </a:TextLine>
        </a:TextBlock>
      </a:PrintSpace>
    </a:Page>
  </a:Layout>
</alto>
```

69 BIT-Alpha Export ALTO (Fortsetzung)

(b) HK-OCR:
FineReader-XML

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://www.abbyy.com/FineReader_xml/FineReader8-schema-v2.xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="1.0" producer="FineReader 8.0" xsi:schemaLocation="http://www.abbyy.com/FineReader_xml/FineReader8-schema-v2.xml
http://www.abbyy.com/FineReader_xml/FineReader8-schema-v2.xml" mainLanguage="OldGerman" languages="OldGerman">
  <page width="1710" height="2311" resolution="300">
    <block blockType="Text" blockName="" isHidden="true" l="413" t="170" r="1596" b="795">
      <region>
        <rect l="413" t="170" r="1596" b="795"/>
      </region>
      <text>
        <par align="Right" rightIndent="1" lineSpacing="85">
          <line baseline="260" l="533" t="177" r="1589" b="279">
            <formatting lang="OldGerman" ff="Arial" fs="17." spacing="-20" style="0">
              <charParams l="533" t="189" r="574" b="263" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="97"
              charConfidence="81" serifProbability="255">L</charParams>
              <charParams l="577" t="207" r="610" b="259" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="97"
              charConfidence="98" serifProbability="255">a</charParams>
              <charParams l="614" t="190" r="641" b="275" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">s</charParams>
              <charParams l="630" t="189" r="659" b="274" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">s</charParams>
              <charParams l="650" t="197" r="673" b="259" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">t</charParams>
              <charParams l="681" t="193" r="714" b="270" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">/</charParams>
              <charParams l="714" t="189" r="733" b="275" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
            </formatting>
            <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
              <charParams l="733" t="190" r="799" b="275" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">H</charParams>
              <charParams l="805" t="205" r="828" b="260" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">e</charParams>
              <charParams l="830" t="206" r="856" b="259" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">r</charParams>
              <charParams l="860" t="206" r="887" b="259" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">r</charParams>
              <charParams l="887" t="190" r="911" b="275" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
              <charParams l="911" t="190" r="966" b="263" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">K</charParams>
              <charParams l="970" t="207" r="1010" b="261" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="98" serifProbability="255">n</charParams>
              <charParams l="1013" t="207" r="1043" b="260" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">o</charParams>
              <charParams l="1048" t="192" r="1080" b="262" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">b</charParams>
              <charParams l="1082" t="192" r="1104" b="261" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">l</charParams>
              <charParams l="1104" t="208" r="1143" b="261" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="81" serifProbability="255">o</charParams>
              <charParams l="1141" t="192" r="1200" b="279" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">c</charParams>
              <charParams l="1141" t="192" r="1200" b="279" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">h</charParams>
              <charParams l="1200" t="190" r="1219" b="271" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
              <charParams l="1219" t="197" r="1253" b="271" suspicious="true" characterHeight="52" hasUncertainHeight="false" baseline="
              wordStart="true" wordFromDictionary="false" wordNormal="false" wordNumeric="false" wordIdentifier="false" wordPenalty="0"
              meanStrokeWidth="97" charConfidence="100" serifProbability="255">/</charParams>
              <charParams l="1253" t="192" r="1275" b="271" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
              <charParams l="1275" t="194" r="1323" b="263" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">E</charParams>
              <charParams l="1326" t="207" r="1364" b="261" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">u</charParams>
              <charParams l="1370" t="192" r="1429" b="277" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="98" serifProbability="255">c</charParams>
              <charParams l="1370" t="192" r="1429" b="277" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="98" serifProbability="255">h</charParams>
              <charParams l="1429" t="177" r="1460" b="264" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
              <charParams l="1460" t="205" r="1491" b="257" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">a</charParams>
              <charParams l="1496" t="199" r="1532" b="255" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="100" serifProbability="255">u</charParams>
              <charParams l="1536" t="177" r="1589" b="264" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
              wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
              charConfidence="80" serifProbability="255">c</charParams>
```

70 HK-OCR Export FineReader-XML

```
<charParams l="1536" t="177" r="1589" b="264" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
charConfidence="80" serifProbability="255">h</charParams>
</formatting>
</line>
</par>
<par startIndent="806" lineSpacing="85">
<line baseline="345" l="1229" t="281" r="1590" b="364">
<formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
<charParams l="1229" t="290" r="1286" b="347" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="100" serifProbability="255">w</charParams>
<charParams l="1290" t="296" r="1313" b="346" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="98" serifProbability="255">e</charParams>
<charParams l="1318" t="290" r="1333" b="345" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="96" serifProbability="255">i</charParams>
<charParams l="1337" t="281" r="1367" b="359" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="1355" t="298" r="1376" b="346" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1381" t="295" r="1417" b="345" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="98" serifProbability="255">n</charParams>
<charParams l="1418" t="285" r="1446" b="358" characterHeight="48" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="55"
charConfidence="100" serifProbability="255">/</charParams>
<charParams l="1446" t="281" r="1533" b="359" characterHeight="48" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="1533" t="286" r="1590" b="364" suspicious="true" characterHeight="48" hasUncertainHeight="false" baseLine="
wordStart="true" wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2"
meanStrokeWidth="55" charConfidence="18" serifProbability="255">^</charParams>
</formatting>
</line>
<line baseline="429" l="423" t="361" r="1392" b="446">
<formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
<charParams l="423" t="361" r="495" b="435" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">S</charParams>
<charParams l="499" t="379" r="531" b="430" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">o</charParams>
<charParams l="534" t="377" r="571" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="576" t="363" r="616" b="445" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="576" t="363" r="616" b="445" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">t</charParams>
<charParams l="625" t="375" r="684" b="430" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">m</charParams>
<charParams l="688" t="377" r="727" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">u</charParams>
<charParams l="730" t="361" r="766" b="444" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="766" t="361" r="778" b="445" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="778" t="363" r="826" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">E</charParams>
<charParams l="830" t="377" r="868" b="427" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">u</charParams>
<charParams l="874" t="361" r="931" b="445" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">c</charParams>
<charParams l="874" t="361" r="931" b="445" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">h</charParams>
<charParams l="931" t="361" r="944" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="944" t="376" r="966" b="427" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="970" t="368" r="986" b="424" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">i</charParams>
<charParams l="989" t="376" r="1026" b="425" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="1026" t="364" r="1042" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="1042" t="364" r="1060" b="442" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">j</charParams>
<charParams l="1063" t="378" r="1085" b="427" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1089" t="364" r="1122" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">d</charParams>
<charParams l="1126" t="378" r="1149" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">e</charParams>
```

71 HK-OCR Export FineReader-XML (Fortsetzung)

```
87          <charParams l="1153" t="379" r="1178" b="430" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
            charConfidence="100" serifProbability="255">r</charParams>
88          <charParams l="1178" t="364" r="1194" b="442" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
89          <charParams l="1194" t="379" r="1229" b="446" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="100" serifProbability="255">p</charParams>
90          <charParams l="1232" t="380" r="1259" b="430" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="100" serifProbability="255">r</charParams>
91          <charParams l="1262" t="381" r="1286" b="431" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="98" serifProbability="255">e</charParams>
92          <charParams l="1291" t="375" r="1304" b="429" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="96" serifProbability="255">i</charParams>
93          <charParams l="1308" t="367" r="1340" b="446" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="100" serifProbability="255">s</charParams>
94          <charParams l="1327" t="381" r="1350" b="433" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="98" serifProbability="255">e</charParams>
95          <charParams l="1353" t="381" r="1392" b="433" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="10
            charConfidence="100" serifProbability="255">n</charParams>
96        </formatting>
97      </line>
98      <line baseline="513" l="420" t="430" r="1275" b="530">
99        <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
100         <charParams l="420" t="451" r="487" b="519" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">D</charParams>
101         <charParams l="493" t="466" r="525" b="513" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">a</charParams>
102         <charParams l="529" t="448" r="565" b="530" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">s</charParams>
103         <charParams l="565" t="448" r="575" b="530" suspicious="true" characterHeight="50" hasUncertainHeight="false" baseLine="0":
            </charParams>
104       </formatting>
105       <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-40" style="1">
106         <charParams l="575" t="452" r="595" b="529" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="80" serifProbability="255">j</charParams>
107         <charParams l="595" t="449" r="630" b="530" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="94" serifProbability="255">h</charParams>
108         <charParams l="630" t="464" r="660" b="511" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="94" serifProbability="255">r</charParams>
109       </formatting>
110       <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
111         <charParams l="660" t="430" r="676" b="529" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
112       </formatting>
113       <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-40" style="1">
114         <charParams l="676" t="449" r="707" b="526" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">s</charParams>
115         <charParams l="696" t="464" r="719" b="513" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">e</charParams>
116         <charParams l="722" t="430" r="756" b="529" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">h</charParams>
117         <charParams l="760" t="463" r="786" b="512" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">r</charParams>
118       </formatting>
119       <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
120         <charParams l="786" t="430" r="802" b="528" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
121         <charParams l="802" t="448" r="834" b="513" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">b</charParams>
122         <charParams l="837" t="462" r="861" b="511" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">e</charParams>
123         <charParams l="861" t="447" r="894" b="525" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">s</charParams>
124         <charParams l="884" t="430" r="943" b="528" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">c</charParams>
125         <charParams l="884" t="430" r="943" b="528" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">h</charParams>
126         <charParams l="948" t="461" r="972" b="511" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="98" serifProbability="255">e</charParams>
127         <charParams l="976" t="456" r="991" b="508" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">i</charParams>
128         <charParams l="996" t="449" r="1028" b="510" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false"
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">d</charParams>
129         <charParams l="1033" t="462" r="1055" b="510" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="false
            wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97
            charConfidence="100" serifProbability="255">e</charParams>
```

72 HK-OCR Export FineReader-XML (Fortsetzung)

```
<charParams l="1055" t="461" r="1096" b="512" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="8" meanStrokeWidth="97"
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="1096" t="430" r="1119" b="528" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="1119" t="447" r="1150" b="527" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="1138" t="465" r="1160" b="514" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1165" t="464" r="1199" b="530" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
charConfidence="100" serifProbability="255">y</charParams>
<charParams l="1205" t="454" r="1238" b="516" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="97"
charConfidence="100" serifProbability="255">d</charParams>
<charParams l="1238" t="447" r="1254" b="516" characterHeight="50" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="1254" t="461" r="1275" b="516" characterHeight="50" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="false" wordNormal="false" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="
charConfidence="96" serifProbability="255">:</charParams>
</formatting>
</line>
<line baseline="598" l="419" t="528" r="1451" b="612">
<formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
<charParams l="419" t="532" r="477" b="612" suspicious="true" characterHeight="49" hasUncertainHeight="false" baseLine="0"
wordStart="true" wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6"
meanStrokeWidth="106" charConfidence="100" serifProbability="255">J</charParams>
<charParams l="481" t="548" r="503" b="597" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="10
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="509" t="543" r="542" b="612" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="10
charConfidence="100" serifProbability="255">t</charParams>
<charParams l="509" t="543" r="542" b="612" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="10
charConfidence="100" serifProbability="255">z</charParams>
<charParams l="547" t="541" r="568" b="598" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="10
charConfidence="98" serifProbability="255">t</charParams>
<charParams l="568" t="532" r="581" b="611" characterHeight="49" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="581" t="532" r="597" b="598" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">l</charParams>
<charParams l="602" t="548" r="633" b="596" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">a</charParams>
<charParams l="637" t="532" r="665" b="611" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="658" t="532" r="684" b="610" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="677" t="539" r="698" b="597" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">t</charParams>
<charParams l="698" t="528" r="714" b="611" characterHeight="49" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="714" t="546" r="736" b="596" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="98" serifProbability="255">e</charParams>
<charParams l="739" t="541" r="795" b="597" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">w</charParams>
<charParams l="798" t="545" r="824" b="594" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">r</charParams>
<charParams l="827" t="544" r="852" b="595" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="861" t="529" r="926" b="610" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">H</charParams>
<charParams l="933" t="545" r="964" b="595" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">o</charParams>
<charParams l="970" t="528" r="1018" b="610" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">f</charParams>
<charParams l="970" t="528" r="1018" b="610" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">f</charParams>
<charParams l="1012" t="546" r="1048" b="596" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="1053" t="545" r="1090" b="597" suspicious="true" characterHeight="49" hasUncertainHeight="false" baseLine="
wordStart="false" wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6"
meanStrokeWidth="106" charConfidence="100" serifProbability="255">u</charParams>
<charParams l="1095" t="547" r="1131" b="598" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="1133" t="549" r="1169" b="612" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6" meanStrokeWidth="1
charConfidence="100" serifProbability="255">g</charParams>
<charParams l="1169" t="528" r="1185" b="609" characterHeight="49" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="1185" t="548" r="1242" b="602" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">m</charParams>
<charParams l="1248" t="551" r="1269" b="599" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
```

73 HK-OCR Export FineReader-XML (Fortsetzung)

```
                charConfidence="100" serifProbability="255">e</charParams>
169             <charParams l="1273" t="552" r="1300" b="602" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
                charConfidence="100" serifProbability="255">r</charParams>
170             <charParams l="1305" t="550" r="1329" b="602" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
                charConfidence="100" serifProbability="255">c</charParams>
171             <charParams l="1330" t="536" r="1358" b="602" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
                charConfidence="87" serifProbability="255">k</charParams>
172             <charParams l="1359" t="550" r="1381" b="600" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
                charConfidence="100" serifProbability="255">e</charParams>
173             <charParams l="1386" t="548" r="1421" b="600" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
                charConfidence="98" serifProbability="255">n</charParams>
174             <charParams l="1423" t="535" r="1451" b="609" characterHeight="49" hasUncertainHeight="false" baseLine="0" wordStart="false
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
                charConfidence="100" serifProbability="255">/</charParams>
175           </formatting>
176         </line>
177         <line baseline="683" l="422" t="616" r="1449" b="696">
178           <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
179             <charParams l="422" t="618" r="482" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="true"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2" meanStrokeWidth="9
                charConfidence="100" serifProbability="255">T</charParams>
180             <charParams l="483" t="634" r="511" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2" meanStrokeWidth="9
                charConfidence="100" serifProbability="255">r</charParams>
181             <charParams l="515" t="633" r="547" b="682" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2" meanStrokeWidth="9
                charConfidence="100" serifProbability="255">a</charParams>
182             <charParams l="549" t="626" r="605" b="685" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2" meanStrokeWidth="9
                charConfidence="100" serifProbability="255">w</charParams>
183             <charParams l="609" t="630" r="634" b="682" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2" meanStrokeWidth="9
                charConfidence="100" serifProbability="255">e</charParams>
184             <charParams l="634" t="624" r="658" b="682" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="2" meanStrokeWidth="9
                charConfidence="100" serifProbability="255">t</charParams>
185             <charParams l="658" t="616" r="672" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0"> </charParams>
186             <charParams l="672" t="616" r="740" b="682" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="true"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">G</charParams>
187             <charParams l="745" t="629" r="777" b="680" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">o</charParams>
188             <charParams l="782" t="621" r="804" b="680" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="98" serifProbability="255">t</charParams>
189             <charParams l="808" t="621" r="829" b="680" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="98" serifProbability="255">t</charParams>
190             <charParams l="829" t="616" r="839" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0"> </charParams>
191             <charParams l="839" t="639" r="857" b="690" suspicious="true" characterHeight="51" hasUncertainHeight="false" baseLine="0"
                wordStart="true" wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6"
                meanStrokeWidth="98" charConfidence="78" serifProbability="255">/</charParams>
192             <charParams l="857" t="621" r="866" b="647" suspicious="true" characterHeight="51" hasUncertainHeight="false" baseLine="0"
                wordStart="false" wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="6"
                meanStrokeWidth="98" charConfidence="18" serifProbability="255">^</charParams>
193             <charParams l="866" t="621" r="877" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0"> </charParams>
194             <charParams l="877" t="629" r="910" b="680" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="true"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">a</charParams>
195             <charParams l="915" t="628" r="952" b="680" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">u</charParams>
196             <charParams l="955" t="623" r="995" b="681" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">s</charParams>
197             <charParams l="995" t="616" r="1008" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0"> </charParams>
198           </formatting>
199           <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-40" style="1">
200             <charParams l="1008" t="618" r="1041" b="682" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="true"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">d</charParams>
201             <charParams l="1046" t="632" r="1068" b="683" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">e</charParams>
202             <charParams l="1072" t="616" r="1099" b="696" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">s</charParams>
203             <charParams l="1091" t="617" r="1118" b="695" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">s</charParams>
204             <charParams l="1110" t="633" r="1133" b="684" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">e</charParams>
205             <charParams l="1136" t="633" r="1174" b="684" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="false"
                wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="100" serifProbability="255">n</charParams>
206           </formatting>
207           <formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
208             <charParams l="1174" t="616" r="1187" b="692" characterHeight="51" hasUncertainHeight="false" baseLine="0"> </charParams>
209             <charParams l="1187" t="619" r="1268" b="687" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="true"
                wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="98"
                charConfidence="98" serifProbability="255">W</charParams>
```

74 HK-OCR Export FineReader-XML (Fortsetzung)

```
<charParams l="1272" t="639" r="1296" b="689" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="9
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1299" t="637" r="1325" b="689" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="9
charConfidence="100" serifProbability="255">r</charParams>
<charParams l="1330" t="638" r="1354" b="691" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="9
charConfidence="100" serifProbability="255">c</charParams>
<charParams l="1355" t="623" r="1382" b="692" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="9
charConfidence="96" serifProbability="255">k</charParams>
<charParams l="1384" t="636" r="1406" b="688" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="9
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1410" t="634" r="1449" b="688" characterHeight="51" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="9
charConfidence="100" serifProbability="255">n</charParams>
</formatting>
</line>
<line baseline="770" l="421" t="700" r="1271" b="788">
<formatting lang="OldGerman" ff="Arial" fs="19." spacing="-40" style="1">
<charParams l="421" t="700" r="479" b="785" suspicious="true" characterHeight="52" hasUncertainHeight="false" baseLine="0"
wordStart="true" wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1"
meanStrokeWidth="106" charConfidence="98" serifProbability="255">I</charParams>
<charParams l="485" t="703" r="517" b="788" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="10
charConfidence="98" serifProbability="255">h</charParams>
<charParams l="520" t="716" r="580" b="773" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="1" meanStrokeWidth="10
charConfidence="100" serifProbability="255">m</charParams>
</formatting>
<formatting lang="OldGerman" ff="Arial" fs="19." spacing="-20" style="1">
<charParams l="580" t="700" r="590" b="787" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="590" t="717" r="629" b="773" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="634" t="717" r="665" b="770" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">o</charParams>
<charParams l="671" t="700" r="730" b="787" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">c</charParams>
<charParams l="671" t="700" r="730" b="787" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">h</charParams>
<charParams l="730" t="700" r="741" b="770" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="741" t="700" r="767" b="769" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">k</charParams>
<charParams l="769" t="716" r="792" b="768" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="795" t="708" r="811" b="767" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">i</charParams>
<charParams l="814" t="714" r="851" b="769" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">n</charParams>
<charParams l="855" t="715" r="878" b="768" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="98" serifProbability="255">e</charParams>
<charParams l="880" t="709" r="920" b="769" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="3" meanStrokeWidth="10
charConfidence="100" serifProbability="255">s</charParams>
<charParams l="920" t="700" r="933" b="770" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="933" t="701" r="968" b="785" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="true"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">h</charParams>
<charParams l="974" t="716" r="1006" b="767" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="false"
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="100" serifProbability="255">a</charParams>
<charParams l="1011" t="708" r="1032" b="767" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="true" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="10
charConfidence="98" serifProbability="255">t</charParams>
<charParams l="1032" t="701" r="1043" b="782" characterHeight="52" hasUncertainHeight="false" baseLine="0"> </charParams>
<charParams l="1043" t="718" r="1079" b="782" suspicious="true" characterHeight="52" hasUncertainHeight="false" baseLine="
wordStart="true" wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0"
meanStrokeWidth="106" charConfidence="85" serifProbability="255">g</charParams>
<charParams l="1084" t="718" r="1106" b="769" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="1
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1109" t="719" r="1136" b="771" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="1
charConfidence="100" serifProbability="255">r</charParams>
<charParams l="1139" t="719" r="1161" b="771" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="1
charConfidence="100" serifProbability="255">e</charParams>
<charParams l="1164" t="714" r="1219" b="775" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="1
charConfidence="100" serifProbability="255">w</charParams>
<charParams l="1225" t="711" r="1246" b="771" characterHeight="52" hasUncertainHeight="false" baseLine="0" wordStart="fals
wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0" meanStrokeWidth="1
charConfidence="100" serifProbability="255">t</charParams>
<charParams l="1250" t="747" r="1271" b="774" suspicious="true" characterHeight="52" hasUncertainHeight="false" baseLine="
wordStart="false" wordFromDictionary="false" wordNormal="true" wordNumeric="false" wordIdentifier="false" wordPenalty="0"
meanStrokeWidth="106" charConfidence="100" serifProbability="255">.</charParams>
</formatting>
</line>
</par>
</text>
</block>
</page>
</document>
```

