Digital Humanities

Fritz Kliche* und Ulrich Heid

Ein einfacher und transparenter Zugang zur sprachtechnologischen Arbeit mit Textdaten: Beispiele aus dem Projekt *e-Identity*

DOI 10.1515/iwp-2015-0059

Zusammenfassung: Die Computerlinguistik hat verschiedene Werkzeuge für die Textanalyse entwickelt, die allerdings aus der Sicht geisteswissenschaftlicher Nutzer oft nicht unmittelbar anwendbar sind. Wir beschreiben in diesem Beitrag eine Web-Anwendung, mit der digitale Textsammlungen möglichst einfach für die Arbeit mit computerlinguistischen Methoden zugänglich gemacht werden können, und wir diskutieren die Konzepte, mit denen dieser einfache Zugang erreicht wird.

Deskriptoren: Computerlinguistik, Korpuserstellung, semi-strukturierte Daten

Easy and transparent access to language technological work on text data: Examples from the project *e-Identity*

Abstract: Computational linguistics has developed several tools for text analysis. However, users from the humanities are often confronted with practical obstacles that make it hard to apply these tools immediately. In this article, we describe a web application that allows users to apply computational linguistic methods to existing text collections in an easy way, and we discuss the concepts underlying this easy access.

Descriptors: computational linguistics, corpus building, semi-structured data

Un accès simple et transparent aux travaux d'ingénierie linguistique appliqués aux données textuelles: exemples du projet *e-Identité*

Résumé: La linguistique informatique a développé plusieurs outils pour l'analyse de texte, mais du point de vue des utilisateurs des sciences humaines, ceux-ci ne sont souvent pas immédiatement applicables. Dans cet article, nous décrivons une application web qui permet de rendre les collections de textes numériques facilement accessibles pour le travail avec des méthodes de linguistique informatique, et nous discutons les concepts permettant cet accès facilité.

Descripteurs: linguistique informatique, création de corpus, données semi-structurées

***Kontaktperson: Fritz Kliche,** Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Pfaffenwaldring 5b, 70569 Stuttgart, Deutschland, E-Mail: klichefz@ims.uni-stuttgart.de
Prof. Dr. Ulrich Heid, Institut für Informationswissenschaft und Sprachtechnologie, Universität Hildesheim, Universitätsplatz 1, 31141 Hildesheim, Deutschland, E-Mail: heid@uni-hildesheim.de

1 Der Zugang zur digitalen Textwissenschaft

Textsammlungen sind eine wichtige Quelle für die Arbeit in den digitalen Geisteswissenschaften. Mit computerlinguistischen Techniken können Textdaten in ihrer gesamten Breite untersucht werden (z. B. im Sinne von *distant reading*; vgl. Moretti, 2013). Sie ermöglichen quantitative Auswertungen, erkennen inhaltliche Konzepte und erlauben die Arbeit mit reichhaltigen Annotationen, z. B. in digitalen Editionen. Durch die zunehmende Digitalisierung sind große Mengen von Text auch leicht verfügbar und austauschbar. Aber computerlinguistische Techniken können nicht auf alle Textsammlungen unmittelbar angewendet werden. Die Werkzeuge benötigen z. B. feste Eingabeformate oder müssen in eine Verarbeitungspipeline eingebunden werden, wie z. B. in GATE (Cunningham et al., 2002) oder UIMA[1]. Die Daten müssen dazu in die passende maschinenlesbare Form überführt werden. Diese Erschließung erfolgt üblicherweise über Programmierarbeit, über die Definition von regulären Ausdrücken oder den Aufbau einer Datenbank. Solche Aufgaben sind für geisteswissenschaftliche Nutzer oft eine Hürde beim Einsatz computerlinguistischer Techniken – obwohl diese Techniken letztlich für sie entwickelt wurden. Hammond

1 https://uima.apache.org

et al. (2013) sprechen von einer „kulturellen Barriere“ zwischen Computerlinguisten und – in ihrem Fall – Literaturwissenschaftlern. Es braucht also Konzepte, die die Barriere zu überwinden helfen, z. B. dadurch, dass den Geisteswissenschaftlern Werkzeuge an die Hand gegeben werden, mit denen sie ohne Programmieraufwand Texte für die Anwendung von computerlinguistischen Techniken vorbereiten können.

Textsammlungen sind in der Regel nicht völlig unstrukturiert. Sie lassen sich in Einheiten (je nach Sammlung z. B. in einzelne Zeitungsartikel, Wörterbucheinträge, Gedichte) segmentieren, die meistens in einem festen, über die Textsammlung hinweg einheitlichen Datenschema repräsentiert werden. Die Elemente des Datenschemas (z. B. Metadaten wie Datumsangaben) sind für den menschlichen Leser meist an einheitlichen Indikatoren auf den ersten Blick erkennbar. Sie können z. B. mit einer festen Zeichenfolge (im Folgenden: *Ankerwörter*) markiert werden (wie z. B. „DATUM:“); oder sie sind an einer einheitlichen Position erkennbar (z. B. die letzte Zeile jeder Einheit). Dennoch müssen solche Elemente erst in ein maschinenlesbares Format überführt werden, damit gezielt auf sie zugegriffen werden kann. Solche Daten werden als semi-strukturierte Daten bezeichnet: Die Struktur ist implizit in den Daten selbst erkennbar, aber das Datenschema ist nicht explizit gemacht oder stellt nur schwache Anforderungen an die Daten (Buneman, 1997). Ziel der hier vorgestellten Arbeiten ist es, die Datenstrukturen solcher semi-strukturierten Materialien nachzuzeichnen und, soweit notwendig, zu dokumentieren.

Wir stellen in diesem Beitrag eine Web-Anwendung vor, mit der semi-strukturierte Textsammlungen in ein strukturiertes Format überführt werden können. Als Textsammlungen bezeichnen wir dabei Textdaten, die in zumindest teilweise gleichförmig strukturierte Einheiten eingeteilt werden können. Die Elemente der Einheiten (z. B. Fließtextabschnitte oder Metadaten) müssen über einheitliche Indikatoren beschrieben werden können. Dieser Fall ist häufig und tritt bei den unterschiedlichsten Arten von Textdaten auf. Als Beispiele verwenden wir neben Texten aus dem Projekt *e-Identity* eine kleine Sammlung mit Fabeln und Gedichten aus dem Gutenberg-Archiv; Transkripte von Radio-Interviews; XML-Dateien eines Herstellers für Kopiergeräte, aus denen für Übersetzer relevante Textstellen extrahiert werden mussten; und Handbücher für Heimwerker. Die Nutzer der Web-Anwendung können die Textsammlung erst in *strukturelle Einheiten* segmentieren, aus denen sie Elemente wie Fließtextabschnitte oder gleichförmig dargestellte Metadaten als *Textobjekte* extrahieren. Für die Extraktion beschreiben die Nutzer Indikatoren für die Textobjekte, z. B. Ankerwörter oder bestimmte Positionen in der strukturellen Einheit. Die Textobjekte werden mit einem Bezeichner versehen und in einer Datenbank als Korpusprojekt abgelegt. Die Nutzer können das Korpus um Rauschen bereinigen. Sie können das Korpus um Dubletten (identische Textobjekte) und Semi-Dubletten (ähnliche Textobjekte) bereinigen. Weiter können leere und zu kurze Textobjekte gefiltert werden, indem eine minimale Länge für Textobjekte angegeben wird. Schließlich können Textobjekte mit einem erhöhten Anteil an nicht-alphanumerischen Zeichen oder einem erhöhten Anteil an Zahlzeichen als Textobjekte ohne Fließtext gefiltert werden. Die Daten können in verschiedenen Formaten exportiert werden.

Ohne dass bisher in größerem Umfang Benutzerstudien oder Prinzipien des *user-centered design (UCD)* angewendet worden wären, ist trotzdem das mittelfristige Ziel, die o. g. Schritte für die Nutzer intuitiv bedien- und nachvollziehbar zu machen. Wir diskutieren in diesem Beitrag Ansätze dafür, wie ein einfacher Zugang zu den computerlinguistischen Anwendungen erreicht werden soll. Zum Beispiel kann die Web-Anwendung Arbeitsschritte automatisieren – damit reduzieren sich aber die Kontrollmöglichkeiten über die Verarbeitung. Die importierten Inhalte können in ein einheitliches Format überführt werden, damit die Anwendung generisch für mehrere Formate verwendet werden kann – damit werden aber nicht alle Informationen der Quellformate in der Datenbank abgebildet.

Abschnitt 2 nennt angrenzende und relevante Arbeiten. Abschnitt 3 stellt das Projekt *e-Identity* vor, in dessen Rahmen die Web-Anwendung entwickelt wurde. In Abschnitt 4 geben wir einen Überblick über die Funktionen der Web-Anwendung. In Abschnitt 5 diskutieren wir die Konzepte, mit denen wir den einfachen Zugang zur Arbeit mit Textsammlungen erreichen möchten. In Abschnitt 6 beschreiben wir anhand unterschiedlicher Textdaten verschiedene Anwendungsfälle für die Web-Anwendung.

2 Ähnliche Arbeiten

Die Web-Anwendung möchte sich mit dem Schwerpunkt auf der Aufbereitung beliebiger semi-strukturierter Daten von bestehenden Angeboten für die Digital Humanities abgrenzen. Brooke et al. (2015) stellen ihr System *GutenTag* vor, das einen einfachen Zugang zur Arbeit mit computerlinguistischen Techniken im Archiv Gutenberg anbietet, das aber für dieses Archiv und seine Formatkonventionen spezifisch ist. GATE (General Architecture for Text Engineering; Cunningham et al., 2002) ist eine Arbeitsumgebung zur Korpusanalyse und zum Text Mining, mit der Korpora erstellt und linguistisch verarbeitet werden können. Im *Text Technology Lab* der Universität Frank-

furt wurden u. a. der *eHumanities Desktop* (Mehler et al., 2011) und der *TTLab Preprocessor* (Gleim und Mehler, 2015) entwickelt. Der *eHumanities Desktop* ist eine Arbeitsumgebung zur Korpusanalyse, die u. a. Tools zur Ressourcenverwaltung, zur Annotation von Metadaten und zu quantitativen Analysen umfasst. Über den *TTLab Preprocessor* können Daten computerlinguistisch vorverarbeitet werden (z. B. Lemmatisierung, Annotation von Wortarten, Konvertierung in TEI P5[2]). Diese Anwendungen sind generisch auf verschiedene Datenformate anwendbar, enthalten aber keine Funktion zur Nachzeichnung des Datenschemas semi-strukturierter Daten.