75 HK-OCR Export FineReader-XML (Fortsetzung)

HK-XML

Mit Koordinaten auf Wort- statt auf Zeichenebene ist auch ein deutlich kompakteres Format möglich:

```xml
<?xml version="1.0" encoding="utf-8"?>
<IMAGE Name="00000008" ImageX="1710" ImageY="2311">
  <WORD Top="0,07529" Left="0,31053" Right="0,41579" Bottom="0,11164">Lasst/</WORD>
  <WORD Top="0,07702" Left="0,42749" Right="0,51696" Bottom="0,10775">Herr</WORD>
  <WORD Top="0,07789" Left="0,53158" Right="0,7" Bottom="0,11813">Knobloch</WORD>
  <WORD Top="0,08265" Left="0,7117" Right="0,73099" Bottom="0,1151">/</WORD>
  <WORD Top="0,08178" Left="0,74444" Right="0,83392" Bottom="0,11856">Euch</WORD>
  <WORD Top="0,08741" Left="0,85263" Right="0,92749" Bottom="0,1138" LB="True">auch</WORD>
  <WORD Top="0,12289" Left="0,71696" Right="0,84327" Bottom="0,15361">weisen/</WORD>
  <WORD Top="0,12289" Left="0,89474" Right="0,92749" Bottom="0,15707" LB="True">^</WORD>
  <WORD Top="0,14929" Left="0,24503" Right="0,44503" Bottom="0,18693">Sonstmu3</WORD>
  <WORD Top="0,15232" Left="0,45263" Right="0,54152" Bottom="0,18866">Euch</WORD>
  <WORD Top="0,15881" Left="0,54912" Right="0,59708" Bottom="0,18044">ein</WORD>
  <WORD Top="0,15405" Left="0,60702" Right="0,68596" Bottom="0,18347">jeder</WORD>
  <WORD Top="0,1614" Left="0,69532" Right="0,81111" Bottom="0,18563" LB="True">preisen</WORD>
  <WORD Top="0,18823" Left="0,24269" Right="0,3269" Bottom="0,22328">Da3</WORD>
  <WORD Top="0,18953" Left="0,33333" Right="0,38246" Bottom="0,21549">jhr</WORD>
  <WORD Top="0,18866" Left="0,3924" Right="0,45614" Bottom="0,21679">sehr</WORD>
  <WORD Top="0,1891" Left="0,46608" Right="0,63743" Bottom="0,21852">bescheiden</WORD>
  <WORD Top="0,19039" Left="0,65146" Right="0,72047" Bottom="0,22068">seyd</WORD>
  <WORD Top="0,19732" Left="0,72982" Right="0,74211" Bottom="0,22112" LB="True">:</WORD>
  <WORD Top="0,22328" Left="0,24152" Right="0,32807" Bottom="0,2527">Jetzt</WORD>
  <WORD Top="0,22415" Left="0,33626" Right="0,40409" Bottom="0,25314">lasst</WORD>
  <WORD Top="0,23107" Left="0,41345" Right="0,67895" Bottom="0,26222">ewreHoffnung</WORD>
  <WORD Top="0,23453" Left="0,68889" Right="0,84386" Bottom="0,26222" LB="True">mercken/</WORD>
  <WORD Top="0,26049" Left="0,24211" Right="0,38012" Bottom="0,28949">Trawet</WORD>
  <WORD Top="0,26093" Left="0,3883" Right="0,48012" Bottom="0,28949">Gott</WORD>
  <WORD Top="0,27174" Left="0,48596" Right="0,50175" Bottom="0,27564">/^</WORD>
  <WORD Top="0,26785" Left="0,50819" Right="0,57719" Bottom="0,29078">aus</WORD>
  <WORD Top="0,26395" Left="0,5848" Right="0,68187" Bottom="0,29338">dessen</WORD>
  <WORD Top="0,26525" Left="0,68947" Right="0,84211" Bottom="0,29641" LB="True">Wercken</WORD>
  <WORD Top="0,29598" Left="0,24094" Right="0,33333" Bottom="0,32843">Ihm</WORD>
  <WORD Top="0,3042" Left="0,33977" Right="0,42105" Bottom="0,33535">noch</WORD>
  <WORD Top="0,29771" Left="0,42807" Right="0,53216" Bottom="0,32843">keines</WORD>
  <WORD Top="0,29944" Left="0,54035" Right="0,59766" Bottom="0,32843">hat</WORD>
  <WORD Top="0,30723" Left="0,60468" Right="0,73743" Bottom="0,33276" LB="True">gerewt.</WORD>
</IMAGE>
```

76 HK-OCR Export kompaktes XML

7.2 Kurzanleitungen für einfache Fälle
7.2.1 OCR (Muster- und Wortbibliothek müssen vorliegen):

(a) BIT-Alpha (Beispiel mit Musterbibliothek in bda-Datei und lexikalischer Korrektur):

Einzelseite

Vorbereitung:
bda-Datei bereitstellen (für Binarisierungseinstellungen und Muster) (s. Kap. 4.1)
lx2-Datei bereitstellen (für Wortkorrektur) (s. Kap. 4.1)

BIT-Alpha-Sitzung:
File > Open
bda-Datei wählen
(Nur falls in einer bit-Datei gespeicherte Muster verwendet werden sollen:)
OCR > Library > Load
bit-Datei wählen
(Nur falls in einer seq-Datei gespeicherte Sequenzen verwendet werden sollen:)
OCR > Sequences > Load
seq-Datei wählen
OCR > Lexical Correction > Load
lx2-Datei wählen
File > Settings > OCR
aktivieren:
Iterate recognition
Disjunction
Spellcheck
File > Load image
TIFF-Bild wählen
=> OCR beginnt
File > Export
gewünschtes Ausgabeformat wählen

Stapelverarbeitung

Vorbereitung:
Zu bearbeitende Bilder in einem Verzeichnis versammeln
bda-Datei bereitstellen (für Binarisierungseinstellungen und Muster) (s. Kap. 4.1)
lx2-Datei bereitstellen (für Wortkorrektur) (s. Kap. 4.1)

BIT-Alpha-Sitzung:
File > Open
bda-Datei wählen
OCR > Lexical Correction > Load
lx2-Datei wählen
File > Settings > OCR
aktivieren:
Iterate recognition
Disjunction
Spellcheck

File > Batch process ...
Bild-Verzeichnis und Export-Verzeichnis und Export-Formate einstellen (Achtung, „Autolearn“ hier im Regelfall deaktivieren!)
> OK

(b) HK-OCR (Beispiel mit Musterbibliothek und Nutzerwörterbuch):

Einzelseite und Stapelverarbeitung

Vorbereitung:
Zu bearbeitende Bilder in einem Verzeichnis versammeln
Bei der Installation werden die Pfade der Muster- und Sprachdateienverzeichnisse eingestellt, z. B.:

```
<VERZEICHNIS>\UserPattern
<VERZEICHNIS>\CustomLanguageFolder
<VERZEICHNIS>\CustomBaseLanguageFolder
<VERZEICHNIS>\DictionaryFolder
```

In diesen Verzeichnissen Muster- und Sprachdateien bereitstellen (s. Kap. 4.1)

HK-OCR-Sitzung:
Bild öffnen > Ordner wählen
Bildverzeichnis einstellen (ist zugleich Exportverzeichnis)
Reiter „Einstellungen“
Standardsprachen
alle deaktivieren, wenn eigene Sprachengruppe genommen wird
Eigene Sprachengruppen
aktivieren der gewünschten Sprachengruppe (Haken setzen kann 2 Klicks erfordern)
Erkennungsmuster
aktivieren: „Musterdatei verwenden“
deaktivieren: „interne Muster verwenden“
Button „..“: Musterdatei auswählen
Schriften
aktivieren: „Normal“
aktivieren: „Fraktur“
deaktivieren: „Schreibmaschine“
weitere Optionen
aktivieren: „Bild ausrichten“
aktivieren: „verw. vorhandene Layouts“
Qualitätsoptimierung
aktivieren: „Voting“
aktivieren: „Versuchsautomatik“
„4 Versuche max.“
aktivieren: „Automatisch Speichern“
Reiter „Speichern“
> XML, PDF, RTF
Ausgabeformate einstellen, mindestens „FR-XML“
> Nach Speichern
alles deaktivieren
OCR starten
[für Einzelseite:] > aktuelles Bild lesen
[für Stapelverarbeitung:] > Alle Bilder lesen

7.2.2 Training (Anlegen bzw. Erweitern von Musterbibliotheken)

(a) BIT-Alpha

Musterdatei wählen oder erstellen:
(Falls bda-Datei schon existiert:)
File > Open
bda-Datei wählen
(Nur falls in einer bit-Datei gespeicherte Muster verwendet werden sollen:)
OCR > Library > Load
bit-Datei wählen
(Nur falls in einer seq-Datei gespeicherte Sequenzen verwendet werden sollen:)
OCR > Sequences > Load
seq-Datei wählen

Bild laden:
File > Load image
TIFF-Bild wählen
=> OCR-Analyse der Seite beginnt, abwarten

Training
View > Layer > Analyzed
sollte Zeichensegmentierung erkennen lassen (wenn nicht: Binarisierungseinstellungen verbessern)
View > Layer > OCR
zeigt in jedem Zeichensegment entweder das binarisierte Bild der Zeichenregion oder, wenn ein mehr oder weniger passendes Muster gefunden wurde, das „erkannte“ Zeichen in einem farbigen Feld von rot („sehr unsicher“) über orange und gelb zu grün („sichere“ Zuordnung zu einem sehr ähnlichen Muster) bis zu blau (für genau dieses Binärbild ist schon beim Training ein Zeichen gelernt worden).
OCR > Autorefresh

- entweder aktivieren (nach jeder Trainingseingabe wird die Musterbibliothek neu kompiliert, das führt jeweils zu mehreren Minuten Wartezeit, zeigt dann aber im Layer „OCR“ sofort den neuen Erkennungsstand der Seite an)
- oder deaktivieren (Trainingseingaben werden ohne Einkompilierung und ohne Neuanzeige der Seite zwischengespeichert), dafür kann ohne Zeitverzug weitertrainiert werden. Eine Neukompilierung der Musterbibliothek und damit Anzeige-Aktualisierung erreicht man jederzeit durch einen Aufruf:
  OCR > Refresh

Im Layer „OCR“ kann nun frei das nächste zu trainierende Zeichen[101] gewählt werden (mit Maus anklicken). Es erscheint das Trainingsfenster „Verify Recognition“, wo das binäre Zeichenmuster in verschiedenen Ansichten gezeigt wird und im Feld „Recognition result > this String-Value“ der Zeichenwert für dieses Muster eingegeben werden kann. Im Rahmen „Action“ wird durch „Update“ der Zeichenwert dem Muster zugeordnet.

[101] Zweckmäßigerweise wird man zunächst die sehr schlecht erkannten Zeichen (rot und orange unterlegte) oder die noch gar nicht erkannten Zeichen (binäres Image im Segment) wählen.

Speichern der Muster nach Ende des Trainings:

- entweder zusammen mit den Binarisierungs- und Segmentierungsparametern in die bda-Datei:
  File > Save
- oder separat in eine bit-Datei:
  OCR > Library > Save

(b) HK-OCR

Musterdatei wählen oder erstellen:

Reiter „Einstellungen“ > Rahmen „Erkennungs-Muster“ > Button „..“

Bild laden:

Bild öffnen > Ordner wählen > Ordner wählen
In Rahmen „Stapel“ Bilddatei auswählen (z. B. Doppelklick)

Training:

Reiter „Einstellungen“
Rahmen „Erkennungs-Muster“:
Checkbox „Musterdatei verwenden“ deaktivieren
Checkbox „Training“ aktivieren
Rahmen „Schriften“
mindestens eine Checkbox für Schriftfamilie aktivieren (z. B. „Fraktur“)
Rahmen „Standardsprachen“ und/oder „Eigene Sprachengruppen“
passende Sprache wählen (hiervon hängt der mögliche Zeichenvorrat ab)
OCR starten > Aktuelles Bild lesen

Nach Einlesen des Bilds erscheint das Trainingsfenster „Pattern Training“ und springt zum ersten bzw. nächsten Zeichensegment.[102] Im Eingabefeld kann ein Zeichen oder eine Zeichenfolge („ligature“) neu eingetragen bzw. geändert und mit „Train“ bestätigt werden. Wenn bereits das richtige Zeichen vorgeschlagen wurde, kann es mit „Train“ bestätigt oder mit „Skip“ übergangen werden. Muster, die nicht trainiert werden sollen, müssen mit „Skip“ übergangen werden. In begrenztem Maße können die Segmentgrenzen vergrößert oder verkleinert werden (Ziehen mit der Maus oder Buttons „<<„ bzw. „>>„ soweit aktiv).

Speichern der Muster nach Ende des Trainings:

Nach Schließen des Trainingsfensters können die trainierten Muster in die ptn-Datei gespeichert werden, nach Erreichen des Seitenendes geschieht das automatisch (zur Kontrolle ist empfohlen, Datum und aktuelle Dateigröße der ptn-Datei im Musterdateiverzeichnis zu überprüfen).

[102] Das nächste zu trainierende Zeichen kann nicht frei ausgewählt werden; man wird von Anfang bis Ende durch die Seite geführt.

7.3 Details zu einzelnen Parametern

(a) Parametervielfalt BIT-Alpha

Hinweis: Die angegebenen Informationen stammen teils aus dem knappen Handbuch, teils aus der Schulung des Herstellers und teils aus eigenen Tests.

Systemeinstellungen:

Menü File > Settings > System > Thread Priority:

Zweck:	Priorität einstellen, mit der BIT-Alpha den Prozessor belasten darf
Werte:	„Idle“ bis „Time critical“ (sehr zu Lasten aller anderen Anwendungen)
Verwendet:	„Normal“

Einstellung einer bda-Datei:

Empfehlung: ein Image mit mittlerer Helligkeit und mittlerem Kontrast laden, um Parameteränderungen testen zu können.

File > Settings [in empfohlener Reihenfolge der Einstellungen bei Beginn:]

– ganze Seite und erste Segmentierung in Textblöcke betreffend:

Menü File > Settings > Color Enhancement:

Settings

Zweck:	Helligkeit, Kontrast und Gammakorrektur für das Image so einstellen, dass eine optimale Binarisierung erreicht wird.
Orientierung:	Ansicht im Layer „Binary“. Änderungen eines der drei Parameter können Nachjustierungen der anderen erforderlich machen.

– Contrast
verwendete Werte: 0.5–0.6 (Durchschnitt: 0.52, am häufigsten: 0.5)

– Intensity
typischer Wertebereich: 0.4–0.6; verwendet: 0.4–0.51 (Durchschnitt: 0.49, am häufigsten: 0.5)
Wann erhöhen? z. B. wenn Buchstaben zusammenkleben
Wann erniedrigen? z. B. wenn das Bild zu hell ist

– Gamma verwendet: 1

Menü File > Settings > Preprocessing:

Margins

Zweck:	Randbereiche des Images von der OCR ausschließen
Orientierung:	Ansicht in Layer „Binary“: dunkelblauen Rahmen passend einstellen

– left (cm)
verwendete Werte: 0.5–0.7 (Durchschnitt: 0.52, am häufigsten: 0.5)

– right (cm)
verwendete Werte: 0.5–0.7 (Durchschnitt: 0.52, am häufigsten: 0.5)

– top (cm)
  verwendete Werte: 0.5–1.2 (Durchschnitt: 0.71, am häufigsten: 0.7)
– bottom (cm)
  verwendete Werte: 0.5–1.2 (Durchschnitt: 0.72, am häufigsten: 0.7)

Blackborder-Elimination

Orientierung: Ansicht in Layer „Binary“: hellblauen Rahmen passend einstellen

– left (cm)
  verwendete Werte: 0.5–0.7 (am häufigsten: 0.5)
– right (cm)
  verwendete Werte: 0.5–0.7 (am häufigsten: 0.5)
– top (cm)
  verwendete Werte: 0.7–1.2 (am häufigsten: 0.7)
– bottom (cm)
  verwendete Werte: 0.5–1.2 (am häufigsten: 0.7)
– Enabled
  verwendet: in der Regel aktiviert

Binarisation

Zweck: Einstellung der „ersten“ (für die Segmentierung verwendeten) Binarisierung
Algorithmen: „Color based algorithm“, „BIT“, „Intensity based algorithm“, „Modified Niblack algorithm“
Verwendete Auswahl: „Modified Niblack algorithm“ (für Fraktur empfohlen)

– Configure
  > Intensity based algorithm
  > Relative Intensity threshold %
  verwendete Werte: 35–60 (Durchschnitt: 46.94, am häufigsten: 50)
  Wann erhöhen: bei hellem Bild
  Wann erniedrigen: bei dunklem Bild
– „Clean binarised image“
  Zweck: entfernt „Wolkenstaub“;
  Wann deaktivieren: bei sehr „dünnen“ Vorlagen mit schwachem Kontrast
– „Remove dots“
  Zweck: enfernt „größere“ Punkte
  Wann aktivieren: z. B. bei Mikrofilm; rechnerintensiv
– min. dist. cm
  verwendete Werte: 0.02–0.09 (Durchschnitt: 0.05, am häufigsten: 0.04)
– max. size cm
  verwendete Werte: 0.05–0.15 (Durchschnitt: 0.10, am häufigsten: 0.09)

Rotation

Zweck: Seitenausrichtung anhand erkannter Linien
Algorithmen: „Linear Algorithm“ (versucht, Rechtwinkligkeit wiederherzustellen), „Circular Algorithm“ (Rotation, Winkel bleiben erhalten)
Verwendet: „Circular Algorithm“ und ohne Rotation

– min. angle (degrees)
  verwendete Werte: 0.01

Menü File > Settings > Lines V

Limits (im Image gespeicherte Maßeinheiten und Auflösung müssen korrekt sein)