Aus technischer Sicht untersucht die Mehrheit der Arbeiten zu semi-strukturierten Daten, wie der Aufbau von HTML- oder XML-Dateien über verschiedene Verfahren zum maschinellen Lernen erkannt werden kann. Werkzeuge für die Extraktion von Einträgen auf Webseiten werden als *Wrapper* bezeichnet; der Prozess, solche Einträge an einem einheitlichen Aufbau über mehrere Webseiten hinweg zu erkennen, wird als *Wrapper Induction* bezeichnet. Zheng et al. (2009) nennen einige Beispiele für relevante Arbeiten; Fiumara et al. (2007) ist ein Survey-Papier.

3 Projektkontext: Das Verbundprojekt *e-Identity*

Die Web-Anwendung wurde im Rahmen des BMBF-Projekts *e-Identity* als die *Explorationswerkbank* entwickelt. Das Gesamtprojekt beschreiben Blessing et al. (2013; 2015). Die Implementierung der Explorationswerkbank wird in Kliche et al. (2014) dargestellt. In *e-Identity* entwickelten Computerlinguisten Werkzeuge für die Forschungsfragen ihrer Projektpartner aus der Politikwissenschaft. Die Politikwissenschaftler waren an unterschiedlichen Identitätskonzepten interessiert, wie z. B. eine gemeinsame europäische, transatlantische oder nationale Identität. In einem großen Korpus von Zeitungsartikeln (>800.000 Artikel) aus dem Zeitraum 1990–2012 untersuchten sie, an welche Identitäten in Diskursen zu internationalen Krisensituationen und militärischen Interventionen appelliert wurde. Ein Ziel war es z. B., die Medienpräsenz bestimmter Identitätskonzepte entweder für einen Zeitraum (z. B. ein Jahr) oder im zeitlichen Verlauf (z. B. für jedes Jahr im Korpus) zu erfassen („Issue Cycles"): Welches Identitätskonzept trat innerhalb eines Jahres besonders hervor? Wie entwickelte sich dessen Medienpräsenz im Zeitverlauf (vgl. z. B. Overbeck, 2014)? Für das Korpus wurden aus verschiedenen Volltextdatenbanken (darunter *LexisNexis* und *Factiva*) über repräsentative Keywords Zeitungsartikel heruntergeladen und in ein einheitliches und bereinigtes Korpus überführt. Wegen der Lizenz-Bestimmungen der Volltextdatenbanken ist das Korpus nicht frei verfügbar.

Für die Operationalisierung der Identitätskonzepte wurde von den Projektpartnern der *Complex Concept Builder* entwickelt (vgl. Blessing et al., 2015). Das Korpus wurde außerdem um Artikel bereinigt, die nicht im Themengebiet von internationalen Krisensituationen lagen (*Off-Topic*-Artikel); z. B. enthalten auch Sportberichte „kriegerische" Metaphern („Angriff", „Gegner", „Verteidigung") und flossen dadurch in das Sample ein. Die Projektpartner klassifizierten dafür die Artikel über das Topic-Analyse-Verfahren Latent Dirichlet Allocation (LDA; Blei et al., 2003). Das Verfahren gibt für jeden Text („Dokument") Werte an, wie stark er jeweils verschiedenen Themen zugeordnet wird. Gleichzeitig werden für die Tokens der Sammlung Werte angegeben, wie repräsentativ sie für die erkannten Themen sind. Anschließend konnten *Off-Topic*-Artikel treffsicher aus dem *e-Identity*-Korpus ausgeschlossen werden, indem über besonders repräsentative Tokens zu verschiedenen *Off-Topics* ein Klassifikator trainiert wurde (Dick, 2015).

4 Übersicht über die Explorationswerkbank

Die Explorationswerkbank wurde als Web-Anwendung konzipiert, damit Installationsschritte oder die manuelle Einbindung sprachtechnologischer Werkzeuge für die Nutzer entfallen. Die Werkbank führt durch die Arbeitsschritte zur Erschließung semi-strukturierter Daten. Die Funktionen umfassen den Import der Rohdaten, die Segmentierung in strukturelle Einheiten, die Extraktion von Textobjekten, die Erstellung und Bereinigung des Korpus und schließlich den Export der Ergebnisse. Die Funktionen sind detailliert in Blessing et al. (2015) beschrieben.

Import der rohen Textdaten

In die Werkbank können Daten aus unterschiedlichsten Formaten importiert werden, darunter DOCX, ODT, HTML, XML, PDF und *plain text*. In die Werkbank wurde das Apache Toolkit Tika[3] integriert, das die verschiedenen Formate

2 http://www.tei-c.org/Guidelines/P5

3 https://tika.apache.org

in *plain text* konvertiert. Dabei gehen zwar die Elemente der Textformatierung (Schriftgröße, Fettdruck usw.) verloren, aber durch das einheitliche Format stehen die Funktionen der Werkbank möglichst generisch für unterschiedliche Rohdaten zur Verfügung. Zusätzlich können so verschiedene Formate in ein einheitliches Korpus integriert werden. Die Nutzer können während des Imports wählen, ob Markup entfernt wird oder nicht. Für Auszeichnungssprachen (XML, HTML) können sie dadurch bestimmen, ob für die Definition der Indikatoren zur Extraktion von Textobjekten die Elemente der Tags verfügbar sind. Die Standardeinstellung sieht vor, das Markup zu entfernen.