Zweck: Entfernung von vertikalen Linien

– max. thickness cm
  verwendete Werte: 0.1–0.8 (Durchschnitt: 0.54, am häufigsten: 0.8)
– min. thickness cm
  verwendete Werte: 0.01
– min. black length cm
  Zweck: minimale Länge eines gedruckten „Zeichens“ in eventuell unterbrochenen Linien
  verwendete Werte: 0.4–0.9 (Durchschnitt: 0.54, am häufigsten: 0.4)
– max. white length cm
  Zweck: maximaler weißer Zwischenraum in eventuell gepunkteten oder unterbrochenen Linien
  verwendete Werte: 0
– min. total length cm
  Zweck: minimale totale Länge, um zu definieren, ab wann eine Linie als solche erkannt werden soll, die eventuell gepunktet oder unterbrochen ist
  typischer Wertebereich: -3
  verwendete Werte: 0.75–10 (Durchschnitt: 5.24, am häufigsten: 10)
  Wann erhöhen: + wenn „falsche Linien“ (z. B. aus übereinander stehenden m-Füßen) erkannt werden
– Use bitmap
  Zweck: Gescanntes Bild der Linie wiedergeben, statt sie ideal nachzuzeichnen
  verwendete Werte: Aktiviert

Menü File > Settings > Lines H

Limits (im Image gespeicherte Maßeinheiten und Auflösung müssen korrekt sein)

Zweck: Entfernung von horizontalen Linien zur besseren Absatztrennung (aber nicht erwünscht, wenn keine Linien da sind)

– max. thickness cm
  Zweck: maximale Breite des Strichs der Linie
  verwendete Werte: 0.1–0.5 (Durchschnitt: 0.36, am häufigsten: 0.4)
– min. thickness cm
  Zweck: minimale Breite des Strichs der Linie
  verwendete Werte: 0
– min. black length cm
  Zweck: minimale Länge eines gedruckten „Zeichens“ in eventuell unterbrochenen Linien
  verwendete Werte: 0.4–0.9 (Durchschnitt: 0.59, am häufigsten: 0.5)

- max. white length cm
  Zweck: maximaler weißer Zwischenraum in eventuell gepunkteten oder unterbrochenen Linien
  verwendete Werte: 0.1
- min. total length cm
  Zweck: minimale totale Länge, um zu definieren, ab wann eine Linie als solche erkannt werden soll, die eventuell gepunktet oder unterbrochen ist
  verwendete Werte: 0.75–10 (Durchschnitt: 5.13, am häufigsten: 10)
- Use bitmap
  Zweck: Gescanntes Bild der Linie wiedergeben, statt sie ideal nachzuzeichnen
  verwendete Werte: Aktiviert

Menü File > Settings > Segmentation

Parameters (im Image gespeicherte Maßeinheiten und Auflösung müssen korrekt sein)

Zweck: Einstellungen für die Segmentierung der Seite in verschiedene Regionen, d. h. Paragraphen, Titel, Untertitel und graphische Elemente

- min. horiz. distance cm
  Zweck: die minimal erlaubte horizontale Distanz zwischen benachbarten Regionen. Näher beieinander liegende Regionen werden zusammengefügt.
  verwendete Werte: 1–2 (Durchschnitt: 1.83, am häufigsten: 2)
  Praxiserfahrungen: Bei zunächst deaktivierter „Dissection“ justieren, bis in der „Binary“-Ansicht der prozessierten Seite die grauen Teilrahmen korrekt sind (z. B. Satzspiegel)
- min. vert. distance cm
  Zweck: die minimal erlaubte vertikale Distanz zwischen benachbarten Regionen. Näher beieinander liegende Regionen werden zusammengefügt.
  verwendete Werte: 0.5–2 (Durchschnitt: 1.69, am häufigsten: 2)
- min. width cm
  Zweck: die minimal erlaubte horizontale Breite für die Erstellung einer Region.
  verwendete Werte: 0.1
- min. height cm
  Zweck: die minimal erlaubte vertikale Höhe für die Erstellung einer Region.
  verwendete Werte: 0.1
- Default Region Type
  Zweck: BIT-Alpha teilt jeder bei der Segmentierung kreierten Region Eigenschaften über den Typ des Bilds in der Region zu.
  Werte: „Binary Region“: monochromes Bild (Schwarz-weiß), „Palette Region“: Graustufenbild, jeder Pixel mit 8 bit kodiert, „color region“: Farbbild, jeder Pixel mit 8 oder 16 bit kodiert
  verwendete Werte: „Binary Region“

File > Settings

– betreffend Segmentierung in Unterblöcke:

Menü File > Settings > Segmentation

Dissection

Zweck: betrifft zusätzlich Unterteilung nach Abständen zwischen Textblöcken; Teilblöcke werden dann automatisch separat eingestellt

Praxiserfahrungen: Justieren, bis in der Binary-Ansicht der prozessierten Seite die grauen Teilrahmen korrekt sind (z. B. Absätze, Strophen ...). Nachjustieren kann Abhilfe schaffen, wenn in manchen Textblöcken keine Zeichen-Segmentierung erfolgt ist.

– min. width cm
  Zweck: Versuch, ab dieser Breite den Textblock zu teilen
  verwendete Werte: 5-10 (Durchschnitt: 7.94, am häufigsten: 9)
– min. height cm
  Zweck: Versuch, ab dieser Höhe den Textblock zu teilen
  verwendete Werte: 4–10 (Durchschnitt: 5.00, am häufigsten: 4)
– enable hor. dissection
  Zweck: versucht, entlang „waagerechter weißer Streifen“ zu teilen
  verwendete Werte: Aktiviert
– hor. dissection threshold % (Prozent-Anteil weiß im Zwischenraum)
  verwendete Werte: 95–99 (Durchschnitt: 97.83, am häufigsten: 98)
– hor. dissection thickness cm (Mindestbreite weiß)
  verwendete Werte: 0.01-0.3 (Durchschnitt: 0.13, am häufigsten: 0.2)
– enable vert. dissection
  Zweck: versucht, entlang „senkrechter weißer Streifen“ zu teilen
  verwendete Werte: Aktiviert
– vert. dissection threshold %
  verwendete Werte: 95-99 (Durchschnitt: 98.11, am häufigsten: 99)
– vert. dissection thickness cm
  verwendete Werte: 0.01-0.5 (Durchschnitt: 0.24, am häufigsten: 0.2)
  Wann verändern: Wenn Textblöcke nicht als solche erkannt werden und „Image detection“ nicht deaktiviert werden soll

Resolution

Zweck: Angabe der korrekten Auflösung, falls in den Image-Metadaten nicht korrekt enthalten

verwendete Werte: Aktiviert

Image detection

Zweck: Erkennung von Bildregionen (Illustrationen usw.), die von der OCR ausgenommen werden sollen

Wann deaktivieren: wenn Textblöcke fälschlich als Bild interpretiert werden (keine Zeichensegmentierung in Ansicht „Analyzed“) und eine Änderung der „vert. dissection thickness cm“ keine Abhilfe bringt

- enabled
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)
- hor.
  verwendete Werte: 0.45-1 (Durchschnitt: 0.57, am häufigsten: 0.45)
- vert.
  verwendete Werte: 0.45-1 (Durchschnitt: 0.57, am häufigsten: 0.45)

File > Settings

- **betreffend Zeichenregionen:**

Menü File > Settings > Binary regions

Detection

- Detect binary regions
  Zweck: Unterscheidung von Zeichen (für OCR) und Bildregionen (für bildliche Wiedergabe)
  Hinweis: Parametrierung unnötig, wenn in Preprocessing > Segmentation > Default Region Type „Binary Region“ eingestellt wurde
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)

Binarisation

Praxiserfahrungen: In Ansicht „Analysed“ und „OCR“ auf gute Zeichensegmentierung achten und darauf, dass keine Textblöcke aus der Zeichensegmentierung herausfallen

Algorithmen: „Color based algorithm“
„BIT-Algorithmus“
„Intensity based algorithm“
„Modified Niblack algorithm“
– Konfigurationsparameter siehe unten –

- Clean binarised image
  Empfehlung: Deaktivieren bei sehr dünnen (aus „Punktwolken“ bestehenden) Schriftzeichen
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)
- Remove dots:
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)
- min. dist. cm
  verwendete Werte: 0.02-0.13 (Durchschnitt: 0.07, am häufigsten: 0.05)
- max. size cm
  verwendete Werte: 0.05-0.25 (Durchschnitt: 0.12, am häufigsten: 0.08)

Binarisation > Color based algorithm

verwendete Werte: Deaktiviert

Menü File > Settings > Binary regions [Forts.]

Binarisation > BIT-Algorithmus > Configure

Hinweis: vom Hersteller für typische Drucke empfohlen; weniger für Zeitungen und Fraktur
verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Deaktiviert)

- Minimum > horiz.
  verwendete Werte: 0.01-0.025 (am häufigsten: 0.02)
  Wann erhöhen: bei blassem Druck
- Minimum > vert.
  verwendete Werte: 0.01-0.025 (am häufigsten: 0.02)
- Maximum > horiz.
  verwendete Werte: 2-4 (am häufigsten: 2)
  Wann erhöhen: bei blassem Druck
- Maximum > vert.
  verwendete Werte: 2-4 (am häufigsten: 2)
- Binarisation > cut-off
  verwendete Werte: 0.4-0.53 (am häufigsten: 0.5)
  Wann erhöhen: bei blassem Druck (wenn Buchstaben dunkler werden sollen)
  Wann erniedrigen: gegen Durchdruck

Binarisation > Intensity based algorithm > Configure

verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Deaktiviert)

- Relative Intensity threshold %
  typischer Wertebereich: 50-70
  verwendete Werte: 50, 61 (am häufigsten: 50)
  Wann erhöhen: wenn schwacher Kontrast
  Wann erniedrigen: wenn Zeichen zusammenkleben (zu dunkel)

Binarisation > Modified Niblack algorithm > Configure

Hinweis: nutzt dynamischen Kontrast (Varianz der Helligkeit) um jeden Buchstaben herum, erkennt eher Kontur als „schwere“ Innenbereiche; für Fraktur empfohlen
verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)

- Noise reduction:
  - Enable noise reduction
    verwendete Werte: Aktiviert
  - Vertical noise frequency (mm):
    verwendete Werte: 0.25-0.5 (am häufigsten: 0.5)
  - Horizontal noise frequency (mm):
    verwendete Werte: 0.25-0.5 (am häufigsten: 0.5)
- Configuration:
  - Vertical adjustment radius (cm):
    verwendete Werte: 0.2-0.25 (am häufigsten: 0.25)
  - Horizontal adjustment radius (cm):
    verwendete Werte: 0.2-0.25 (am häufigsten: 0.25)
  - Noise threshold (%):
    verwendete Werte: 9-10 (am häufigsten: 10)
    Wann erhöhen: wenn zu dunkel
    Wann erniedrigen: wenn Zeichen brüchig sind
  - Noise factor:
    verwendete Werte: 1

Menü File > Settings > OCR

Target Size

Zweck: minimale und maximale akzeptierte Zeichengröße

– min. width (cm)
verwendete Werte: 0.03–0.05 (Durchschnitt: 0.04, am häufigsten: 0.03)
Wann erniedrigen: wenn Text fälschlich als Miniatur erkannt wird
Praxiserfahrungen: in Ansicht „OCR“ so groß wie ohne Lücken möglich einstellen

– max. width (cm)
verwendete Werte: 0.035–6 (Durchschnitt: 3.84, am häufigsten: 4)
Hinweis: ggf. an Initialen denken

– min. height (cm)
verwendete Werte: 0.03–0.05 (Durchschnitt: 0.04, am häufigsten: 0.03)

– max. height (cm)
verwendete Werte: 3-4 (Durchschnitt: 3.83, am häufigsten: 4)
Hinweis: ggf. an Initialen denken

Options

Hinweis: Optionen primär für OCR-Lauf sinnvoll; im Training dagegen diese zunächst deaktivieren! Nach dem ersten Training kann erst mit „Iterate“, dann mit „Iterate+Disjunction“ das Training wiederholt werden.

– Iterate recognition
Zweck: entfernt z. B. Punkte bei nicht erkannten Zeichen.
verwendete Werte (im OCR-Lauf): Aktiviert

– Disjunction
Zweck: Trennung von zusammengeklebten Charakteren für die OCR-Bearbeitung
verwendete Werte (im OCR-Lauf): Aktiviert, Deaktiviert (am häufigsten: Aktiviert)

– Spellcheck
Hinweis: Sprachkorrektur per Wortliste für die OCR-Bearbeitung aktivieren (s. „Lexika“)
verwendete Werte (im OCR-Lauf): Aktiviert, Deaktiviert (am häufigsten: Aktiviert)

Menü File > Settings > OCR > Advanced Settings > Edit
(muss für jede Schriftfamilie separat eingestellt werden)

Target Size

– min. width (cm)
verwendete Werte: 0.03-0.05 (am häufigsten: 0.04)

– max. width (cm)
verwendete Werte: 3-6 (am häufigsten: 4)

– min. height (cm)
verwendete Werte: 0.03-0.05 (am häufigsten: 0.04)

– max. height (cm)
verwendete Werte: 3-4 (am häufigsten: 4)

Menü File > Settings > OCR > Advanced Settings > Edit [Forts.]

Miniatures

Zweck: Erkennung von Initialen

– Min. Overlap
  verwendete Werte: 0.02–0.3333 (Durchschnitt: 0.12, am häufigsten: 0.05)
– min. X (rel.)
  verwendete Werte: 3–9 (Durchschnitt: 5.78, am häufigsten: 6)
– min. Y (rel.)
  verwendete Werte: 3–9 (Durchschnitt: 5.78, am häufigsten: 6)

Classification

Zweck: je nach Variabilität der Schriftform[103], relativ zur Minuskelhöhe einstellbare Größenverhältnisse von: [P]atte (senkrechtes Bein bzw. kleiner senkrechter Strich, z. B. i ohne Punkt), [M]inuscule (Kleinbuchstabe), [G]rand (Großbuchstabe bzw. großer Strich, [R]este (kleineres Element als Patte, Minuscule oder Grand, z. B. „.“ oder „-“)

– Minuscule min. X
  verwendete Werte: 0.3-0.5 (Durchschnitt: 0.4, am häufigsten: 0.4)
– Minuscule min. Y
  verwendete Werte: 0.5-0.8 (Durchschnitt: 0.71, am häufigsten: 0.7)
– Minuscule max. Y
  verwendete Werte: 1.05-1.2 (Durchschnitt: 1.11, am häufigsten: 1.1)
– Collation min. Y
  Zweck: Abschneiden von über die Gesamtgröße hinausgehenden Zeichen
  verwendete Werte: 1.8-2.5 (Durchschnitt: 2.03, am häufigsten: 2)
  Praxiserfahrungen: Heraufsetzen kann Abhilfe schaffen, wenn in Bereichen keine Zeichensegmentierung erfolgte.

Lines

Hinweis: muss für jede Schriftfamilie separat eingestellt werden
Zweck: Zeileneigenschaften
Hinweis: In Menü und Handbuch wird „Line“ teils für „Zeile“, teils für „Linie“ verwendet.

– Topline min.
  verwendete Werte: 0
– Bottomline min.
  verwendete Werte: 0.25-0.5 (Durchschnitt: 0.31, am häufigsten: 0.27)
– Topline max.
  verwendete Werte: 0.25-0.6 (Durchschnitt: 0.37, am häufigsten: 0.35)
– Bottomline max.
  verwendete Werte: 0.3-0.6 (Durchschnitt: 0.36, am häufigsten: 0.3)
– Rigidity
  Zweck: Starrheit der (Zeilen-)Linienführung
  verwendete Werte: 5-9 (Durchschnitt: 6.44, am häufigsten: 7)
  Wann erhöhen: + um weniger Krümmung zu erlauben
  Wann erniedrigen: – um mehr Krümmung zu erlauben

[103] Die Parametrierung dieses Menüs wird grundsätzlich von B.I.T. selbst und nicht vom Benutzer vorgenommen. Die Einstellungen sind für Fraktur, Kursivschrift der Renaissance oder gerade Antiqua unterschiedlich.

- Min. Limit
  verwendete Werte: 0.3-0.9 (Durchschnitt: 0.63, am häufigsten: 0.8)
- Max. Limit
  verwendete Werte: 1-1.5 (Durchschnitt: 1.33, am häufigsten: 1.5)
- Large Smear
  Zweck: Berechnung der Zeilenerkennung mittels „Verwischen“ der Großbuchstaben
  verwendete Werte: 3-7 (Durchschnitt: 4.67, am häufigsten: 5)
  Praxiserfahrungen: Erniedrigen kann Abhilfe schaffen bei „verschluckten“ Zeilen
- Small Smear
  Zweck: Berechnung der Zeilenerkennung mittels „Verwischen“ der Kleinbuchstaben
  verwendete Werte: 0.3-0.5 (Durchschnitt: 0.41, am häufigsten: 0.5)
- Merge Distance
  Zweck: Einstellung, ab wann Zeilen als separat gesehen werden sollen
  verwendete Werte: 0.75-3 (Durchschnitt: 1.53, am häufigsten: 1)
- Merge Overlap
  verwendete Werte: 0.3-0.9 (Durchschnitt: 0.56, am häufigsten: 0.9)
- Up-Factor
  verwendete Werte: 0.13-0.5 (Durchschnitt: 0.24, am häufigsten: 0.2)
- Down-Factor
  verwendete Werte: 0.15-0.5 (Durchschnitt: 0.22, am häufigsten: 0.17)
- Horiz. Locality
  verwendete Werte: 2-4 (Durchschnitt: 3.61, am häufigsten: 4)
- Vert. Locality
  verwendete Werte: 1-2 (Durchschnitt: 1.83, am häufigsten: 2)

R-R (Abstand bei Kombination der Klassen „Reste“-“Reste“)

- Horizontal
  verwendete Werte: 0.05-0.3 (Durchschnitt: 0.12, am häufigsten: 0.1)
- Vertical
  verwendete Werte: 0.2-0.9 (Durchschnitt: 0.72, am häufigsten: 0.9)

R-P (Abstand bei Kombination der Klassen „Reste“-“Patte“)

- Horizontal
  verwendete Werte: 0.1-0.2 (Durchschnitt: 0.13, am häufigsten: 0.1)
- Vertical
  verwendete Werte: 0.7-1.5 (Durchschnitt: 0.98, am häufigsten: 0.9)

R-M (Abstand bei Kombination der Klassen „Reste“-“Minuscule“)

- Horizontal
  verwendete Werte: 0.05-0.2 (Durchschnitt: 0.12, am häufigsten: 0.1)
- Vertical
  verwendete Werte: 0.7-1.5 (Durchschnitt: 0.98, am häufigsten: 0.9)

R-G (Abstand bei Kombination der Klassen „Reste“-“Grand“)

- Horizontal
  verwendete Werte: 0.1-0.2 (Durchschnitt: 0.13, am häufigsten: 0.1)
- Vertical
  verwendete Werte: 0.4-1.5 (Durchschnitt: 1.02, am häufigsten: 0.9)

Menü File > Settings > OCR > Advanced Settings > Edit [Forts.]

M-M (Abstand bei Kombination der Klassen „Minuscule“-“Minuscule“)

– Horizontal
  verwendete Werte: 0.02-0.1 (Durchschnitt: 0.04, am häufigsten: 0.03)
– Vertical
  verwendete Werte: 0.9-1.5 (Durchschnitt: 0.98, am häufigsten: 0.9)

P-P (Abstand bei Kombination der Klassen „Patte“-“Patte“)

– Horizontal
  verwendete Werte: 0.05-0.2 (Durchschnitt: 0.12, am häufigsten: 0.1)
– Vertical
  verwendete Werte: 0.5-0.9 (Durchschnitt: 0.85, am häufigsten: 0.9)

M-P (Abstand bei Kombination der Klassen „Minuscule“-“Patte“)

– Horizontal
  verwendete Werte: 0.05-0.15 (Durchschnitt: 0.09, am häufigsten: 0.1)
– Vertical
  verwendete Werte: 0.5-0.9 (Durchschnitt: 0.88, am häufigsten: 0.9)

G-P (Abstand bei Kombination der Klassen „Grand“-“Minuscule“)

– Horizontal
  verwendete Werte: 0.05-0.2 (Durchschnitt: 0.08, am häufigsten: 0.05)
– Vertical
  verwendete Werte: 0.5-5 (Durchschnitt: 1.29, am häufigsten: 0.9)

G-G (Abstand bei Kombination der Klassen „Grand“-“Grand“)

– Horizontal
  verwendete Werte: 0.02-0.07 (am häufigsten: 0.05)
– Vertical
  verwendete Werte: 0.9-2 (Durchschnitt: 1.13, am häufigsten: 0.9)

M-G (Abstand bei Kombination der Klassen „Minuscule“-“Grand“)

– Horizontal
  verwendete Werte: 0.02-0.05 (Durchschnitt: 0.04, am häufigsten: 0.05)
– Vertical
  verwendete Werte: 0.9-2 (Durchschnitt: 1.13, am häufigsten: 0.9)

Menü File > Settings > Export

PDF

– only text
  Zweck: Bild nicht mitexportieren
  verwendete Werte: Deaktiviert
– one single image ebedded per page
  Zweck: Nur ein Bild pro Seite (Kopie des Original-Faksimiles) einbetten. Der per OCR gelesene Text ist für Textsuche und Highlighting als unsichtbares Hintergrundbild hinterlegt.
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Deaktiviert)

– one image embedded for each segmented region
  Zweck: Jede bei der Segmentierung der Seite erstellte Region auch als Bild einbetten (kann dann separat aus dem PDF geöffnet werden).
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)
– Create high-resolution links for text regions
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Deaktiviert)

METS/ALTO

– relative to the original image
  Zweck: Koordinaten auf das originale Bild beziehen (sinnvoll, wenn dieses und nicht auf das zur OCR ggf. gedrehte Bild präsentiert wird)
  verwendete Werte: Aktiviert

OCR

OCR > Library > Learning parameters ...

Learning configuration

– min.:
  typischer Wertebereich: 0.55 bei „use Spellcheck“, 0.8 ohne Spellcheck
– max.:
  typischer Wertebereich: 0.75 bei „use Spellcheck“, 0.9 ohne Spellcheck
– use Spellcheck
  verwendete Werte: Deaktiviert

OCR > Library > Settings...

Recognition

– mass-threshold
  verwendete Werte: 0.12-0.2 (Durchschnitt: 0.15, am häufigsten: 0.14)
– position-threshold
  verwendete Werte: 0.12-0.16 (Durchschnitt: 0.14, am häufigsten: 0.14)
– stretching-threshold
  verwendete Werte: 0.12-0.3 (am häufigsten: 0.14)
– density-threshold
  verwendete Werte: 0.12-0.2 (Durchschnitt: 0.15, am häufigsten: 0.14)
– confidence-threshold (nur in bestimmten BIT-Alpha-Versionen)
  verwendete Werte: 0.7-0.8 (Durchschnitt: 0.78, am häufigsten: 0.8)
– blanks low-limit
  verwendete Werte: 0.05-0.25 (Durchschnitt: 0.16, am häufigsten: 0.2)
– blanks high-limit
  verwendete Werte: 0.6-1.5 (Durchschnitt: 0.78, am häufigsten: 0.6)
– blanks coefficient
  verwendete Werte: 0.6-1.2 (Durchschnitt: 0.97, am häufigsten: 1)
  Wann erhöhen: wenn mehr zusammengehalten werden soll
  Wann erniedrigen: wenn mehr getrennt werden soll

OCR > Library > Settings... [Forts.]

Separation

- min. confidence
  verwendete Werte: 0.7-0.75 (Durchschnitt: 0.73, am häufigsten: 0.75)
- min. confidence
  verwendete Werte: 0.6-0.75 (Durchschnitt: 0.67, am häufigsten: 0.65)
- min.width
  verwendete Werte: 0.4-0.5 (Durchschnitt: 0.49, am häufigsten: 0.5)
- limit (left)
  verwendete Werte: 0.2-0.3 (Durchschnitt: 0.22, am häufigsten: 0.2)
- limit (right)
  verwendete Werte: 1.2-1.7 (Durchschnitt: 1.44, am häufigsten: 1.5)
- min. (right)
  verwendete Werte: 0.15-0.2 (Durchschnitt: 0.19, am häufigsten: 0.2)
- min. thickness
  verwendete Werte: 0.1-0.5 (am häufigsten: 0.4)
- min. complexity (left)
  verwendete Werte: 1
- min. complexity (right)
  verwendete Werte: 1

Algorithm

- by form
  verwendete Werte: Aktiviert
- by mass
  Zweck: empfohlen z. B. bei chinesischen Zeichen

OCR > Lexical Correction > Edit

Coefficients:

- Default:
  verwendete Werte: 1-1.51 (am häufigsten: 1)
- Split:
  verwendete Werte: 0.70-1 (am häufigsten: 1)
  Wann erhöhen: Trennung erschweren (auf 2 oder 3 um Trennungen zu verhindern)
  Wann erniedrigen: Trennung erleichtern
- Symbol added/deleted:
  verwendete Werte: 1-2.1 (am häufigsten: 1)
- (Text-area)
  Zweck: Für einzelne Zeichenpaare gewichtete „Ersetzungskosten“ einstellen (0 = Ersetzung ungehindert; 1 = keine Ersetzung)

Signature:

Zweck: wie oft ein Wort korrigiert werden kann

- [Entry]
  verwendete Werte: 2 oder 3 (am häufigsten: 2)
- Disable correction for words starting with capital letters
  verwendete Werte: Deaktiviert

Correction:

- max. distance:
  verwendete Werte: 0.2-0.5 (Durchschnitt: 0.35, am häufigsten: 0.20)
  Wann erhöhen: wenn großzügiger ersetzt werden soll
  Wann erniedrigen: wenn zurückhaltender ersetzt werden soll

Split mode:

- Recurse fragments
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Deaktiviert)
- Split fragments
  verwendete Werte: Aktiviert, Deaktiviert (am häufigsten: Aktiviert)
- No fragmentation
  verwendete Werte: Deaktiviert

(b) HK-OCR / FineReader: Gibt es Potentiale durch die Erschließung von ABBYY-seitig angelegten Konfigurationsmöglichkeiten?

Über die Oberfläche von HK-OCR sind im wesentlichen nur die in den Kapiteln 4 und 5 beschriebenen Parameter konfigurierbar.

Für ein Interesse an eventuell weiterführenden, in HK-OCR derzeit noch nicht realisierten Konfigurationsmöglichkeiten der FineReader-Engine muss auf einschlägige ABBYY-Entwickler-Dokumentationen, speziell die darin enthaltenen Objekt-Spezifikationen verwiesen werden.

7.4 Einige Mindestanforderungen an Bedieneroberflächen

Man muss die inzwischen kaum noch überschaubare Menge von Normen, Styleguides und Empfehlungen zu Softwareergonomie[104] keineswegs systematisch durcharbeiten, um zumindest folgende (systematisch meist unter Steuerbarkeit und Erwartungskonformität einzuordnende) Anforderungen für plausibel zu halten. Sie spielten schon im Projektverlauf eine Rolle und können im OCR-Produktionseinsatz erst recht zum kritischen Faktor werden:

– Jede Arbeitssituation muss jederzeit konsistent beendet werden können.

Konfigurations-, Muster-, Lexikon- und alle anderen Dateien sollten jederzeit entladen werden können (Fortsetzung der Sitzung erkennbar mit den jeweiligen Default-Werten), auch ohne dass dazu eine neue entsprechende Datei geladen werden muss. (Die Anzeige der aktuellen Parameterdateinamen in Titel- oder Statuszeile muss dem natürlich folgen).

Besonders während Verarbeitungsschritten mit Zeitbedarf (häufig während Layoutanalyse und OCR) muss eine Möglichkeit zum sofortigen Abbruch angeboten werden.
Bei Abbrüchen von Stapelläufen muss es eine einfache Möglichkeit zur Fortsetzung am Abbruchpunkt geben, ohne dass Verzeichnisinhalte umorganisiert werden müssen.

– Jede Arbeitssituation muss auf den ersten Blick erkennbar sein.

Die Namen aller aktuell geladenen bzw. wirksamen Parameter-, Image- und sonstigen Dateien sollten dem Bearbeiter jederzeit ohne weitere Handgriffe, Mausaktionen, Suchschritte vor Augen stehen, z. B. in der Titelzeile und/oder der Statuszeile.

Es muss erkennbar sein, ob und wann manuelle Parameteränderungen in einer Konfigurationsdatei gespeichert werden, d. h. diese verändern (was man in der aktuellen Sitzung nicht immer will).
Benutzeraktionen sollen stets (auch) über beschriftete Menüs erreichbar sein, nicht allein durch einen ikonographischen Button, dessen Bedeutung erst erlernt werden muss oder erst nach Mausaktionen angezeigt wird.

Aktionen, die der Anwender wahlweise anstoßen möchte oder nicht, sollten je explizit durch einen passenden Menübefehl ausgelöst werden und nicht implizit in einem unerwarteten Zusammenhang beginnen.

In Dateiauswahl-Dialogen muss statt der Verwendung des mehrdeutigen Labels „Öffnen“ jeweils erkennbar sein, ob eine gewählte Datei (a) zum Lesen bzw. Betrachten, (b) zum Bearbeiten oder (c) zum Ausführen darin enthaltenen Codes geladen wird.

– Safety first: Bedienelemente müssen eine sichere Arbeitsweise erleichtern und unsichere Aktionen erschweren.

In keiner Situation darf die riskantere Aktion leichter fallen als die sicherere Aktion. In Editier-Umgebungen muss das Übergehen bzw. Verwerfen einer Eingabe (per Tasten oder Button-Fokus) mindestens genauso schnell erreichbar sein wie die Bestätigung einer Eingabe, die etwas verändert.

Für Situationen, in denen aus Dateien, Datensätzen usw. nur passiv gelesen werden soll, dürfen zum „Öffnen“ der Datei bzw. des Datensatzes keine Komponenten mit Schreibmöglichkeit verwendet werden.

– Bedienelemente sollen eine schnelle Arbeitsweise ermöglichen.

Um sofort in einer bewährten Arbeitsumgebung beginnen zu können, sollte es möglich sein zu wählen, ob Fenstergröße und -position sowie die aktuellen Verzeichnisse der letzten Sitzung wiederverwendet werden oder nicht.

Parameterdateinamen, Bilddateiliste, Ausgabeverzeichnis usw. sollten auch bereits als Programmparameter beim Aufruf übergeben werden können, um (in Batchdateien bzw. Shell-Skripten oder Desktop-Icons usw.) passende Aufrufe für Sitzungen zu ermöglichen.

[104] Ein kompakter Überblick mit weiterführenden Quellenangaben findet man z. B. in RUDLOF 2006; Normen u. a.: EN ISO 9241

Für häufige Aktionen sollte außer einem Maus-Button wenn möglich immer auch eine Taste angeboten werden, möglichst in Anlehnung an allgemein übliche, z. B. für „Refresh“ so etwas wie [F5] oder [Ctrl]+[R], für „weiter“ so etwas wie [Tab] oder Cursortasten usw.

Solange mit beiden Tastaturhälften gearbeitet werden muss, ist der Wechsel zur Maus und zurück immer ein Zeitverlust. Und umgekehrt: In Situationen, wo die Mausbedienung effektiver oder notwendig ist, sollte nach Möglichkeit nur eine Hand ergänzend auf der Tastatur nötig sein.

Bedienelemente sollen sowohl örtlich als auch hinsichtlich des Fokuswechsels in einer der Bearbeitung entsprechenden Reihenfolge angeordnet sein. Für absehbare Schrittfolgen sollte die [Enter]- oder [Tab]-Taste zum nächsten Schritt führen.

Listen von Einträgen müssen zum schnellen Auffinden eines Eintrags jederzeit in nachvollziehbaren Ordnungen sortierbar sein. Lange Listen sollten eine Mehrfachselektion ermöglichen, um Gruppen von Einträgen zusammen behandeln zu können.

Das erwünschte Speichern von kleinen Parameteränderungen soll nicht dadurch erschwert werden, dass unnötig viel anderes jeweils mitgespeichert werden muss – eine störend lange Speicherzeit verleitet zum Verzicht auf rechtzeitiges Sichern von Einstellungen und führt so leicht zum Verlust bereits erarbeiteter Konfigurationen.

– **Navigation einzuschränken hat nur Sinn, wenn dadurch nicht zeitaufwendige Umwege für absehbar notwendige Arbeitsgänge entstehen.**

Die Geometrie der Oberfläche sollte niemals dazu führen, sinnvolle Aktionen für unmöglich oder zu aufwendig zu halten.

– **Eine schnelle und fehlerfreie Navigation erfordert Anlehnung an die vom Betriebssystem gewohnten Standard-Abläufe.**

Dateiauswahldialoge sollten an einem nachvollziehbaren, aber bequemen Ort beginnen, z. B. im Verzeichnis der aktuell an dieser Stelle geltenden Auswahl – d. h. nicht stets in der Wurzel des Dateisystems und nicht in einem irgendwann früher benutzten, aber aktuell nicht geltenden Ordner.

In Speicherdialogen zu einer bereits geöffneten Datei wird ein editierbarer Vorschlag der Speicherung am bisherigen Ort unter bisherigem Dateinamen, so wie man das als Standardverhalten von Anwendungen kennt, erwartet.

Dialoge zur Dateiauswahl und -speicherung sollten stets in einer echten Combobox bereitgestellt werden, in der alternativ zum „Durchklicken“ der Pfad auch als kompletter String ins Textfeld kopiert werden kann.

Eine routinemäßige Speicherabfrage („Soll .. gespeichert werden?“) sollte nur dann angeboten werden, wenn tatsächlich Einstellungen geändert wurden (nur dann hätte die Abfrage auch eine sinnvolle Hinweisfunktion darauf, dass man in der Sitzung etwas geändert hat, und man hätte Anlass zu überlegen, ob versehentlich oder nicht).

7.5 Literatur- und Linkliste

Im Folgenden wird sowohl zitierte als auch weiterführende Literatur angegeben. Für systematischen bzw. weiterführenden Informationsbedarf wird auf die Literaturangaben in den genannten Publikationen verwiesen.

Materialspezifik der Funeralschriften; Frakturschrift

FRACTUR 1808
Die Fractur
in: Adelung, Johann Christoph: Grammatisch-kritisches Wörterbuch der hochdeutschen Mundart.
Wien: Pichler. Bd. 2. 1808. S. 261
FRAKTUR 1954
Fraktur
in: Der große Brockhaus. 16., völlig neubearb. Aufl. Wiesbaden : Brockhaus. Bd. 4. 1954. S. 194
HORN 1894
Horn, E.: Zur Orthographie von U und V, I und J. Eine historisch typographische Erörterung
in: Zentralblatt für Bibliothekswesen 11(1894), S. 385-400
JENSEN 1969
Jensen, Hans: Die Schrift in Vergangenheit und Gegenwart. 3., neubearb. u. erw. Aufl. Berlin : Dt. Verl. der Wiss., 1969. 607 S.
KAPR 1959
Kapr, Albert: Deutsche Schriftkunst. Versuch einer neuen historischen Darstellung. 2., verb. Aufl. - Dresden : Verl. der Kunst, 1959. 287 S.
KAPR 1993
Kapr, Albert: Fraktur. Form und Geschichte der gebrochenen Schriften. Mainz : Schmidt, 1993. 248 S.
LENZ 1989
Lenz, Rudolf[Hg.]: Leichenpredigten. Quellen zur Erforschung der Frühen Neuzeit. Marburg/Lahn: Forschungsstelle für Personalschriften, 1989. 20 S.
KILLIUS 1999
Killius, Christina: Die Antiqua-Fraktur Debatte um 1800 und ihre historische Herleitung. Wiesbaden : Harrassowitz, 1999. 488 S. (=Mainzer Studien zur Buchwissenschaft Bd. 7)
REHSE 1998
Rehse, Ernst-Günther: Gebrochene Schriften. Alphabete, Abkürzungen, Zeichen, Druckbeispiele, Fonts. Gotisch, Rundgotisch, Schwabacher, Fraktur, Fraktur-Varianten, Schreibschrift. Schaubuch, Nachschlagewerk und Hilfsbuch für den Umgang mit gebrochenen Schriften. Itzehoe : Verl. Beruf + Schule, 1998. 240 S.
SCHNEIDER
Schneider, Michael: Geschichte der deutschen Orthographie. Marburg. 30 S.
http://decemsys.de/sonstig/gesch-rs.pdf [Gesehen am 12.02.2013]
TSCHICHOLD 1952
Tschichold, Jan: Meisterbuch der Schrift. Ein Lehrbuch mit vorbildlichen Schriften aus Vergangenheit und Gegenwart für Schriftenmaler, Graphiker, Bildhauer, Graveure, Lithographen, Verlagshersteller, Buchdrucker, Architekten und Kunstschulen. Ravensburg : Maier, 1952. 238 S.
VEREIN FÜR COMPUTERGENEALOGIE
Alte Krankheitsbezeichnungen. Verein für Computergenealogie
http://wiki-de.genealogy.net/Kategorie:Krankheitsbezeichnung [Gesehen am 12.02.2013]
VOESTE 2008
Voeste, Anja: Orthographie und Innovation. Die Segmentierung des Wortes im 16. Jahrhundert. Hildesheim [u. a.] : Olms, 2008. VIII, 250 S.
http://ebooks.ciando.com/book/index.cfm/bok_id/335655 (Online-Publikation 2012)
WALTHER 2006, 1
Walther, Karl Klaus: Buchschmuck
in: Lexikon der Buchkunst und der Bibliophilie. Hamburg : Nikol, 2006. S. 122-130
WALTHER 2006, 2
Walther, Karl Klaus: Druckschrift
in: Lexikon der Buchkunst und der Bibliophilie. Hamburg : Nikol, 2006. S. 146-150
WALTHER 2006, 3
Walther, Karl Klaus: Fraktur
in: Lexikon der Buchkunst und der Bibliophilie. Hamburg : Nikol, 2006. S. 186-187

WALTHER 2006, 4
Walther, Karl Klaus: Personalschriften
in: Lexikon der Buchkunst und der Bibliophilie. Hamburg : Nikol, 2006. S. 287-293
WITZEL 2003
Witzel, Jörg: Der Tod kam im Examen. Zur Auswertung von Leichenpredigten in der Marburger Forschungsstelle für Personalschriften
in: Marburger Uni-Journal 15(2003), S. 40-43
http://www.uni-marburg.de/aktuelles/unijournal/15/Personalschriften [Gesehen am 12.02.2013]
WITZEL 2010
Witzel, Jörg: Texte für die Ewigkeit
in: Epoc 3(2010), S. 64-69
ZIPPEL 2009
Typographie der Pausen. 2009. 4 S.
http://www.logo-sz.de/site/ESSAYS/Puenktchen_Puenktchen_Komma_Strich.pdf [Gesehen am 12.02.2013]