Segmentierung in strukturelle Einheiten

Die Nutzer können die importierten Textdateien bei Bedarf zuerst in strukturelle Einheiten segmentieren. Das wird dort nötig, wo viele Texte in einer Datei enthalten sind, z. B. bei Dateien aus einem Zeitungsarchiv, die alle Artikel einer Zeitung aus der Produktion eines Tages enthalten, oder bei Einträgen eines Katalogs oder eines Wörterbuchs. Dafür können sich die Nutzer Ausschnitte der importierten Daten in einem Vorschaufenster anzeigen lassen, um nach Indikatoren für die Grenzen der Einheiten zu suchen. Unser Ansatz ist, dass in semistrukturierten Textsammlungen die Grenzen zwischen solchen Einheiten für den menschlichen Leser leicht erkennbar sind, aber in Anweisungen für die maschinelle Prozessierung übersetzt werden müssen. Die Nutzer können das Ende einer Einheit über *Segmentierungsregeln* beschreiben. Sie können dafür Ankerwörter oder reguläre Ausdrücke definieren, eine minimale oder maximale Zeilenlänge festlegen oder Ankerwörter in benachbarten Zeilen angeben.

Abbildung 1 zeigt das Vorschaufenster mit einem Ausschnitt importierter Textdaten. Es handelt sich hierbei um eine kleine Sammlung von Fabeln und Gedichten aus dem Projekt Gutenberg. Absätze erscheinen im Vorschaufenster ohne Zeilenumbruch in einer Zeile. Im Ausschnitt sind die Grenzen zwischen den Einheiten leicht an der Breadcrumb-Navigation aus dem Gutenberg-Archiv („Gutenberg > Heinrich von Kleist >“) zu erkennen. Dieser Indikator wurde in eine Regel übertragen, nach der eine Zeile das Ende einer Einheit darstellt, wenn die darauffolgende Zeile den Indikator „Gutenberg >“ enthält. Im Vorschaufenster wird markiert, welche Zeilen nach dieser Regel als das Ende einer strukturellen Einheit erkannt werden.

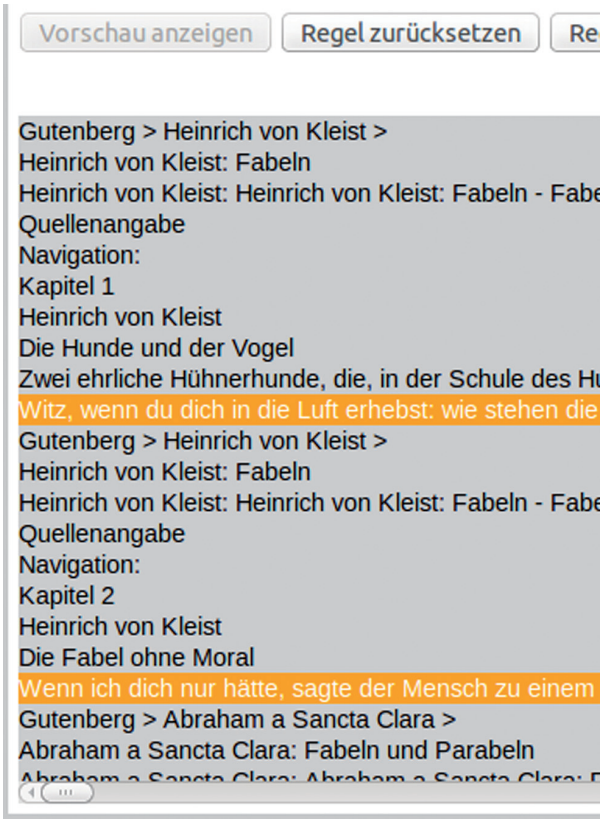


Abbildung 1: Vorschau zu einer Segmentierungsregel.

Extraktion von Textobjekten

Aus den strukturellen Einheiten werden verschiedene *Textobjekte* extrahiert: Fließtextabschnitte, Metadaten oder andere Textelemente des Datenschemas. Dazu definieren die Nutzer *Extraktionsregeln*. Für jede Regel vergeben sie einen Bezeichner, mit dem die erkannten Textobjekte ausgezeichnet werden. Als Indikatoren stehen wieder die Elemente der Segmentierungsregeln zur Verfügung; zusätzlich können Tokenfolgen definiert werden. Tokens werden über Kategorien wie klein- oder großgeschriebenes Wort, Wort in Versalien, Ziffernfolge, über die Wortart oder über einen Abgleich mit einer nutzerdefinierten Wort- oder Terminologieliste beschrieben. Als Textobjekte können (1) der gefundene Indikator, (2) der Satz, der den Indikator enthält, (3) die Zeile, die den Indikator enthält, und (4) mehrzeilige Bereiche definiert werden.