OCR und Korrekturverfahren

ANDERSON 2010
Anderson, Niall: Best Practice Guide: Optical Character Recognition. 2010
http://www.impact-project.eu/uploads/media/IMPACT-ocr-bpg-pilot-s3.pdf [Gesehen am 12.02.2013]
BÜCHLER 2011
Büchler, Marco: Textvervollständigung, OCR- und Rechtschreibkorrektur. Drei Sichten auf gleiche Methoden. Vortrag 12.10.2012, München
http://www.slideshare.net/mdz-bsb/digitalisierungspraxis-bchler-textvervollstndigung [Gesehen am 12.02.2013]
FURRER 2011
Furrer, L; Volk, M.: Reducing OCR errors in Gothic-script documents
in: 8th International Conference on Recent Advances in Natural Language Processing (RANLP 2011), Hisar, 16 September 2011 – 16 September 2011, 97-103.
http://www.zora.uzh.ch/49812/4/Furrer_Volk_2011_V.pdf [Gesehen am 12.02.2013]
GEIERHOS 2007
Geierhos, Michaela: Grammatik der Menschenbezeichner in biographischen Kontexten. Magisterarbeit, CIS München, gekürzte Fassung. 2007. 160 S.
http://www.cis.uni-muenchen.de/download/cis_ais/ais-002.pdf [Gesehen am 12.02.2013]
HAUSER 2007
Hauser, Andreas W.: OCR Postcorrection of Historical Texts. Schriftliche Hausarbeit im Fach Computerlinguistik, CIS München. 2007. 90 S.
http://www.cip.ifi.lmu.de/~hauser/papers/histOCRNachkorrektur.pdf [Gesehen am 12.02.2013]
HÖHN 2006
Höhn, Winfried: Mustererkennung in Frühdrucken. Diplomarbeit, Institut für Informatik, Lehrstuhl für Informatik II Würzburg. 2006. 56 S.
http://opus.bibliothek.uni-wuerzburg.de/volltexte/2008/3042/pdf/Diplomarb_Hoehn.pdf [Gesehen am 12.02.2013]
KÄMMERER 2009
Kämmerer, Carmen: Vom Image zum Volltext – Möglichkeiten und Grenzen des Einsatzes von OCR beim alten Buch
in: Bibliotheksdienst 43(2009) S. 626-659
http://www.zlb.de/aktivitaeten/bd_neu/heftinhalte2009/Technik010609BD.pdf [Gesehen am 12.02.2013]
MÜHLBERGER 2011
Mühlberger, Günter: Strukturanalyse auf der Basis von OCR Ergebnissen. Vortrag 11.10.2012, München.
http://www.slideshare.net/impactproject/bsb-demo-day-mhlberger-dokumentstrukturanalyse [Gesehen am 12.02.2013]
NEUMANN 2008
Neumann, Andreas: Konstruktion historischer Wörterbücher. München. 2008. 116 S.
http://www.neumann.biz/data/computerlinguistik/Konstruktion_historischer_W%C3%B6rterb%C3%BCcher.pdf
[Gesehen am 12.02.2013]
OCR-WORKSHOP 2011
Historische Dokumente auf dem Weg zum digitalen Volltext. Workshop 11.-12. Oktober 2011, BSB München: Präsentationen
http://www.slideshare.net/mdz-bsb [Gesehen am 12.02.2013]
OPITZ 2002
Opitz, Andrea: Document Type Definitions zur Erschließung von Gattungen des Barock im Internet. Ein Projekt an der Herzog August Bibliothek Wolfenbüttel
In: Jahrbuch für Computerphilologie 5(2003) S. 55-64
http://computerphilologie.uni-muenchen.de/jg03/opitz.html [Gesehen am 12.02.2013]

REFFLE 2011
Reffle, Ulrich: Analyse und Nachkorrektur historischer und OCR- erfasster Ergebnisse. Vortrag 11.10.2012, München
http://www.slideshare.net/impactproject/bsb-demo-day-reffle-analyse-und-nachkorrektur [Gesehen am 12.02.2013]
RINGLSTETTER 2003
Ringlstetter, Christoph: OCR-Korrektur und Bestimmung von Levenshtein-Gewichten. Magisterarbeit, CIS München. 2003. 130 S.
http://www.cis.uni-muenchen.de/people/kristof/Publications/ringlstetter03.pdf [Gesehen am 12.02.2013]
RUDLOF 2006
Rudlof, Christiane: Handbuch Software-Ergonomie. 2., überarb. Aufl. Tübingen : Unfallkasse Post und Telekom, 2006. 144 S.
http://www.ukpt.de/pages/dateien/software-ergonomie.pdf [Gesehen am 12.02.2013]
SCHLARB 2011
Schlarb, Sven: Entscheidungsfindung in der Digitalisierung durch experimentelle Workflow-Entwicklung. Vortrag, 11.10.2012, München
http://www.slideshare.net/impactproject/bsb-demo-day-schlarb-workflowdesign [Gesehen am 12.02.2013]
SCHULZ 2006
Schulz, Klaus U.: Nachkorrektur von Ergebnissen einer optischen Charaktererkennung. 2006. 108 S.
http://www.cis.uni-muenchen.de/~uli/kurse/ws0708/hist_ocr/material/schulz_ocr.pdf [Gesehen am 12.02.2013]
STÄCKER 2002
Stäcker, Thomas: XML für Alte Drucke – welche Erschließungspotentiale bietet die neue Auszeichnungssprache? 2002. 7 S.
http://www.hab.de/bibliothek/wdb/barockdtd/staecker.pdf [Gesehen am 12.02.2013]
STOCKMANN 2010
Stockmann, Ralf: Was tun mit den Ergebnissen der OCR? Vortrag 04.03.2010
http://www.slideshare.net/impactproject/stockmann-endnutzer-impact-workshop-muc-3344329 [Gesehen am 12.02.2013]
STROHMAIER 2004
Strohmaier, Christian M.: Methoden der lexikalischen Nachkorrektur OCR-erfasster Dokumente. Dissertation, LMU München: Fakultät für Sprach- und Literaturwissenschaften. 2004. 158 S.
http://edoc.ub.uni-muenchen.de/3674/1/Strohmaier_Christian.pdf [Gesehen am 12.02.2013]
WIENERS 2008
Wieners, Jan Gerrit: Zur Erweiterungsfähigkeit bestehender OCR Verfahren auf den Bereich extrem früher Drucke. Magisterarbeit, Historisch-Kulturwiss. Informationsverarbeitung. Köln, 2008. 81 S.
http://www.hki.uni-koeln.de/files/MA_wieners.pdf [Gesehen am 12.02.2013]

Projekte und Web-Adressen

BERLINER FUNERALSCHRIFTEN
Personale Gelegenheitsschriften. Historische Drucke
http://staatsbibliothek-berlin.de/die-staatsbibliothek/abteilungen/historische-drucke/sammlungen/bestaende/personale-gelegenheitsschriften/ [Gesehen am 12.02.2013]
Pilotprojekt zum OCR-Einsatz bei der Digitalisierung der Funeralschriften der Staatsbibliothek zu Berlin
http://staatsbibliothek-berlin.de/die-staatsbibliothek/abteilungen/historische-drucke/aufgaben-profil/projekte/funeralschriften/ [Gesehen am 12.02.2013]
Federbusch, Maria: Praxistest zweier OCR-Softwareprodukte am Beispiel ausgewählter Funeralschriftenbestände der SBB. Vortrag 12.10.2011, München
http://www.slideshare.net/mdz-bsb/digitalisierungspraxis-federbusch-ocrpraxistest [Gesehen am 12.02.2013]
Metadaten zu den Berliner Funeralschriften. 2008
http://staatsbibliothek-berlin.de/fileadmin/user_upload/zentrale_Seiten/historische_drucke/pdf/Metadaten_Funeralschriften.pdf [Gesehen am 12.02.2013]
DEUTSCHES TEXTARCHIV / Volltexterfassung schlesischer Leichenpredigten
http://www.deutschestextarchiv.de/ [Gesehen am 12.02.2013]
HELMSTEDTER DRUCKE ONLINE
Helmstedter Drucke Online
Projekt der HAB Wolfenbüttel. Volltextgenerierung durch OCR mit BIT-Alpha
http://www.hab.de/de/home/wissenschaft/projekte/helmstedter-drucke-online.html [Gesehen am 12.02.2013]

IMPACT
Impact. Centre of competence
http://www.digitisation.eu/ [Gesehen am 12.02.2013]
Improving Access to Text
http://www.impact-project.eu/ [Gesehen am 12.02.2013]
Tools and applications
neue Seite: http://www.digitisation.eu/tools/ [Gesehen am 12.02.2013]
alte Seite: http://www.impact-project.eu/taa/tech/ [Gesehen am 12.02.2013]
Anderson, Niall: Best Practice Guide: Optical Character Recognition
http://www.impact-project.eu/uploads/media/IMPACT-ocr-bpg-pilot-s1.pdf bis ...s3 [Gesehen am 12.02.2013]
Impact Day / Impact Conference: Presentations. Dec. 2009-Dec. 2011
http://www.slideshare.net/impactproject [Gesehen am 12.02.2013]
MARBURGER PERSONALSCHRIFTENSTELLE
Forschungsstelle für Personalschriften. Akademie der Wissenschaften und der Literatur Mainz
http://www.personalschriften.de/ [Gesehen am 12.02.2013]
THELO
Thesaurus Locorum
http://www.personalschriften.de/datenbanken/thelo.html [Gesehen am 12.02.2013]
THEPRO
Thesaurus Professionum
http://www.personalschriften.de/datenbanken/thepro.html [Gesehen am 12.02.2013]
NACHKORREKTUR VON OCR-ERGEBNISSEN
Projekt: Domänen- und dokumentenadaptive Verfahren zur Nachkorrektur von OCR-Ergebnissen. CIS München. 2003-2010
Eintrag 5419670 in der DFG-Datenbak GEPRIS
http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=5419670 [Gesehen am 12.02.2013]

8. Erfahrungsbericht Helmstedter Drucke Online an der Herzog August Bibliothek Wolfenbüttel

Thomas Stäcker

8.1 Zusammenfassung

Ziel des Projekts ist die Image- und Volltextdigitalisierung der in Helmstedt gedruckten Werke, die sich in der Herzog August Bibliothek befinden. Die Herzog August Bibliothek besitzt etwa 10.000 Drucke mit dem Druckort Helmstedt, die bis zur Schließung der Universität Helmstedt 1810 in Helmstedt gedruckt worden sind. Die auf das 16. und 17. Jh. entfallenden Drucke sollen als Image und im Volltext per OCR digitalisiert, mit Strukturdaten zur Navigation versehen und im Verbundkatalog sowie im VD 16 und VD 17 nachgewiesen werden. Der aus dem 18 Jh. stammende Anteil wird digitalisiert, auf nationalbibliographischem Niveau katalogisiert und zur Nachnutzung einem möglichen VD 18 zur Verfügung gestellt. Mit der nahezu kompletten Digitalisierung der Produktion eines der wichtigsten norddeutschen Universitätsdruckorte – Helmstedt findet sich unter den 10 am häufigsten nachgewiesenen Druckorten im VD 17 – wird nicht nur ein substantieller Beitrag zur Komplettdigitalisierung des deutschen gedruckten Kulturerbes geleistet, sondern erstmals ein Überblick über die Druckproduktion einer bedeutenden frühneuzeitlichen Universität geschaffen. Das Projekt flankiert den programmatischen und auf mehrere Jahre angelegten Forschungsschwerpunkt der HAB zur Erforschung der Universitätsgeschichte Helmstedts und kommt unmittelbar mehreren Forschungsprojekten in diesem Feld zugute.

8.2 Bedeutung der Helmstedter Drucke

Das Vorhaben der Digitalisierung der Helmstedter Drucke steht im Kontext der Erforschung der Universitäts- und Wissenschaftsgeschichte der Frühen Neuzeit.[105] In ihrer Selbstbeschreibung formulierte jede frühneuzeitliche Universität einerseits ihre Ziele und Ansprüche, andererseits deren Einlösung. Da der Status universitären Wissens von spezifischen Geltungsgründen und der Einhaltung bestimmter Verfahren abhing, war eine Hochschule zu intensiver Selbstbeobachtung gezwungen. Deren Ergebnisse bilden einen wichtigen zeitgenössischen Beurteilungsmaßstab für die Qualität und Leistung einer Universität und insofern einen geeigneten Ausgangspunkt für ein retrospektives *reviewing*, zumal sie aufgrund der kompetetiven Grundstruktur des Wissenschafts- und Universitätssystems im Rahmen der fürstenstaatlichen Ordnung und der gelehrten Welt kommuniziert werden mussten. Diese als universitäre Repräsentation und Selbtvergewisserung zu bezeichnende Aufgabe bündelt sich wie in einem Brennglas in der gesamten Druckproduktion der privilegierten Universitätsbuchdruckerei. Das Spezifikum der Druckproduktion einer Universitätsdruckerei[106] ist, dass sie an einer fürstenstaatlich initiierten, finanzierten und kontrollierten Einrichtung stattfand und insofern politisch wie konfessionell autorisiert, legitimiert und funktionalisiert wurde. Die Druckproduktion der Universität kann damit in besonderer Weise über universitäre und staatliche Autorisierungsmechanismen Aufschluss geben. Sie manifestiert die Paradigmen akademischen Wissens in den verschiedenen Fakultäten ebenso wie die verfolgten Lehrpraktiken, denn frühneuzeitliche Universitäten waren noch weit bis ins 18. Jh. hinein in noch stärkerem Maße als heute Unterrichts- und Ausbildungsanstalten.

Wurden die Hochschulschriften, also die akademischen Dissertationen, Reden und Programme, lange Zeit als eine eher uninteressante Quellengruppe eingestuft, hat sich diese Wertung seit den 1980er Jahren geradezu umgekehrt, wobei insbesondere die Dissertationen des 16. bis 18. Jahrhunderts mittlerweile als „ungehobene Schätze“ betrachtet werden.[107] Gerade an diesen universitären Gelegenheitsschriften lässt sich ablesen, wie die eigentliche „Wissensproduktion“ an den Universitäten aussah, womit ein wichtiger Beitrag zur Alltagsgeschichte der Bildung und Ausbildung in der Frühen Neuzeit geleistet

[105] Zur Universitäts- und Bildungsgeschichte der Frühen Neuzeit im Allgemeinen vgl. insb.: Paulsen, Friedrich: Geschichte des gelehrten Unterrichts auf den deutschen Schulen und Universitäten vom Ausgang des Mittelalters bis zur Gegenwart. Mit besonderer Rücksicht auf den klassischen Unterricht, 2 Bde., Leipzig / Berlin 1919 / 1921 (ND 1965); Hammerstein, Notker (Hg.): Handbuch der deutschen Bildungsgeschichte, Bd. I: 15. bis 17. Jahrhundert. Von der Renaissance bis zum Ende der Glaubenskämpfe, München 1996; Rüegg, Walter (Hg.): Geschichte der Universität in Europa, Bd. II: Von der Reformation bis zur Französischen Revolution (1500 – 1800), München 1996; Schindling, Anton: Bildung und Wissenschaft in der Frühen Neuzeit 1650 – 1800, 2. Aufl., München 1999; Hammerstein, Notker: Bildung und Wissenschaft vom 15. bis zum 17. Jahrhundert, München 2003; Bruning, Jens: Innovation in Forschung und Lehre : die Philosophische Fakultät der Universität Helmstedt in der Frühaufklärung 1680 – 1740. Wiesbaden 2012 (Wolfenbütteler Forschungen, 132).– Eine Auswahlbibliographie zum Standort Helmstedt steht über das Helmstedt-Portal zur Verfügung: http://uni-helmstedt.hab.de/docs/bibliographie_helmstedt.pdf [Stand: 02/2013]

[106] Zu dieser: Eule, Wilhelm: Helmstedter Universitätsbuchdrucker. Helmstedt 1921. S.a. Benzing, Josef: Die Buchdrucker des 16. und 17. Jahrhunderts im deutschen Sprachgebiet. 2 verb. Aufl. Wiesbaden 1982, S. 200-203 mit weiterer Literatur zu den einzelnen Druckern.

[107] Eule (wie Anm. 2), S. 29ff.

werden kann. Zu betonen ist dabei, dass den akademischen Druckerzeugnissen nicht nur als Einzelpublikationen, sondern auch als „Massenquelle“ bzw. „serielle“ Quelle besondere Bedeutung zukommt und sie gerade in diesem Charakterzug einen einzigartigen Blick in die Entwicklung der verschiedenen Wissensdisziplinen ermöglichen.

8.3 Umfang und Art der Helmstedter Druckproduktion

Die wichtigsten Drucker des 16. und 17. Jh. in Helmstedt waren Jakob Lucius I (1579-1597), Jakob Lucius II (1598-1616, Erben bis 1633 vermutlich unter der Leitung von Henning Müller d. Ä.), Georg Calixt (1629-1676 mit Pächtern, insb. Henning Müller d. J.), Jacob Lucius III (1634-1639), Henning Müller d. J. (1640-1674), Johann Heitmüller (1656-1677), Jakob Müller (1661-1681), Georg Wolfgang Hamm (1681-1714), Heinrich Hesse (1681-1715/16) und Salomon Schnorr (1690-1723). Für das 18. Jh. sind insbesondere Johann Drimborn und Johann Heinrich Kühnlin zu nennen. Bemerkenswert ist die Betätigung des berühmten Gelehrten und Theologen Georg Calixt als Druckherr. Hier zeigt sich die enge Verschränkung von Universität und Druckerei in besonders augenfälliger Weise.[108] Besonders wichtig war für die Helmstedter Druckgeschichte Henning Müller d. J. Seine Produktion zeigt denn auch das Spektrum der Produktion auf exemplarische Weise. Von den 2.422 im VD 17 nachgewiesenen Drucken aus seiner Offizin entfallen mehr als die Hälfte, nämlich 1.342 auf Dissertationen und 276 auf Hochschulschriften eine bis dato vernachlässigte Quellengruppe.[109] Dazu kommen 91 Leichenpredigten und 590 sonstige Gelegenheitsschriften, die in besonderer Weise dazu geeignet sind, biographische Hintergründe des akademischen Personals aufzuhellen. Rund 500 Werke entfallen auf sonstige Werke und wissenschaftliche Abhandlungen, insbesondere von Professoren der Universität Georg Calixt, Johannes Caselius, Hermann Conring, Hermann v.d. Hardt, Heinrich Meibom d. Ä. und d. J., Johann Lorenz von Mosheim, Heinrich Rixner, Heinrich Julius Scheuerl, Valentin Heinrich Vogler und Giordano Bruno (ein Jahr in Helmstedt, nicht als Professor, hielt aber vermutlich Privatvorlesungen), deren Arbeiten von der hohen Qualität der Forschung in Helmstedt in der Frühen Neuzeit Zeugnis ablegen.

Das im engeren Sinne akademische Schrifttum (Dissertationen, Abhandlungen) wurde, wie üblich in dieser Zeit, fast ausnahmslos in Latein verfasst. Allerdings finden sich in den Drucken immer wieder deutsche Einsprengsel und Zitate. Daneben druckten die Universitätsdrucker vor allem Predigten oder Leichenpredigten in Deutsch. Lateinische Texte wurden in Antiqua, Deutsche in Fraktur gesetzt, wobei die Grenze mitunter im Wort selbst verlief. So finden sich Fälle, wo die Endung „e“ des Worts „Materie“ in Fraktur, der Rest „Materi“ in Antiqua gesetzt wurde, weil es sich um ein lateinisches Wort handelt, das eigentlich die Endung „a“ trägt. Typischerweise enthalten die Drucke Mischformen von Antiqua, Fraktur und Kursive als Brotschriften. Hinzu kommen barocke Auszeichnungsschriften, die die Drucker aber für das in der Masse wenig aufwendig gestaltete Universitätsschrifttum vergleichsweise zurückhaltend einsetzten. Die verwendeten Fonts weisen wenig Varianten auf und scheinen über die Jahre weitgehend stabil gewesen zu sein, da Fonts von Universitätsdrucker zu Universitätsdrucker vererbt und der Erwerb neuer Fonts aus ökonomischen Gründen vermieden wurde.