Neben der (a) Extraktion von Textobjekten können über die Regeln (b) Bereiche aus den erkannten Textobjekten entfernt werden (Bereinigung der Textobjekte); (c) die Extraktion durch andere Regeln kann verhindert werden (Blocker-Regeln); und (d) innerhalb von erkannten Text objekten können Textstellen ausgezeichnet werden (Anno-

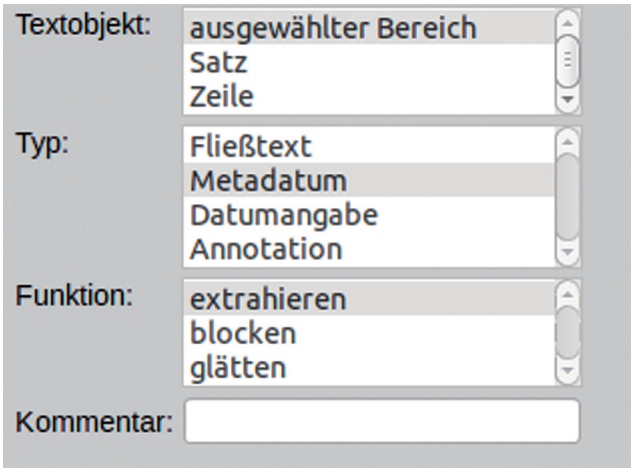


Abbildung 2: Angaben zu einer Extraktionsregel.

tationsregeln). Die erkannten Textobjekte bilden das Korpus. Anschließend kann das Korpus um (Semi-)Dubletten, um leere und zu kurze Textobjekte sowie um Textobjekte ohne Fließtext bereinigt werden. Abbildungen 2 und 3 zeigen Ausschnitte zur Definition einer Extraktionsregel. Um die Angabe zum Kapitel als Metadatum zu extrahieren (Abbildung 2), wurde die Tokenfolge [Token: Kapitel] [Zahlen] definiert (Abbildung 3). Die Tokenfolge extrahiert aus strukturellen Einheiten die Textstellen, in denen der Anker „Kapitel“ von einer Zahlenfolge gefolgt wird.

Export der Ergebnisse

Das Korpus kann im TSV-Format (*tab-separated values*) und nach Microsoft Excel exportiert werden. Zusätzlich bieten wir den Aufbau einer TEI-Datei (Format entsprechend den P5-Guidelines) aus den erkannten Textobjekten an. Für die Erstellung des TEI-Headers füllen die Nutzer verschiedene Eingabefelder aus, die neben einem Titel für das Korpus auch Angaben zu den Editoren, den importierten Daten (Quelle, Lizenz, Dokumenttyp) und möglichen Revisionen (z. B. Änderungen der Regeln) abfragen. Die Werkbank hält zusätzlich die verwendeten computerlinguistischen Prozesse und die Anzahl der Tokens des Korpus fest.

5 Konzepte für einen einfachen Zugang

Der aktuelle Prototyp der Explorationswerkbank wurde in enger Kooperation mit den Politikwissenschaftlern im

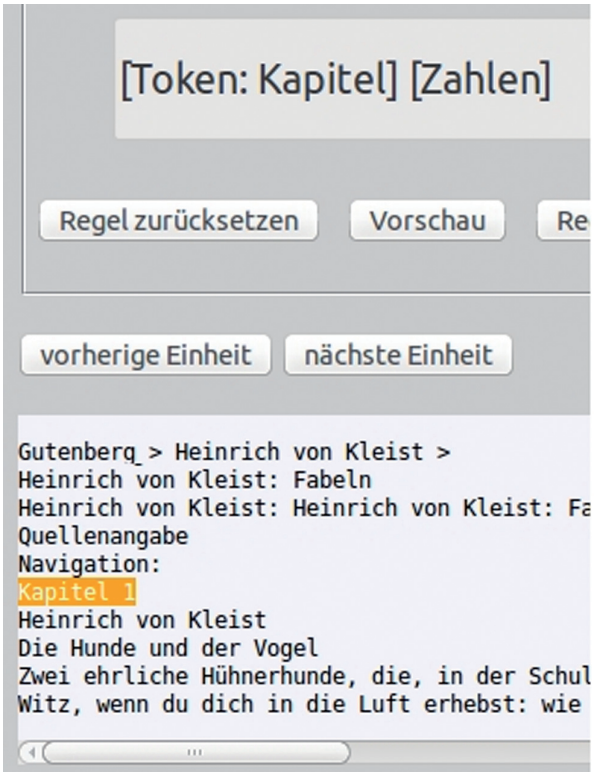


Abbildung 3: Definition einer Tokenfolge.

e-Identity-Projekt entwickelt und u. a. in einem Workshop mit weiteren Forschern aus den Digital Humanities ausführlich erprobt und diskutiert; eine Entwicklung auf der Basis detaillierter Anforderungsanalysen im Sinne des *UCD* oder eine Usability-Evaluierung waren im Projektrahmen aber nicht realisierbar. Die grafische Benutzerschnittstelle ist nicht Schwerpunkt des vorliegenden Artikels; jedoch standen die Benutzung eines intuitiv verständlichen Vokabulars für die Funktionen und Optionen der Schnittstelle, Selbstbeschreibungsfähigkeit und Erwartungskonformität sowie flexible Nutzbarkeit durch Nutzer mit und ohne Anwendungserfahrung im Vordergrund. Designentscheidungen auf der funktionalen Ebene sind im Folgenden kurz dargestellt.

Transparenz durch Prozessmetadaten

Ein über die Werkbank erstelltes Korpus ist nicht mit dem ursprünglichen Textmaterial gleichzusetzen. Seine Inhalte und sein Umfang werden von den Nutzern bestimmt. Die Werkbank ermöglicht die Wiederverwendung von Daten, indem spezifische Textformate in ein „eigenes“, strukturiertes Format überführt werden; dabei muss nicht das gesamte importierte Textmaterial in das entstehende Korpus einfließen. Die Werkbank kann auch dazu verwendet

werden, gezielt einzelne Textobjekte aus Textdaten zu extrahieren, bzw. Texte nach bestimmten Kriterien in das entstehende Korpus einzubinden oder von dort auszuschließen. Während des Imports wird die Textformatierung entfernt.

Um diese Schritte transparent und reproduzierbar zu halten, werden Prozessmetadaten festgehalten. Wenn Extraktionsregeln auf strukturelle Einheiten angewendet werden und ein Korpus entsteht, wird ein Zeitstempel erstellt und der Login-Name des angemeldeten Nutzers festgehalten. Für jedes Textobjekt werden eine Identifikationsnummer sowie eine Identifikationsnummer für die zugrundeliegende strukturelle Einheit und der Name der importierten Textdatei festgehalten. Für jedes Korpus wird ein Bezeichner vergeben. Zudem werden die von den Regeln verlangten und bei ihrer Anwendung eingesetzten computerlinguistischen Verarbeitungsschritte notiert.

Schließlich ändert auch die Filterung um (Semi-)Dubletten und defekte Artikel den Umfang des Korpus. Erkannte (Semi-)Dubletten und defekte Einträge werden nicht aus dem Korpus gelöscht, sondern markiert.

Reduktion der Wahlmöglichkeiten

Die Web-Anwendung führt die Nutzer durch die Arbeitsschritte (*Wizard*-Konzept, vgl. z. B. Tidwell, 2010, 54 ff.). Um den Einstieg in die Funktionen zu erleichtern, führt die Software dabei durch die elementaren Funktionen (z. B. Extraktionsregeln über feste Ankerwörter); komplexere Funktionen (z. B. reguläre Ausdrücke) werden nur auf Wunsch der Nutzer angewandt (ähnlich wie „advanced search“ in vielen Suchmaschinen). In den Vorschaufenstern können die Nutzer zunächst für einige strukturelle Einheiten prüfen, ob ein Verarbeitungsschritt die gewünschten Ergebnisse liefert, ehe sie die Verarbeitung des gesamten importierten Datenmaterials starten.

Die Ankerwörter und die elementaren Typen der Tokenfolgen sind als Unterstützung für die Erstellung regulärer Ausdrücke gedacht, auch wenn dabei der Funktionsumfang gegenüber der vollen Mächtigkeit regulärer Ausdrücke reduziert wird. Nutzer können aber auf Wunsch auch reguläre Ausdrücke eingeben. Für die Verwendung von Wortarten in Tokenfolgen werden die Daten mit dem TreeTagger (Schmid, 1994) verarbeitet. Das Tagging ist möglich für Deutsch, Englisch und Französisch. Die Wortarten, die für die Tokenfolgen verwendet werden können, wurden auf Nomen, Verb, Adjektiv, Adverb, Artikel, Präposition, Pronomen, Konjunktion und Satzzeichen reduziert (vgl. das *Universal Tagset*).

Implizite Steuerung der computerlinguistischen Techniken

Die Extraktionsregeln verlangen unterschiedliche Verarbeitungsschritte: Tokenisierung, Erkennung von Wortarten, den Abgleich mit den nutzerdefinierten Terminologielisten und die Bestimmung der Tokenfolgen. Während der Anwendung der Extraktionsregeln ermittelt die Explorationswerkbank die notwendigen Verarbeitungsschritte und steuert den Aufruf der eingesetzten computerlinguistischen Werkzeuge. Die Nutzer müssen dadurch nicht selbst entscheiden, wann ein computerlinguistisches Werkzeug einzusetzen ist, und können die Prozesskette vom intendierten Ergebnis her denken. Die computerlinguistischen Werkzeuge sollen also als Hilfsmittel für die Funktionen der Werkbank dienen und sollen nur angewendet werden, wenn Bedarf besteht. In der Datenbank werden zu jedem Textobjekt auch die computerlinguistisch annotierten Daten abgelegt. Interessierte Nutzer können auf Wunsch die Annotationen einsehen; im Default-Fall bleiben sie unsichtbar. Durch die oben beschriebene Kombination von Vereinfachung und optionalem Zugriff auf komplexere Operationen soll sichergestellt werden, dass verschiedene Nutzer mit unterschiedlichem Bedarf an die Kontrolle der Zwischenergebnisse mit der Werkbank arbeiten können.

6 Fallbeispiele

Die Funktionen der Explorationswerkbank wurden zunächst für die Erstellung des *e-Identity*-Korpus entwickelt. Dessen Daten stammen aus zwölf Zeitungen aus sechs Ländern und beinhalten deutsche, französische und englische Artikel. Die Artikel wurden aus fünf Volltextdatenbanken heruntergeladen, die jeweils eigene Datenschemata verwenden. Wir beschreiben in diesem Abschnitt weitere Anwendungen auf Texte des Gegenwartsdeutschen, als Beispiele für verschiedene Arten semi-strukturierter Textdaten. Diese Beispiele sind Transkripte von Radio-Interviews, Beschreibungen für Kopiergeräte und Handbücher für Heimwerker.