Obwohl es zur Frage des deutschen bzw. europäischen Schriftenhandels beklagenswert wenige wissenschaftliche Untersuchungen gibt, deutet doch das wenige, was wir haben, darauf hin, dass man auch unabhängig von diesem innerhelmstedtischen Befund von weitgehender Einheitlichkeit bei den Schriften ausgehen kann und dass gerade im 17. Jh. ein starker Zentralisierungsprozess im Handel mit Schriften eingetreten ist.[110] Hervorzuheben ist hier vor allem die erfolgreiche Tätigkeit der Egenolffschen/Lutherschen Schriftgießerei in Frankfurt. Insofern kann man von einem hohen Homogenitätsgrad ausgehen, was für die OCR-Konversion der Barockschriften nicht ohne Belang ist.

[109] Vgl. Komorowski, Manfred: Die Hochschulschriften des 17. Jahrhunderts und ihre bibliographische Erfassung. In Wolfenbütteler Barock-Nachrichten 24 (1997) 1, S. 19-42.

[110] Vgl. hierzu Carter, Harry: A View of Early Typography up to about 1600. Reprinted. London 2002.

8.4 Projektziele, Rahmenbedingungen

Ziel des Projekts ist die Image- und Volltextdigitalisierung sowie Erschließung der in Helmstedt gedruckten Ausgaben, die vom Beginn des Universitätsdrucks bis zur Schließung der Universität 1810 in Helmstedt gedruckt wurden und sich heute in der Herzog August Bibliothek befinden.

Die Wolfenbütteler Sammlung bietet optimale Ausgangsbedingungen für ein Digitalisierungsprojekt der Helmstedter Druckproduktion. Die Helmstedter Universitätsdrucker waren verpflichtet, Exemplare ihrer gesamten Produktion an die eigene Universitätsbibliothek und die fürstliche Bibliothek in Wolfenbüttel zu überweisen.[111] Da beide Sammlungen, die Wolfenbütteler und die Helmstedter nach Schließung der Universität Helmstedt wieder in Wolfenbüttel vereint wurden, kann von einer sehr dichten Überlieferungslage vor Ort ausgegangen werden. Da diese mit Ausnahme der in Helmstedt verbliebenen Drucke[112] und der Drucke des 18. Jh. in den bereits abgeschlossenen nationalbibliographischen Verzeichnissen des VD 16[113] und VD 17[114] durch die HAB Wolfenbüttel katalogisiert wurden, liegen weitgehend alle erforderlichen Metadaten für die Digitalisierung vor. Im VD 16 sind 1.216, im VD 17 sind 7.485 (Stand: 3.10.2009) Drucke für den Druckort Helmstedt nachgewiesen. Von diesen besitzt die HAB gerundet aus dem 16. 789, aus dem 17. 5.366 und aus dem 18./19. Jh.(-1810) 4.390 Ausgaben, also insgesamt rund 10.000 Ausgaben mit geschätzt 1 Mio. Seiten.

Ein Novum für ein Digitalisierungsprojekt zur Frühen Neuzeit war, neben der eigentlichen Imagedigitalisierung auch eine OCR-Konversion des Schriftguts in Angriff zu nehmen. Allerdings wurde die ursprünglich intendierte vollständige Bearbeitung des Korpus auf Wunsch der DFG auf ein kleineres Sample von 120.000 Seiten reduziert, um angesichts der erheblichen Kosten zunächst Erfahrungen mit dem Verfahren zu sammeln.

Bei der anvisierten Komplettdigitalisierung der Druckerzeugnisse mit dem Druckort Helmstedt mussten bereits laufende Digitalisierungsprojekte, insbesondere an der Bayerischen Staatsbibliothek München, aber auch anderorts berücksichtigt werden, um Doppeldigitalisierungen so weit wie möglich zu vermeiden. Gerade mit Blick auf die OCR-Bearbeitung schien das Google-Projekt der Bayerischen Staatsbibliothek zunächst von Interesse. Eine genauere Vorprüfung zeigte aber, dass die Google Digitalisate nicht für das Projekt nutzbar waren. Zum einen ist die von Google im Internet bereitgestellte Qualität zu schlecht für eine nachgeordnete OCR-Bearbeitung (Gerüchten zufolge besitzt Google intern bessere Scans, was aber für aktuell durchgeführte Projekte leider wenig nützt), zum anderen ist es nach den Geschäftsbedingungen von Google[115] nicht gestattet, Volltexte außerhalb von Google zu benutzen, was eklatant Forschungsinteressen bezüglich der freien Nutzung der Volltexte widerspricht, z. B. in dem Wunsch nach unabhängiger Eigenindexierung und Bildung von reproduzierbaren Rankingmechanismen.[116] Insofern scheint nach dem gegenwärtigen Stand ein auf Volltextgenerierung angelegtes wissenschaftliches Digitalisierungsprojekt grundsätzlich nur mit Digitalisaten aus öffentlicher Hand möglich zu sein, zumindest sofern entsprechende CC-Rechte am Volltext eingeräumt werden, was vor dem Hintergrund einer DFG-Förderung typischerweise der Fall sein dürfte.

[111] Vgl. Schneider, Heinrich: Beiträge zur Geschichte der Universitätsbibliothek Helmstedt. Helmstedt 1924, S. 75, Anm. 194.

[112] Diese sind in einem eigenständigen Projekt „Katalogisierung der im Juleum Helmstedt verbliebenen Drucke der ehemaligen Universitätsbibliothek Helmstedt“ derzeit in Bearbeitung: http://www.hab.de/de/home/wissenschaft/projekte/katalogisierung-der-im-juleum-helmstedt-verbliebenen-drucke-der-ehemaligen-universitaetsbibliothek-helmstedt.html [Stand: 02/2013]

[113] http://www.vd16.de [Stand: 02/2013]

[114] http://www.vd17.de [Stand: 02/2013]

[115] https://www.google.de/intl/de/policies/terms/regional.html; vgl. a. https://play.google.com/intl/de/about/books-terms.html [Stand: 02/2013]

[116] Zum Problem der mangelnden Wissenschaftstauglichkeit von Googles Suchmaschine vgl. Elli Pariser: The filter bubble : what the Internet is hiding from you. London 2011.

Die Volltexte sollten nach vorausgegangenen Tests mit der Software BIT-Alpha der Fa. B.I.T. Tomasi generiert werden. Wie oben geschildert, besteht das Charakteristikum der Software darin, dass sie eine effiziente Oberfläche zum schnellen Erlernen von Schriften bietet. Hierzu wurde eine Hilfskraft eingestellt, die Schriften von Helmstedter Druckern trainierte. Dabei handelte es sich überwiegend um Antiquaschriften inklusive Kursive, zu einem kleineren Teil um Fraktur. Anschließend sollten die Drucke bearbeitet und die Ergebnisse in ALTO[117] exportiert und zur Nachnutzung in der Wolfenbütteler Digitalen Bibliothek in ein TEI P 5 kompatibles Format konvertiert werden. Die Suche in den Texten erfolgte mit eXist, einer nativen XML-Datenbank.

8.5 Ergebnisse, Erfahrungen

Nach der Digitalisierung von 4.800 Drucken (Stand 01.06.2012) hat sich ein Seitendurchschnitt von 87 Seiten pro Ausgabe ergeben. Die gegenüber sonstigen Projekten[118] eher geringe durchschnittliche Seitenzahl erklärt sich daraus, dass es sich bei Universitätsschriften meist um weniger umfangreiche Druckerzeugnisse wie Dissertationen und Gelegenheitsschriften handelt. Für die Archivkopie fallen derzeit pro Image durchschnittlich 28,5 MB pro Seite (uncompressed TIFF) an.

Ziel der ersten Phase des Projekts war es, 120.000 Seiten per OCR zu bearbeiten. Die OCR-Komponente des Antrags war ein wichtiger neuer und innovativer Schritt im Prozess der Massendigitalisierung, weil bislang bei Materialien des 16.-18. Jh. fast ausschließlich die Imagedigitalisierung zum Einsatz kam. Wegen der dargelegten Schwierigkeiten der Vorlagen kam nach einigen Tests mit anderen Anbietern die OCR-Software der Firma B.I.T. Bureau Ingénieur Tomasi zum Einsatz. Das Spezifikum dieser Software liegt darin, dass sie nicht, wie z. B. Abbyy, auf weitgehend werkseitig vortrainierte Fonts angewiesen ist, so dass die typischen Mischtexte dieser Zeit (Antiqua, Fraktur, Kursive, Griechisch und andere Typen) nicht korrekt erkannt werden, sondern z. B. entweder nur der Antiqua- oder nur der Frakturanteil. Einen typischen Fall solcher Mischung zeigt sich in Abb. 77.

Die jeweiligen Typen mussten zunächst trainiert werden; dies konnte schriftübergreifend erfolgen. Effekte des „Übertrainierens", wie man sie wohl bei anderen Produkten wie Abbyy beobachten kann, traten nicht auf. Das Training konzentrierte sich zunächst auf das 17. Jh. und den Helmstedter Großdrucker Henning Müller d. J., der allein für rund 3.000 Druckerzeugnisse verantwortlich ist. Damit verband sich die Absicht, im weiteren Verlauf auf der Basis eines weitgehend homogenen Schriftmaterials arbeiten zu können. Trainiert werden mussten Antiqua und Fraktur, ebenso Kursivschrift und altgriechische Schrift. Letztere wurde allerdings aus Kapazitätsgründen bisher nur in Ansätzen trainiert, weil dafür spezialisierte Kompetenzen erforderlich sind. Sie soll im Laufe eines Fortsetzungsprojekts durch Mitarbeiter der Bibliothek weiter trainiert und die Erkennungsgenauigkeiten für frühneuzeitliche griechische Fonts verbessert werden. Die Trainingsphase verlief weitgehend ohne Probleme; die Software erwies sich als sehr komfortabel in der Handhabung. Besonders effizient war der so genannte „Sequencer", der es erlaubte typische Buchstabenkombinationen zu identifizieren (s. Kap. 4.4), typische Fälle sind etwa „n+i" Verbindungen, die als „m" zu lesen sind. Trainiert wurden von der studentischen Hilfskraft im Laufe der ersten Projektphase (24 Monate) 132.000 Zeichen mit einem Aufwand von durchschnittlich fünf Wochenstunden.

[117] http://www.loc.gov/standards/alto/ [Stand: 02/2013]
[118] Durchschnittszahlen auf dem Server der HAB: 140 Seiten bei 14.000 Werken. Bei Dünnhaupt Digital sind es im Schnitt 235 Seiten.

putationi includere, & eruditorum placido examini publici exercitij gratia ſubjicere eſt animus: & quidem res præcipuas tam olim quam hodienum in Comitiis tractari conſuetas, in quantum à me ex hiſtoriis peti potuerunt, theſibus hiſce inſerere conabor. Quod ut feliciter ſuccedat, tu Deus largire vires, & fac ut omnia in nominis tui gloriam cedant.

VIII. Res vero quarum cauſa Comitia haberi ſolent, pro varietate negotiorum Imperio incumbentium, uti olim ita & hodie diverſæ ſunt. Omnia nimirum illa quæ ad ſalutem vel Statum Regis & regni pertinebant, in Comitiis Carolinorum Regum ævo tractabantur. *Hincm. Epiſt. 3. c. 33.* Hodieque omnia negotia quæ ad Imperium vel conſervandum vel defendendum pertinent, in Comitiis tractari ſolent R. A. *de anno 1512.* Et ut paucis me abſolvam, quicquid eſt, quod ad gloriam Dei, conſervationem juſtitiæ, honeſtatis & publici ſtatus ædificationem conferre poſſit, de eo in conventibus publicis agitur. Reformation guter Policey zu Augßburg *anno 1530.* §. 1. Sampt jhnen alles das fürzunehmen zu rahtſchlagen / zu handeln und zu ſchlieſſen / daß zu fürderſt Gott dem Allmächtigen zu Ehr und Lob / gemeiner Chriſtenheit und Teutſcher Nation zu Wolfahrt Fried und Einigkeit / auch dem H. Röm. Reich zu nutz auffnehmen und gedeyen gereichen möcht. R. A. des Reichßtags zu Regenßburg *de anno 1541.* §. 1. Alles anders zu handeln und zu ſchlieſſen / daß dem H. Reich / und deſſelben Ständen zu wolfahrt nutz und gutem reichen mög. Ac in noviſſima Pacis conſtitutione ea de re ita cavetur. *Gaudeant ſine contradictione jure ſuffragii in omnibus deliberationibus ſuper negotiis Imperii, præſertim ubi leges ferendæ, vel interpretandæ, bellum decernendum, tributa indicenda, delectus aut hoſpitationes militum inſtituendæ,*

77 Antiqua, Fraktur u. Kursive aus Hermann Conring: Dissertatio De Praecipuis Negotiis In Comitiis Imperii Germanici Olim Et Hodienum Tractari Solitis Helmstedt, 1666

Die Konversionsphase gestaltete sich anders als die Trainingsphase steiniger, was im Wesentlichen auf zwei Gründe zurückzuführen war. Zum einen erwiesen sich Parametrierung und Einstellungen der Software als zu komplex, zum anderen gab es zahlreiche Probleme mit den Vorlagen. Was Ersteres anlangte, so konnte in der fast zweijährigen Pilotphase trotz intensiver Bemühungen nicht nachvollzogen werden, wie sich welche Einstellungen auf das Ergebnis auswirkten (vgl. Kap. 5 u. 6.3 sowie Anh. 7.3). Grund war einerseits das Fehlen einer Dokumentation, die im Detail verdeutlicht hätte, wie sich die vielfältigen Parametrierungsmöglichkeiten auswirken. Andererseits entwickelte die Firma die Software laufend fort, was zwar zu Verbesserungen in der Erkennungsqualität führte, jedoch auch Änderungen bei den Parametereinstellungen zur Folge hatte. Insgesamt war es so weitgehend unmöglich, empirisch reproduzierbare Effekte zu erzielen, da Änderungen an einem Parameter dazu führten, dass andere Einstellungen unwirksam wurden, sich Einstellungen gegenseitig aufhoben oder Einstellungen in einer älteren Version sich nicht mehr so verhielten wie Einstellungen einer neueren Version. Bemühungen, die Software zu „verstehen“ wurden daher nach einer gewissen Zeit abgebrochen und es weitgehend der Firma überlassen, nach Maßgabe der jeweiligen Vorlagen die richtigen Einstellungen festzulegen. Die Zusammenarbeit in dieser Hinsicht verlief jedoch reibungslos und die Firma zeigte sich sehr kooperativ und entgegenkommend. Dass materialabhängig Anpassungen überhaupt sinnvoll sind, hängt mit den zugrundeliegenden Softwarealgorithmen zusammen. Marktübliche Software wie Abbyy oder Omnipage gehen von „durchschnittlichen” Einstellungen aus, was zwar dazu führt, dass nicht das Optimum aus der Software herausgeholt wird, was es aber dem Nutzer erlaubt, ohne intime Kennisse der Einstellungsmöglichkeiten doch noch befriedigende Ergebnisse zu erzielen. Letztlich führten diese Umstände im Projekt zu der Einsicht, dass die Software von B.I.T. für den durchschnittlichen bzw. selbst für den erfahrenen Anwender mit Ausnahme des hervorragenden Trainingsinterfaces eigentlich nicht geeignet ist und legte für das geplante Folgeprojekt einen Wechsel im Vorgehen nahe. Abgesehen vom Training einiger Sonderschriften (vor allem Griechisch), das mangels entsprechender Kompentenzen der Firma in der Bibliothek erfolgen muss, soll die OCR-Wandlung künftig vollständig durch die Firma in einem Dienstleistungsverhältnis durchgeführt werden.

Trotz dieser Einschränkungen konnten aber Stärken und Schwächen der Software zumindest im Allgemeinen bewertet werden. Gemäß den Prozessschritten Binarisierung, Segmentierung bzw. Layoutanalyse, *pattern matching* und *intelligent character recognition* (Wörterbuch) kann man hervorheben, dass die Stärken der Software gerade im Bereich der Binarisierung und des *pattern matching* hervorstechen; einzelne Probleme, wie stark gekrümmte Seiten und die damit verbundene problematische Grundlinienfindung, konnten mittels neuprogrammierter Algorithmen gelöst werden. Bei der Segmentierung gab es verschiedentlich noch Probleme, die teils im Laufe des Projekts behoben werden konnten (z. B. bei der Identifizierung von Initialen), teils noch bestehen, wie bei der Erkennung von Marginalien oder Zeilenzählungen, die zu dicht am Text gedruckt sind. Hier scheinen weitere Verbesserungen möglich, zumal Abbyy bessere Segmentierungsergebnisse zu erzielen vermag. Auch bei der Wörterbuchunterstützung sind noch Verbesserungen möglich. Dies liegt allerdings vor allem daran, dass geeignete Wörterbücher für das Frühneuhochdeutsche oder auch Lateinische fehlen, die das typische universitäre Vokabular der frühen Neuzeit abzubilden vermöchten. Die Ergebnisse der durch Wörterbuch unterstützten Konversion wirkten sich in der ersten Phase des Projekts noch nicht aus bzw. es können dazu noch keine verallgemeinerbaren Aussagen getroffen werden. Die noch nicht repräsentativen Ergebnisse waren einerseits besser, führten andererseits aber auch zu neuen Fehlern durch falsche Korrekturen. Andererseits gibt es eine Reihe von Phänomenen im Text, die nur mit Hilfe von Wörterbüchern gelöst werden können und bei denen reines *pattern matching* nicht weiterführt. Dies ist z. B. der Fall, wenn kleines „c” und „e”, da unsauber gedruckt, nicht visuell unterschieden oder wo innerhalb eines Worts „i+n” im Sequencer nur durch ein Wörterbuch disambiguiert werden können, weil beides „in“ und „m“ möglich ist. Der Weg, ein effizientes *pattern matching* mit einem Wörterbuch zu kombinieren, ist daher zwingend, weil eine rein „visuelle“ Erkennung bestimmte Grenzen nicht zu überschreiten vermag.

Neben den Schwierigkeiten mit der Parametrierung der Software bestanden aber auch Probleme, die aus der Qualität der Vorlagen resultierten. Die ursprüngliche Ansicht, man müsse die Software nur einen bestimmten Schrifttyp lernen lassen, um zu guten Ergebnissen zu gelangen, ist zwar im Grundsatz richtig, doch zeigte sich, dass die Schriftvarianten, auf die man bei verschiedenen Druckern trifft, sich weniger negativ auswirkten als mindere Papier- und Druckqualität, Widerdruck oder sonstige Verschmutzungen. Unterstreichungen, die oft als Durchstreichungen ausgeführt sind, machten eine Texterkennung so gut wie unmöglich.

Trotz dieser Probleme sind aber die bisherigen Ergebnisse so gut, dass eine Fortsetzung mit größeren Margen und Komplettkonversion dieses Korpus und anderer Korpora sinnvoll scheint, zumal im Fall der Helmstedter Drucke, bei denen es sich überwiegend um serielles Material handelt und der eigentliche Nutzen sich unstreitig erst aus dem Volltext ergibt.

Um aber hierzu nachvollziehbare Aussagen machen zu können, müssen zunächst zwei Gesichtspunkte näher erörtert werden. Zum einen die Frage, welche Genauigkeit der Texterkennung erforderlich ist, zum anderen wie man diese Genauigkeiten misst (8.7) . Erst aus der Beantwortung dieser Fragen lässt sich auch eine Vorstellung über den nötigen und angemessenen ökonomischen Aufwand für ein OCR-Projekt gewinnen (8.8).

8.6 Überlegungen zu Textgenauigkeiten und deren Messung

Im allgemeinen herrschen durchaus disparate Vorstellungen davon, was ein qualitativ ausreichender Text ist. An der Bayerischen Staatsbibliothek (BSB) z. B. teilt man Texte in vier Klassen ein: sehr gut (99,6-99,95% Genauigkeit), gut (97-99,5%), durchschnittlich (90-96%) und schlecht (unter 90%).[119] Gewöhnlich wird argumentiert – so auch in den bisherigen DFG-Praxisregeln (gültig bis Ende 2012) –, dass eine in diesem Sinne sehr gute Qualität Voraussetzung für eine wissenschaftliche Nutzung sei. Dem ist nach traditionellem Verständnis durchaus zuzustimmen. Gleichwohl stehen dem Überlegungen entgegen, wie sie von Thaller[120] und anderen angestellt werden, dass auch eine weniger exakte, nach den bisherigen Vorstellungen eher unzulängliche OCR-Konversion noch wissenschaftlichen Nutzen bringt, selbst wenn *argumenta ex silentio* nicht mehr möglich sein sollten. Die heutige Technik erlaubt Mengen von Texten automatisch zu bearbeiten, die im Laufe eines Gelehrtenlebens nicht hätten gelesen werden können, so dass man mit geeigneten Recherchewerkzeugen ein neues, wenn auch nicht ganz zuverlässiges Bild, über Inhalte gewinnt, zu denen man früher ohne dieses Hilfsmittel keinen Zugang gehabt hätte. Angesichts dessen relativiert sich die Frage des zuverlässigen Texts zumindest für Fragestellungen, bei denen ein anderer, meint traditionell „lesender" Zugang[121] ohnehin verwehrt wäre. Die intensive Nutzung von Google durch die Wissenschaft scheint diese Behauptung zu stützen, ebenso wie der verstärkte Rückgriff auf quantitative Methoden in den *digital humanities*. Dennoch darf man sich nicht täuschen, dass diese Nutzungsszenarien von der Forschung als „provisorisch" empfunden werden und dass z. B. strategische Großvorhaben wie das EU-Projekt CLARIN[122], in denen vor allem linguistische Verfahren zum Tragen kommen, großen Wert auf „gute" Texte legen, weil „schmutziges OCR" wissenschaftliche Fragestellungen verfälschen und vor allem dann kritisch gesehen werden müssen, wenn die Herstellungsbedingungen solcher Texte, also die eingesetzten Algorithmen der OCR-Konversion und Qualitätsstufe unbekannt sind. Was also für eine wissenschaftliche Nutzung hinreichende Qualität ist und wie sie bewertet oder gemessen wird, ist daher von mehreren Faktoren abhängig, von denen die innere Textqualität, also die höchstmögliche Übereinstimmung einer Kopie mit dem Original, nur einen Faktor darstellt. Wichtige weitere Faktoren sind u. a. die Kosten (Machbarkeit) oder die Forschungsfrage, die sich mit der Textkonversion verbindet. Gerade die Kostenfrage dürfte ein Schlüsselelement in der Bewertung von OCR-Verfahren bilden, denn offenbar gibt es in diesem Bereich einen *Pareto-Effekt*,

[119] http://www.digitale-sammlungen.de/~mdz/mdz/content/service/docs/2011-10-19_Brantl_Digitization_Lifecycle.pdf [Stand: 02/2013]

[120] http://www.slideshare.net/mdz-bsb/digitalisierungspraxis-thaller-volltextdigitalisierung?ref=http://mdzblog.wordpress.com/2011/10/page/2/

[121] Gemeint ist das Intensivlesen. Neuere Konzepte wie *hyperreading*, *machine reading* oder *distant reading* versuchen genau diesem Problem zu begegnen. Vgl. Stäcker, Thomas: Wie schreibt man Digital Humanities richtig? - Überlegungen zum wissenschaftlichen Publizieren im digitalen Zeitalter. In: Bibliotheksdienst 47 (2013) 1, S. 24-50.

[122] http://www.clarin.eu [Stand: 02/2013]

der dazu führt, dass der Wunsch nach hoher Textgenauigkeit die Kosten gerade für den Schritt zu sehr guten Texten sprunghaft ansteigen lässt. Um die Kosten und Qualität in eine angemessene Relation zu bringen, bedürfte es jedoch einerseits klarerer Vorstellungen darüber, was unter Qualität zu verstehen, andererseits Verfahren, wie diese Qualität zu messen ist. Beides ist derzeit nicht oder nur mit Einschränkung gegeben. So legen die genannten Zahlen aus der BSB zwar Genauigkeitsgrenzen fest, sagen aber weder, wie diese zu begründen sind, noch, wie sie gemessen werden. Obwohl es nicht Aufgabe des Wolfenbütteler Pilotprojekts Helmstedter Drucke Online I war, diese Aspekte systematisch zu untersuchen, sondern nur ein Praktikabilitätsszenario für die massenhafte OCR-Konversionen frühneuzeitlicher Drucke zu entwickeln, ergaben sich einige Beobachtungen, die es erlauben, sich der Frage anzunähern.

Die Bezifferung von Textgenauigkeiten, also der Übereinstimmung von Original und Kopie, ist der entscheidende Faktor in der Bewertung von OCR bzw. Konversionsverfahren. So simpel diese Aussage ist, so schwierig ist ihre Umsetzung und damit Ermittlung des „Werts" in der Praxis möglich. Hauptprobleme sind a) die Möglichkeit der Überprüfung (exakter Referenztext, so genannte „ground truth"), b) die Festlegung der Art der Genauigkeitsfeststellung (Buchstaben, Worte, Layout) und c) Normierung des Gegenstands (Digitalisierungsvorlage, Digitalisat) der Messung.

Zu a) Bei der hier bearbeiteten Materie, also universitäres Schrifttum vor allem des 17. und 18. Jahrhunderts liegen keine Referenztexte vor, von denen eine Differenzmenge gebildet und die absolute Genauigkeit des Konversionsergebnisses festgestellt werden könnte. Gleichwohl wurde versucht, die Genauigkeit der OCR zumindest näherungsweise zu ermitteln, indem im Laufe der ersten zwei Jahre des Projekts immer wieder Stichproben genommen wurden. Die Projektmitarbeiterin überprüfte jeweils einzelne Zeilen an definierten Stellen des Buchs, die bei kleineren Drucken den Seiten 4, 5 und 6, bei umfangreicheren den Seiten 10,15 und 20 entnommen wurden. Auf diese Weise wertete sie 35 Drucke aus. Gezählt wurde, wie gesagt, die korrekte Erkennung einzelner Buchstaben, nicht die von Wörtern. Die Genauigkeit lag im Schnitt bei diesen Werken bei 96%. Die Schwankungsbreite war vorlagenbedingt erheblich, die Genauigkeit fiel jedoch nie unter 90%. Bei optimaler Erkennung wurden 99%, in Einzelfällen sogar mehr erreicht. Angesichts des Umstands, dass das Projekt es mit ausserordentlich anspruchsvollen Materialien zu tun hatte (schlechte Druckvorlagen der Handpressenzeit, Mischung von Antiqua, Kursive und Fraktur), kann man das erzielte Ergebnis als durchaus bemerkenswert betrachten, da andere Softwarelösungen ein solches Ergebnis kaum erzielt hätten (vgl. hierzu die obigen Ergebnisse des Berliner Projekts). Allerdings ist ein Vergleich von BIT-Alpha mit Abbyy oder Omnipage, also einem lernenden mit einem statischen System, nur bedingt möglich, weil ein Versuchsaufbau „unter gleichen Bedingungen" implizit das statische System privilegiert, denn ein lernendes System setzt voraus, dass die Erkennungssoftware individuell für das jeweilige Objekt optimiert wird. Der eigentliche Vorteil des lernenden Systems wird damit relativiert, ist aber in einem solchen Vergleich unvermeidbar. Umgekehrt kann in Massenprojekten nur ein begrenzter Aufwand getrieben werden, um Software anzupassen, so dass die Frage nach der besseren Software auch ökonomisch entschieden werden muss (vgl. unten).

Ad b) ist festzulegen, was man als Fehler interpretiert. Sind z. B. nicht erkannte Layoutinformationen wie Marginalien, wenn sie zwar richtig erkannt, aber an der falschen Stelle in den Haupttext einsortiert wurden, ein Fehler? Bei den Stichproben wurden lediglich Buchstabengenauigkeiten, also das engere *pattern matching*, gewertet. Worttrennungen hingegen, die oft auch in der Vorlage nur „intellektuell" zu erkennen waren, blieben unberücksichtigt. Der Einsatz von Wörterbüchern wirkte sich bei der Worttrennung positiv aus, wie sie auch die Eregbnisse nachvollziehbar verbesserten, allerdings nur in Texten, wo es keine Mischung von deutscher Fraktur und lateinischer Antiqua gab.[123]

Systematisch untersucht werden konnte der Einsatz nicht, jedoch scheint es eine Art absolute Grenze zu geben (bei ca. 99%), jenseits der reines *pattern matching* bzw. Training keine signifikante Verbesserung mehr zeitigt. Dies liegt vor allem daran, dass bestimmte Buchstaben nicht mehr visuell, sondern nur intellektuell differenziert werden können (wie bereits erwähnt ist dies u. a. bei c und schlecht gedrucktem e häufig der Fall). Verbesserungen jenseits dieser Grenze sind demzufolge nur noch mit Werkzeugen der *intelligent character recognition* – vor allem durch den Einsatz von Wörterbüchern – möglich.

[123] Hier besteht sicher Verbesserungsbedarf und dem Vernehmen nach versucht die Firma über den Ansatz der Fontserkennung auch die Wörterbuchanwendung zu optimieren. In einem ersten Schritt hatte die Mischung von Wörterbüchern positive Effekte, jedoch konnte das noch nicht systematisch untersucht werden.

Ad c) waren die Grundlage entweder hochwertige TIFFs oder hochauflösende JPEGs auf der Basis der DFG-Praxisregeln. Der Binarisierungsprozess fand also auf der Basis von guten bis sehr guten Digitalisierungsvorlagen statt (es wurden keine Vorlagen vom Film oder in bitonaler Qualität verwendet). Dennoch konnte man auch hier nicht von gleichbleibenden bzw. Normbedingungen ausgehen, denn im Digitalisierungsprozess der HAB wird Rücksicht auf die Materialität der Bücher genommen. Wenn das Buch eine starke Wölbung im Papier aufweist, so wird nicht versucht, diese durch mechanische Maßnahmen (Anpressen) zu nivellieren. Gelegentlich vorkommende leichte Schräglagen des Buchblocks werden nicht ausgeglichen, so dass die Software auf heterogene Fälle trifft, die softwareseitig ausgerichtet werden müssen und negative Einflüsse auf die Erkennungsergebnisse haben.

Noch stärker kommt dieser Effekt mit Blick auf die Materialität der Objekte selbst zum Tragen (intrinsische Effekte). Die schon erwähnten Schattenbilder durch Widerdruck oder auch Unterstreichungen im Text beeinflussen die Konversionsergebnisse massiv. Gleiches gilt für Faktoren wie die Schrifttype, Satz und Papiergüte (Bräunung, Verschmutzung, etc.).

Nach knapp zwei Jahren Erfahrung in diesem Bereich kann man zwar sagen, welche Faktoren das Ergebnis beeinflussen, nicht jedoch wie diese im Schnitt gemessen werden sollen. Je nach Scanbeschaffenheit und Vorlage erreicht man bei einem Buch 99%, bei nächsten nur noch 95%, selbst wenn alle Software-, Trainings- und Wörterbuchparameter optimiert wurden. Nötig wäre hier eine Art Qualitätsindex für das Digitalisat und die Vorlage. Dabei ist nicht nur von Interesse, ob die Software „gut" oder „schlecht" arbeitet, sondern auch, mit welchen Genauigkeiten man bei welcher Art von Material rechnen kann. Mit anderen Worten, um eine bessere Vorhersage zu möglichen Qualitäten zu machen und Ergebnisse überhaupt bewerten zu können, wäre auch eine Typisierung der Vorlagen bzw. eine differenzierte Herangehensweise erforderlich.

Trägt man alle diese Faktoren zusammen, so wird deutlich, dass Genauigkeit nur über eine sehr komplexe Formel, wenn überhaupt präzise zu ermitteln ist. Selbst wenn man einen Referenztext hätte, könnte man nicht zuverlässig davon ausgehen, dass die Genauigkeit, die bei diesem Text erzielt wurde, auch auf alle anderen Texte übertragbar ist. Die Daten zur Genauigkeit, die im Laufe von Helmstedt I erhoben wurden, sind daher als Tendenz, nicht als absolute Größe zu verstehen und variieren, je nachdem, welchen Aspekt man hervorhebt. In diesem Sinne kann man aus den Erfahrungen davon ausgehen, dass das Ergebnis nach der Münchner Einteilung tendenziell befriedigend bis gut ist.

Dennoch war den Projeknehmern nach dieser ersten Phase klar, dass, obwohl Erfahrungen sich mit zunehmender Menge der Stichproben zwar nach dem Gesetz der großen Zahl konsolidieren lassen, eine zuverlässige Einschätzung erst auf der Grundlage methodisch zuverlässiger statistischer Verfahren möglich ist. Daher wurden die in der Folge beschriebenen statistischen Verfahren entwickelt, die, was die Überprüfung der gelieferten Genauigkeiten anlangt, im Folgeprojekt zum Einsatz kommen sollen.

8.7 Messung der Textgenauigkeit[124]

Die Frage der Messung der Textgenauigkeit ist von entscheidender Bedeutung für eine wissenschaftliche und ökonomische Bewertung der Textgüte und Textproduktion. Da wegen der Heterogenität des Materials eine Messung an Hand vorgegebener Muster und eine vollständige Überprüfung der Texte nicht möglich ist, bedarf es statistischer Verfahren, um mittels einer Stichprobe Genauigkeitswahrscheinlichkeiten bzw. Irrtumswahrscheinlichkeiten zu ermitteln.

Bei der Ermittlung von Genauigkeiten sind in Abhängigkeit von der Fragestellung zwei Verfahren in Betracht zu ziehen. Das erste Verfahren dient dazu, grundsätzlich die Güte einer Software zu testen, das zweite erlaubt die behauptete Genauigkeit eines Texts zu überprüfen.

[124] Die folgenden Überlegungen entstanden im Gespräch mit Herrn Burkard Rosenberger (Münster), der den Autor in den mathematischen Aspekten der Messung beraten und auch die genannten Verfahren entwickelt hat, wofür ihm herzlich gedankt sei.

8.7.1 Testen der Güte einer Software

Das hier beschriebene Verfahren kann dann zum Einsatz kommen, wenn es darum geht, die unbekannte Erkennungsquote einer Software abzuschätzen und dann die Güte dieser Schätzung zu bestimmen. Es wäre darin für Projektnehmer von Interesse, die den Einsatz einer bestimmten Software erwägen, oder aber auch für Hersteller von OCR-Software, um eine Erkennungsquote ihrer Software zu bestimmen.

Gegeben ist eine Grundgesamtheit von allen gescannten und OCR-erkannten Seiten aus mehreren Büchern. Die Zufallsvariable ist die Quote der pro Seite richtig erkannten Zeichen, wobei unterstellt wird, dass die Quote normalverteilt ist. Das ist realiter bei sehr heterogenen Materialien aus der Frühen Neuzeit zwar nicht immer der Fall, aber man kann von „typischen" Materialen ausgehen und sollte für die Stichprobe nicht Seiten wählen, die durch einen ungewöhnlich hohen Grad von Verschmutzung oder Annotationen bzw. Durchstreichungen das OCR-Ergebnis massiv verfälschen würden. Das Ziel der Ermittlung ist die Quote der Erkennungsgenauigkeit durch den Mittelwert einer Stichprobe abzuschätzen und die Güte der Schätzung zu bestimmen. Die Güte der Schätzung hängt von der Stichprobengröße und der Standardabweichung der Stichprobe ab.

1. Schritt: Wahl einer Konfidenzzahl. Diese bestimmt die gewünschte Wahrscheinlichkeit der Aussage.

Bsp. Konfidenzzahl $\gamma = 0.99$, d. h. mit einer Wahrscheinlichkeit von 99% liegt die wirkliche Erkennungquote im Intervall [x,y]. Die Konfidenzzahl gibt die Güte der Schätzung an.

2. Schritt: Erhebung einer Stichprobe, z. B. 10 Seiten aus einem Korpus druck- und zeittypischer Seiten. Die Auszählung der zehn Stichprobenseiten ergibt z. B. folgende Genauigkeitswerte: 0,93/0,87/0,99/0,95/0,95/0,94/0,9/0,8/0,87/0,92.

3. Schritt: Berechne Mittelwert und Standardabweichung dieser Stichprobe. Mittelwert hier 0,912 (Schätzgröße für die Erkennungsquote). Die Standardabweichung s beträgt 0,05411921 (je geringer, umso besser ist die Schätzung).

4. Schritt: Man ermittle c aus der Tabelle für die so genannte Stundentsche t-Verteilung.[125] Dies erfolgt mittels des Stichprobenumfangs minus 1 (n-1) und der Formel $0.5*(1+\gamma)$, indem n-1 die Zeile, das Ergebnis aus der Berechnung von $0.5*(1+\gamma)$ die Spalte der Tabelle der Studentschen t-Verteilung bilden. Aus der Spalte (hier $0.5*(1+0.99)=0.995$) und der Zeile (hier n=10-1 =9) ist also der Wert für c zu entnehmen (hier c= 3,25).

5. Schritt: Berechnung des Konfidenzintervalls a: Zusammen mit den Angaben zur Stichprobe (n) und der Standardabweichung (s) kann nun mit Hilfe der Formel $a = c*s / \sqrt{n}$ das Konfidenzintervall berechnet werden ($a= 3,25*0,05411921/\sqrt{10} = 0,05562049$)

Als Ergebnis der Berechnung kann festgestellt werden, dass mit einer 99%igen Wahrscheinlichkeit das Ergebnis im Intervall [Mittelwert-a, Mittelwert+a], (hier [0,912-0,05562049;0,912+0,05562049]) also im Bereich zwischen 0,85637951 und 0,96762049 liegt. Wir haben hier also noch eine recht große Schwankungsbreite, was vor allem an der relativ geringen Stichprobengröße liegt. Mit anderen Worten, je größer die Stichprobe und je geringer die Standardabweichung, umso schmaler wird dieser Schwankungsbereich werden. Natürlich kann man auch mit einer abweichenden Konfidenzzahl, z. B. 80%, rechnen, doch scheint dies für eine möglichst genaue Bewertung des OCR-Ergebnisses nicht zielführend.

[125] http://de.wikipedia.org/wiki/Studentsche_t-Verteilung [Stand: 02/2013]

8.7.2 Überprüfen der Güte eines Texts

Neben der Frage nach der unbekannten Güte der Erkennungsquote der Software ist vor allem die Frage nach der behaupteten Güte des Texts von Bedeutung, wenn Arbeitsergebnisse von Dienstleistern überprüft werden müssen. Auch hier verbietet die Textmenge aus pragmatischen Gründen eine Auszählung bzw. Überprüfung und erfordert ein statistisches Verfahren zur Ermittlung der Genauigkeit. Es geht dabei darum, anhand einer Stichprobe zu überprüfen, ob die vom Dienstleister behauptete Erkennungsquote stimmt, wobei man die Wahrscheinlichkeit für einen eigenen Irrtum möglichst gering halten möchte. Das ist ein sog. Bernoulli-Experiment, und die zugehörige diskrete Verteilung ist die Binominalverteilung. Dafür gibt es einige markante „Grenzen" in Abhängigkeit von Stichprobengröße und akzeptierter Irrtumswahrscheinlichkeit.