6.1 Transkripte von Radio-Interviews

Transkripte von Radio-Interviews sollten aufbereitet und in ein maschinenlesbares Format überführt werden. Die Transkripte konnten als PDF-Dateien auf der Website des Radiosenders heruntergeladen werden und waren einheitlich strukturiert. Die PDFs enthielten vor der Konvertierung in

plain text Formatangaben und als grafisches Element das Logo des Radiosenders.[4] Jedes Dokument hatte einen gleichförmig aufgebauten „Kopf“. Er enthielt für alle Interviews einheitlich den Titel der Sendung und den Namen des Radiosenders; anschließend wurden Metadaten zum Autor der Sendung, zur interviewten Person, zur Redaktion und zur Sendezeit (Datum und Uhrzeit) angegeben.

Der Aufbau dieser Daten war für menschliche Leser unmittelbar erkennbar. Die Metadaten waren durch Ankerwörter (z. B. „Autor:“, „Redaktion:“) markiert. Anschließend folgten als die textlichen Inhalte die Fragen und die Antworten der interviewten Person, ebenfalls eingeleitet durch feste Ankerwörter (der Name des Radiosenders für die Fragen und die Initialen der interviewten Person für die Antworten). Jedes Interview umfasste mehrere PDF-Seiten, die ab der zweiten Seite eine Kopfzeile mit dem Namen der Sendung und der Seitenangabe enthielten.

Die Daten wurden in die Werkbank importiert und in *plain text* konvertiert. Über Segmentierungsregeln wurden der Kopf jedes Interviews sowie jeweils einzeln die Fragen und Antworten als strukturelle Einheiten beschrieben. Über Extraktionsregeln wurden in den strukturellen Einheiten Textobjekte (Metadaten des Kopfs, Fragen, Antworten) definiert und jeweils mit einem Bezeichner versehen. Eine Regel zur Textbereinigung entfernte die störenden Kopfzeilen in den Fließtexten. Das entstandene Korpus wurde im TSV-Format exportiert. Die Exportdatei enthielt für jedes Textobjekt die Identifikationsnummer, die Identifikationsnummer der zugrundeliegenden strukturellen Einheit und den Dateinamen der importierten Datei.

6.2 Bedienungsanleitungen für Kopiergeräte

Ein Übersetzungsbüro sollte für einen Elektronik-Hersteller Web-Inhalte mit Bedienungsanleitungen für Kopiergeräte übersetzen. Die rohen Textdaten lagen in HTML und XML vor. Aus den Rohdaten sollten gezielt bestimmte Textobjekte extrahiert werden. In einer begleitenden PDF-Datei wurden (1) Typen zusammengehörender und einheitlich strukturierter HTML- und XML-Seiten und (2) die Elemente, die übersetzt werden sollten, beschrieben. Abbildung 4 zeigt einen (anonymisierten) Ausschnitt aus dem PDF: Auf „Titel“-Seiten sollten z. B. die Inhalte der „TitleLeaf“-Elemente übersetzt werden. Die HTML- und XML-Dateien wurden in die Werkbank importiert. Das Markup wurde beibehalten. Für dieses Beispiel wurden keine Segmentierungsregeln geschrieben, damit als strukturelle Einheit jeweils eine gesamte importierte Datei gewertet wurde. Als Extraktionsregeln wurden die im Schema beschriebenen Inhalte mit regulären Ausdrücken beschrieben (Beispiel: „/<TitleLeaf.*<\/TitleLeaf>/“) und als Textobjekte extrahiert. Über Regeln zur Textbereinigung können die HTML- und XML-Elemente entfernt werden.

Translate the text of the "TitleLeaf" element.