Beispiel: Ein Anbieter behauptet, dass seine Software mindestens 90% der Zeichen korrekt erkennt. Wenn man nun eine Stichprobe erhebt und darin exakt 90% der Zeichen korrekt erkannt werden, kann man nicht ohne weiteres davon ausgehen, dass die Aussage des Anbieters auch wirklich stimmt, denn jedes Ergebnis einer Stichprobe streut um den errechenbaren Mittelwert. Es könnte also durchaus sein, dass die Erkennungsquote bei 85% liegt und nur zufällig in der Stichprobe ein Ausreißer nach oben vorlag. Tatsächlich liegt die Wahrscheinlichkeit, dass diese Aussage zutrifft, bei lediglich 50%.

Es erhebt sich nun die Frage, wie selten solche Ausreißer in Abhängigkeit von der Stichprobengröße sind, d. h. welche Irrtumswahrscheinlichkeit man bereit ist zu akzeptieren und welche Stichprobengröße mindestens einzuplanen ist, um ein akzeptables Ergebnis zu erlangen. Es leuchtet unmittelbar ein, dass die Irrtumswahrscheinlichkeit abnimmt, je größer die Stichprobe wird, jedoch möchte man die Stichprobe so klein wie möglich und nötig halten, um den Aufwand des Auszählens zu optimieren bzw. überhaupt einen pragmatischen Zugang zu einer soliden Abschätzung zu eröffnen. Dabei muss man eine plausible Balance finden zwischen akzeptierter Irrtumswahrscheinlichkeit und Größe der Stichprobe.

Da die Berechnung mathematisch kompliziert ist,[126] seien im Folgenden mehrere Tabellen mit fest vorgegebenen Werten aufgeführt, die typische Angaben enthalten. Vorgeschlagen wird eine Stichprobengröße von 500 beliebigen Zeichen, die idealerweise durch Benutzung eines Zufallsgenerators ausgewählt werden. Unter dieser Voraussetzung gilt folgende Tabelle. In der linken Spalte ist die behauptete Erkennungsquote angegeben, in der rechten die Zahl der in der Stichprobe mindestens korrekt erkannten Zeichen, die vorliegen muss, um die Behauptung des Dienstleisters bei einer Irrtumswahrscheinlichkeit von 2,5% als korrekt einstufen zu können. Wenn also ein Dienstleister behauptet, dass ein Text eine Genauigkeit von 96% hat, müssen in der Stichprobe von 500 Zeichen mindestens 489 Zeichen korrekt erkannt werden, damit bei einer Irrtumswahrscheinlichkeit von 2,5% die Behauptung des Dienstleisters akzeptiert werden kann.

[126] Zur Binominalverteilung vgl. das Skript von Albers/Yanik: http://www.math.uni-bremen.de/didaktik/ma/ralbers/Veranstaltungen/Stochastik11/Material/ScriptK7_110703.pdf [Stand: 02/2013]

Behauptete Erkennungsquote	Mindestzahl der korrekt erkannten Zeichen (Stichprobengröße = 500)
95%	485 (96,9%)
96%	489 (97,8%)
97%	493 (98,5%)
98%	496 (99,3%)
99%	499 (99,9%)
>99%	500 (100%)

Zum Vergleich eine Tabelle mit einer Stichprobengröße von 2.000 Zeichen:

Behauptete Erkennungsquote	Mindestzahl der korrekt erkannten Zeichen (Stichprobengröße = 2000)
95%	1920 (96%)
96%	1938 (96,9%)
97%	1956 (97,8%)
98%	1972 (98,6%)
99%	1988 (99,4%)
>99%	2000 (100%)

Bei einer behaupteten Erkennungsquote von über 99% scheint es sinnvoll, ggf. die Stichprobengröße zu erhöhen. Nachstehend zwei Tabellen, die beispielhaft zeigen, wie hoch die ermittelte Erkennungsquote in Abhängigkeit von einer bestimmten Stichprobengröße sein muss, wenn Texte eine behauptete Erkennungsquote von 99,5% bzw. 99,7% haben. Die Irrtumswahrscheinlichkeit liegt jeweils bei 2,5%:

Behauptete Genauigkeit: 99,5%	
Stichprobengröße	Mindestzahl der korrekt erkannten Zeichen
500	500
1000	999
2000	1996
5000	4985
10000	9960

Behauptete Genauigkeit: 99,7%	
Stichprobengröße	Mindestzahl der korrekt erkannten Zeichen
500	500
1000	1000
2000	1998
5000	4995
10000	9990

Abschließend eine Tabelle mit einer höheren Irrtumswahrscheinlichkeit von 5% statt 2,5%:

Behauptete Erkennungsquote	Mindestzahl der korrekt erkannten Zeichen (Stichprobengröße = 2000)
95%	1916 (95,8%)
96%	1934 (96,7%)
97%	1952 (97,6%)
98%	1970 (98,5%)
99%	1988 (99,4%)
>99%	2000 (100%)

Die Frage, welche Irrtumswahrscheinlichkeit man sinnvollerweise zugrunde legt, kann nicht verbindlich beantwortet werden. Erfahrungswerte zeigen, dass 5% eine sinnvolle Größe darstellt, die nicht überschritten werden sollte. Da der Mehraufwand der Abschätzung bei 2,5% gegenüber 5% gering ist, sollte eine Irrtumswahrscheinlichkeit von 2,5% zugrunde gelegt werden, um ein verlässliches Ergebnis zu erhalten.

Ein Excelsheet, mit dem man weitere Angaben durch Eingabe alternativer Parameter berechnen kann, findet sich auf der Projektseite der Herzog August Bibliothek.[127] Eine Schwierigkeit besteht darin, ein pragmatisches Verfahren zur Auszählung zu finden. Zunächst muss die Stichprobe per Zufallsprinzip erhoben werden. Nutzen kann man dafür die in Programmiersprachen meist vorhandenen Zufallsgeneratoren. Um echte Zufallszahlen[128] zu generieren, lassen sich geeignete Webservices nutzen.[129] Um die Auszählung zu erleichtern, entwickelt die Herzog August Bibliothek derzeit für die Messung von Genauigkeiten einen Webservice, der im zweiten Teil des Projekts, Helmstedter Drucke online II, zum Einsatz kommen soll.

8.8 Ökonomische Betrachtungen

Die Kosten für die Konversion einer Seite können beträchtlich schwanken. Bei der Nutzung von Standard-OCR-Software liegen die sächlichen Kosten nur in der Beschaffung der Software. Die Personalkosten für die Auswahl, Vorbereitung und Prozessverwaltung müssen jedoch berücksichtigt werden. Gerade bei Massenkonversionsprozessen ist die Etablierung fester Workflows wichtig und der Einsatz von leistungsfähiger Hardware, denn der OCR-Prozess ist sehr rechenintensiv und daher bei schlechter Hardwareausstattung sehr zeitaufwendig und mit Kosten verbunden. Insofern ist es eine irrige Annahme, man könne mit Standardsoftware nahezu kostenneutral arbeiten.

Dies vorangeschickt, ist an dieser Stelle die grundsätzliche Frage zu stellen, welche Qualität erforderlich ist und wie viel sie kosten darf. Denn ohne eine Qualitätsanforderung zu stellen, sind auch differenzierte ökonomische Betrachtungen wertlos; die Entscheidung wäre dann einfach und müsste immer für die günstigste Variante ausfallen. Wenn aber die günstigste Variante nicht die Qualität zu liefern vermag, die mindestens erforderlich ist, dann ist das hier aufgewendete Geld schlicht verschwendet und selbst die günstigste Variante noch zu teuer. Sich diese wirtschaftlichen Gemeinplätze immer wieder zu vergegenwärtigen und Aspekte von Effektivität und Effizienz in die Diskussion um die Qualität von OCR-Software einzubringen, ist dringend erforderlich, wenn man realistische Szenarien für eine massenhafte Konversion von Drucken der Frühen Neuzeit in maschinenlesbare Texte entwickeln will.

Den Ausgang für solche Überlegungen sollte man bei einer Festlegung der Zwecke nehmen, für die maschinenlesbare Texte verwendet werden sollen und für die bestimmte Qualitäten notwendig sind.

[127] Siehe: http://www.hab.de/de/home/wissenschaft/projekte/helmstedter-drucke-online.html [Stand: 02/2013]
[128] http://de.wikipedia.org/wiki/Zufallszahl [Stand: 02/2013]
[129] Z. B. https://www.random.org/ [Stand: 02/2013]

Leider gibt es bislang hierzu in der wissenschaftlichen *community* keine normativen Aussagen. Sicher kann man davon ausgehen, dass ein wissenschaftlich brauchbarer Text, also ein Text der sowohl als Lesefassung als auch für zuverlässige Recherchen *ex silentio*[130] taugt, eine Genauigkeit von mindestens 99,95%[131] besitzen muss, d. h. man nimmt pro Seite im Schnitt etwa 1-2 Buchstabenfehler in Kauf. Texte dieser Güte lassen sich für Materialien der Handpressenzeit (15. Jh. bis 18. Jh.) derzeit nur durch Abschreiben (*double keying*) erreichen. Die Kosten schwanken zwischen 1,20 und 2,20 € pro Seite, wobei ggf. weitere Dienstleistungen wie die XML-Grundkodierungen der Dokumente hinzukommen. Ginge man davon aus, dass z. B. im 17. Jh. etwa 200.000 Drucke konvertiert werden sollten und legte man eine Durchschnittszahl von 140 Seiten zugrunde,[132] müssten für die 28 Mio Seiten mindestens 33,6 Mio Euro aufgewendet werden. Selbst wenn eine solche Qualität wünschbar wäre, scheint sie doch angesichts der hohen Kosten unrealistisch, ja nicht einmal nötig, da man bei ausgewählten Materialien auch mit einer geringeren Qualität arbeiten könnte. Doch welche Qualität ist ausreichend?

Standard-OCR-Software führt, wie man weiss, bei Materialien der Frühen Neuzeit zu hohen Fehlerquoten und die Qualität sinkt bei den hier betrachteten Materialien deutlich unter den Wert, den man nach der Münchner Skala als „schlecht" bezeichnen könnte. Da es im Bereich des „Schlechten" aber durchaus noch einen Bereich des „Nützlichen" gibt, bleibt diese Qualifizierung abhängig von der Frage, welchen Gebrauch man von dieserart Texten machen möchte. Auch Aspekte der nachträglichen Verbesserungen (semantische Analyse, Wörtbücher, *crowdsourcing*) stehen unter diesem Vorbehalt.[133] Darauf nach dem derzeitigen Stand des Wissens eine angemessene und für alle wissenschaftlichen Felder gleichermaßen zufriedenstellende Antwort zu finden, dürfte schwer fallen. Dennoch kann man Tendenzen feststellen, die zumindest eine grobe Orientierung bieten.

Klammert man editorische Ansprüche aus, die eigene Anforderungen mitbringen,[134] dürfte der maßgebliche Gebrauch, der von Volltexten in wissenschaftlichem Kontext gemacht wird, die Suche sein. Demgegenüber scheint die Lesbarkeit eines Texts minder wichtig, zumindest dann, wenn die Alignierung des digitalen Faksimiles mit dem Volltext gewährleistet ist. Das digitale Faksimile übernimmt hierbei die Funktion der Lesefassung. Ein in den Geisteswissenschaften noch wenig wahrgenommener Nutzen liegt in quantitativen Analyseverfahren und der semantischen Verarbeitung maschinenlesbarer Texte, die innerhalb der voranschreitenden *digital humanities* entwickelt werden. Auch diese bringen einen eigenen Qualitätsanspruch mit.

Wenn man die Suche als das derzeit wichtigste Instrument herausgreift und fragt, welche Textgüte einer wissenschaftlichen Anforderung genügt, lassen sich zwei Typen von hauptsächlichen Suchverfahren unterscheiden, nämlich phraseologische und Suchen nach Entitäten oder Stichwörtern. Phraseologische Suchen nach bestimmten sprachlichen, aus mehreren Wörtern bzw. aus längeren Phrasen bestehenden Textteilen können bei einer Qualität, die nicht bei 99,95% liegt, vermutlich nicht sinnvoll durchgeführt werden oder die Ergebnisse sind stark zufallsbehaftet. Anders liegt der Fall bei Suchen nach Entitäten oder einzelnen Stichwörtern. Das ist auch typischerweise das, was Forscher bei Google und ähnlichen Suchmaschinen suchen. Entitäten- oder in einem weiteren Sinn Einwortsuchen – sind auch in Texten möglich, die man als „schlecht" bezeichnen könnte, indem das gesuchte Wort „zufällig" richtig erkannt wurde. Allerdings lauern hier Gefahren der Fehleinschätzung des Suchergebnisses. Die Wahrscheinlichkeit bei Texten der Frühen Neuzeit einen Treffer zu erzielen, ist bei handelsüblicher Software stark eingeschränkt, denn eine Erkennungsgüte von unter 90%, die hier nicht untypisch ist, bedeutet, dass ca. jedes vierte Wort falsch ist. Das klingt zunächst nicht dramatisch, weil es ja auch bedeutet, dass drei Wörter richtig sind und man z. B. einen Namen sehr wohl finden könnte, jedoch muss man in Betracht ziehen, dass diese Fehler sich nicht statistisch antizipieren lassen und möglicherweise nur „uninteressante" Wörter korrekt sind. Problematisch in Texten der Frühen Neuzeit sind Mischtexte, in denen bei dieserart Software nur Antiqua oder nur Fraktur korrekt erkannt werden, so dass gerade die

[130] D. h. dass man ein negatives Rechercheergebnis so werten kann, dass ein bestimmter Begriff nicht vorkommt.
[131] Hier und im Folgenden ist von Buchstabengenauigkeit die Rede.
[132] Vgl. Anm. 118
[133] Vgl. dazu die Ergebnisse des IMPACT Projekts (http://www.digitisation.eu/) [Stand: 02/2013]
[134] Zu den Wünschen von Nutzern, die an eine elektronische Edition gestellt werden, vgl. Sören A. Steding, Computer-based scholarly editions : context, concept, creation, clientele. Berlin 2002, S. 234.

interessierenden deutschen oder lateinischen Namensformen nicht gefunden werden. Beide Umstände lassen die Wahrscheinlichkeit eines Treffers mit handelsüblicher Software weiter sinken. Die Gefahr der Fehleinschätzung des Suchergebnisses besteht darin, dass der Suchende entweder glaubt, dass ein Text einen Namen nur singulär nennt, wo er in Wahrheit allgegenwärtig ist, oder er meint, dass bei einem negativen Resultat der Name, wenn er vorkommt, doch nur selten vorkommt, denn andernfalls hätte man ihn doch wenigstens einmal finden müssen. Um diese Effekte abzufedern, hilft die Forderung der Praxisregeln der DFG, schmutziges OCR nicht zu verbergen, weil sichtbar wird, wie gut oder schlecht der Text tatsächlich ist, doch kann sie dieserart Fehleinschätzungen nicht grundsätzlich verhindern bzw. der Suchende kann eigentlich nie zuverlässig davon ausgehen, dass das, was ihm als Ergebnis vorliegt, tatsächlich so zu interpretieren ist, wie es sich zeigt. Mit anderen Worten, der Fund ist eigentlich nutzlos oder nur in einem sehr oberflächlichen Sinne hilfreich, indem man argumentiert, dass es immer besser ist, etwas zu haben, als nichts.

Zusammenfassend scheint daher, dass der Nutzen einer wissenschaftlich motivierten Volltextsuche mit schmutzigem OCR oder „schlechtem Text" (< 90% nach der Münchner Skala) gering bzw. sogar fragwürdig ist, weil es nahezu unmöglich ist, Treffer zu kontextualisieren und korrekt zu bewerten, – wenn man sich nicht die Mühe macht oder aus zeitlichen Gründen machen kann, das Dokument, aus dem der Fund stammt, zu lesen. Positive Treffer bleiben zufällig und das Risiko der Fehlinterpretation ist hoch, auch wenn natürlich ein Tauchgang die eine oder andere Perle zutage zu fördern vermag. Angesichts dieser Grundüberlegungen scheint die Bereitstellung von Volltext frühneuzeitlicher Texte unterhalb eines bestimmten Qualitätsniveaus als nicht sinnvoll, zumindest für den Bereich der direkten Suche. Statistische oder probabilistische Verfahren, wie sie sich mit Konzepten aus den digital humanities verbinden (z. B. Google ngram Viewer), folgen dabei anderen Gesetzmäßigkeiten, weil hier Mängel im Detail durch statistische Masse ausgeglichen werden. Welche Qualität in dieserart quantitativen Projekten ausreichend ist, liesse sich mit Hilfe statistischer Verfahren sicher ermitteln, möglicherweise wäre ein Nutzen auch noch bei „schlechten" Texten gegeben. Erfahrungen geschweige denn Studien dazu gibt es bisher kaum, so dass an dieser Stelle nicht näher darauf eingegangen werden kann.

Obzwar deutlich scheint, dass eine Qualität unterhalb von 90% für die Suche unzureichend ist, bleibt schwer zu entscheiden, welche Qualität für welche Zwecke zwischen 90% und 99,95% benötigt wird und welche Kosten angemessen sind. Mit den von B.I.T. Tomasi nach händischen Ermittlungsmethoden erhobenen 96% ist sicher ein sinnvolles Mittelmaß zwischen Qualitätsanspruch und bereitgestellter Menge für wissenschaftliche Zwecke erreicht, die preislich etwa in der Mitte zwischen einer Standard-OCR-Lösung[135] und einer händischen Transkription liegt. In der Projektplanung kommt es also darauf an zu entscheiden, ob man Texte im Umfang mit einer Qualität von 96% oder alternativ weniger Texte im Umfang von ca. 1/3 auf einem sehr hohen Niveau per händischer Abschrift zur Verfügung stellt. Im Falle der Helmstedter Drucke, bei denen es sich überwiegend um serielles Material (serielles Universitätsschrifttum mit rekursiven Phänomenen) handelt, wäre eine solche Umfangsreduktion nicht zu rechtfertigen, mag es auch in anderen Fällen, z. B. herausragende Lexika, literaturwissenschaftlich oder historisch zentrale Werke u.ä., unabweislich erforderlich sein. Hier muss gelten, das richtige, meint effizienteste Werkzeug zu gebrauchen, wo also Kosten und Nutzen miteinander in Einklang stehen. Dieser Mittelweg scheint mit der formulierten Qualitätsanforderung und dem hier verfolgten Konversionsverfahren gefunden worden zu sein, auch wenn perspektivisch die Kosten für das Verfahren noch sinken müssen, wenn es in noch größerem Maße wirtschaftlich eingesetzt werden soll.

[135] Vgl. z. B. das Angebot der VZG in Göttingen: http://www.gbv.de/Verbundzentrale/serviceangebote/ocr-service-der-vzg [Stand: 02/2013]

9 Register

Der vorliegende Werkstattbericht aus der Abteilung Historische Drucke der Staatsbibliothek zu Berlin – Preußischer Kulturbesitz beschreibt Testszenarien zur automatischen Texterkennung (OCR) von frühneuzeitlichen Funeralschriften. In einem von der Deutschen Forschungsgemeinschaft (DFG) geförderten Pilotprojekt wurden Optimierungs- und Konfigurationsmöglichkeiten zweier ausgewählter Softwareprodukte getestet, um gattungsspezifische Optionen der Volltextgenerierung auszuloten. Neben den Ergebnissen dieser Tests werden die Vor- und Nachteile der verschiedenen Softwarelösungen dargestellt sowie allgemeine Anforderungen an die OCR Alter Drucke formuliert. Ein Erfahrungsbericht aus dem Projekt „Helmstedter Drucke Online“ der Herzog August Bibliothek Wolfenbüttel ergänzt die Abhandlung. Thomas Stäcker beschreibt darin auch Methoden der Ermittlung von Textgenauigkeiten und geht auf ökonomische Aspekte der Volltexterstellung Alter Drucke ein.

ISBN 978-3-88053-185-7