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE TitleLeafs PUBLIC>
<TitleLeafs name=
"fc090778-09aa-4175-9beb-e3f84a967b32">
<TitleLeaf Language="en">
<Heading id="_f10b95b8-a78b-4657-8203">
Adding Paper</Heading>
</TitleLeaf>
</TitleLeafs>
```

Abbildung 4: Explizite Beschreibung von XML-Dateien.

6.3 Handbücher für Heimwerker

Eine weitere Sammlung enthielt 19 Handbücher für Heimwerker. Ihre Inhalte sollten in ein größeres Korpus mit Texten aus dem Heimwerker-Bereich fließen. Die Handbücher waren einheitlich strukturiert. Die Abschnitte waren an einheitlichen Überschriften erkennbar: „Allgemeine Sicherheitshinweise“; „Abgebildete Komponenten“; „Funktionsbeschreibung“ usw. Die Überschriften konnten gut als Indikatoren für die Extraktion mehrzeiliger Textbereiche verwendet werden. Die Abschnitte bestanden teilweise aus identischen oder sehr ähnlichen Textbausteinen. Sie konnten in der Werkbank als (Semi-)Dubletten gefiltert werden.

7 Zusammenfassung

Wir haben in diesem Artikel dargestellt, wie über eine Web-Anwendung Forschern aus den Geisteswissenschaften ein einfacher Zugang zur Arbeit mit digitalen Textsammlungen ermöglicht werden soll. Die Web-Anwendung wurde für semi-strukturierte Daten konzipiert: Für solche Daten können menschliche Leser leicht die zugrundeliegende Datenstruktur erkennen. Dennoch müssen die Daten erst in ein maschinenlesbares Format überführt werden, in dem auf die Elemente der Datenstruktur zugegriffen werden kann. Um den *einfachen* Zugang zu erreichen, werden in der Web-

4 Die Daten werden anonymisiert dargestellt, um Urheberrechtsverletzungen zu vermeiden.

Anwendung Reduktionen in Kauf genommen. Beispielsweise wird nicht das gesamte Textmaterial in der strukturierten Form abgebildet, sondern nutzerdefinierte *Textobjekte*. Die computerlinguistischen Verarbeitungsschritte werden implizit gesteuert und die Nutzer haben im Default-Fall keinen Einblick in die computerlinguistischen Annotationen. Abschließend haben wir Beispiele diskutiert, wie unterschiedliche Textsorten für verschiedene Anwendungsfälle erschlossen werden können.

Literatur

Blei, D. M.; Ng, A. Y.; Jordan, M. I. (2003). Latent dirichlet allocation. In: *Journal of Machine Learning Research*. Bd. 3, 2003, S. 993–1022.

Blessing, A.; Sonntag, J.; Kliche, F.; Heid, U.; Kuhn, J.; Stede, M. (2013). Towards a tool for interactive concept building for large scale analysis in the humanities. In: *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2013)*. Sofia, Bulgarien: ACL, S. 55–64.

Blessing, A.; Kliche, F.; Heid, U.; Kantner, C.; Kuhn, J. (2015). Computerlinguistische Werkzeuge zur Erschließung und Exploration großer Textsammlungen aus der Perspektive fachspezifischer Theorien. In: Baum, C.; Stäcker, T. (Hrsg.). *Zeitschrift für Digital Humanities. Sonderband 1: Grenzen und Möglichkeiten der Digital Humanities*.

Brooke, J.; Hammond, A.; Hirst, G. (2015). GutenTag. An NLP-driven Tool for Digital Humanities Research in the Project Gutenberg Corpus. In: *Proceedings of the 4th Workshop on Computational Linguistics for Literature (CLfL 2015)*. Denver, USA: ACL, S. 42–47.

Buneman, P. (1997). Semistructured data. In: *Proceedings of the 16th ACM SIGACT SIGMOD-SIGART symposium on Principles of database systems (PODS 1997)*. New York, USA: ACM, S. 117–121.

Cunningham, H.; Maynard, D.; Bontcheva, K.; Tablan, V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*. Philadelphia, USA: ACL, S. 168–175.

Dick, M. (2015). Erstellung themenspezifischer Korpora mit dem LDA-basierten Klassifikationsmodell. In: Elbeshausen et al. (Hrsg.). *Proceedings des 9. Hildesheimer Evaluierungs- und Retrievalworkshops (HiER 2015)*. Hildesheim: Universitätsverlag Hildesheim, S. 19–26.

Fiumara, G. (2007). Automated information extraction from web sources: a survey. In: *Proceedings of the Between Ontologies and Folksonomies Workshop (BOF07)*. East Lansing, USA, S. 1–9.

Gleim, R.; Mehler, A. (2015). TTLab Preprocessor – Eine generische Web-Anwendung für die Vorverarbeitung von Texten und deren Evaluation. In: *Book of Abstracts der 2. Jahrestagung der Digital Humanities im deutschsprachigen Raum (DHd 2015)*. Graz, Österreich.

Hammond, A.; Brooke, J.; Hirst, G. (2013). A tale of two cultures: Bringing literary analysis and computational linguistics together. In: *Proceedings of the 2nd Workshop on Computational Literature for Literature (CLFL 2013)*. Atlanta, USA: ACL.

Kliche, F.; Blessing, A.; Heid, U.; Sonntag, J. (2014). The eIdentity Text Exploration Workbench. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Island: ELRA.

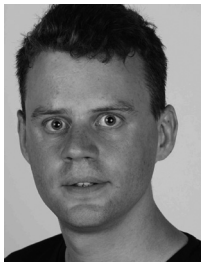
Mehler, A.; Schwandt, S.; Gleim, R.; Jussen, B. (2011). Der eHumanities Desktop als Werkzeug in der historischen Semantik: Funktionsspektrum und Einsatzszenarien. In: *Journal for Language Technology and Computational Linguistics (JLCL)*. Bd. 26, Nr. 1, S. 97–117.

Moretti, F. (2013). *Distant Reading*. London: Verso.

Overbeck, M. (2014). European debates during the Libya crisis of 2011: shared identity, divergent action. In: *European Security*. Bd. 23, Nr. 4, S. 583–600.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In: *Proceedings of International Conference on New Methods in Language Processing*. Manchester, GB.

Tidwell, J. (2010). *Designing Interfaces: Patterns for Effective Interaction Design* (2. Ausgabe). O'Reilly Media.



Fritz Kliche
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5b
70569 Stuttgart
klichefz@ims.uni-stuttgart.de

Fritz Kliche promoviert an der Universität Hildesheim und hat derzeit ein Stipendium des Graduiertenkollegs des SFB 732 am Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart. Er interessiert sich besonders für die Digital Humanities: Wie finden computerlinguistische Techniken und ihre geisteswissenschaftlichen Anwender zusammen?

Prof. Dr. Ulrich Heid
Institut für Informationswissenschaft und Sprachtechnologie
Universität Hildesheim
Universitätsplatz 1
31141 Hildesheim
heid@uni-hildesheim.de

Ulrich Heid ist Professor für Sprachtechnologie/Computerlinguistik an der Universität Hildesheim. Er arbeitet zu computerlinguistischen Werkzeugen und Ressourcen und zu ihren Anwendungen.