Typisierte Varianz-Analyse von Texten

Balbach, Nico
nico.balbach@gmail.com
Zentrum für Philologie und Digitalität „Kallimachos“, Universität Würzburg, Deutschland

Reul, Christian
christian.reul@uni-wuerzburg.de
Zentrum für Philologie und Digitalität „Kallimachos“, Universität Würzburg, Deutschland

Puppe, Frank
frank.puppe@uni-wuerzburg.de
Zentrum für Philologie und Digitalität „Kallimachos“, Universität Würzburg, Deutschland; Lehrstuhl für Künstliche Intelligenz und Angewandte Informatik, Universität Würzburg, Deutschland

Einleitung

Häufig gibt es unterschiedliche Quellen oder Auflagen zu einem Werk, deren Analyse Rückschlüsse auf die Entstehungsgeschichte oder auf unterschiedliche Akzentuierungen, aber z. B. auch auf Transkriptionsfehler zulässt. Dazu gehören nicht nur Differenzen in den Texten, sondern auch in der Typographie (z. B. kursive Hervorhebungen oder Fontgröße). Wir präsentieren das Open-Source Web-Tool „Variance-Viewer"[1] das, anders als die übliche Diff-Funktion in Texteditoren, nicht nur zwei Texte vergleichen und die Unterschiede markieren und hervorheben, sondern auch die Varianzen mit Regeln in Typen einteilen kann. Die verschiedenen Typen können ein- oder ausgeblendet sowie mit unterschiedlichen Farben markiert werden. Weiterhin können die zu vergleichenden Texte vor dem Vergleich normalisiert werden. Dadurch wird die Übersichtlichkeit bei vielen kleineren Unterschieden erheblich gesteigert, und es kann auf fachlich relevante Differenzen fokussiert werden. Es ist ein TEI-Export verfügbar, in dem für die Varianzen vordefinierte Tags generiert werden. Folgender Workflow soll beim Vergleich zweier Werke unterstützt werden:

1. Übersicht über Differenzen bekommen (dafür eignet sich praktisch jedes Diff-Tool).
2. Wiederhole:

a. Definition von Typen der Differenzen mittels Konfigurationsdatei.

b. Ein- und Ausblenden der Typen und Untersuchung der Restkategorie, ob weitere Typ-Definitionen sinnvoll sind.

3. Weitere editorische Arbeiten, ggf. TEI-Export der typisierten Differenzen.

Verwandte Arbeiten

Die meisten Texteditoren verfügen über eine Diff-Funktion, mit der sich zwei Texte (auch Programmcode oder DNA-Sequenzen) vergleichen und insbesondere Änderungshistorien von Dokumenten nachverfolgen lassen (vgl. z. B. die Darstellungen von Varianten in der Faust-Edition[2]). Dabei gibt es häufig zwei Darstellungen: zum einen die der Änderungen innerhalb eines Dokumentes und zum anderen die Gegenüberstellung der beiden Dokumente mit jeweiliger Hervorhebung der Änderungen. Viele Algorithmen basieren auf der Publikation von Myers (Myers 1986), der gezeigt hat, dass die Suche nach der längsten gemeinsamen Teilfolge und der kürzesten Transformation eines Strings A in einen String B als äquivalent angesehen werden können. Eine Implementierung ist die Suche nach einem kürzesten Weg in einem Edit-Graphen bzw. einer Matrix, der aus den Wörtern oder Buchstaben der beiden Dokumente als Zeilen bzw. Spalten besteht. Für literarische Texte ist im Allgemeinen eine feinere Differenzierung wünschenswert, in der Typen von Änderungen erkannt und ein- oder ausgeblendet werden können. Diese können sich sowohl auf den Text als auch die Typographie beziehen. Da die Typen von den individuellen Interessen der jeweiligen Philologen abhängen, sollten sie nicht fest vorgegeben, sondern leicht anpassbar sein. Weiterhin ist neben einer Visualisierung auch ein Export nach TEI wünschenswert. Im Folgenden präsentieren wir ein solches Tool, da wir kein vergleichbares, einfach bedienbares Werkzeug kennen (so wird z. B. in der Übersicht über Digital-Humanities-Tools und Services in (Bulatovic 2016) diese Kategorie nicht erwähnt). Ein ähnliches, aber anspruchsvolleres Tool ist CollateX[3] (Haentjens Dekker 2014), das in der Lage ist, zwei und mehr Texte zu kollationieren und das Ergebnis als Graph zu visualisieren. Dabei können auch Transpositionen, d. h. verschobene Texte gefunden werden, teilweise einschließlich Erkennung von Varianten der verschobenen Texte. Ein weiteres anspruchsvolles Tool ist Stemmaweb[4], dessen GitHub-Repository[5] jedoch darauf hindeutet, dass es nicht oder kaum noch aktiv gepflegt wird. Im Beitrag (Andrews 2014), bei dem es um eine kritische Bewertung der Entstehungsgeschichte von drei Werken geht, wird u. a. kritisiert, dass bestimmte Typen von Änderungen vorschnell als „insignifikant“ bewertet werden. Entwurfsregeln zur Visualisierung von Text-Varianz-Graphen werden in (Jänicke 2014) dargestellt. Der Variance-Viewer hat einen anderen Schwerpunkt: er gibt die Typen von Änderungen nicht vor, sondern überlässt deren Definition dem Anwender durch einfache Konfiguration. Die Darstellung enthält keine Graphen, sondern eine farbige Kennzeichnung und bietet das Aus- und Einblenden von Typen von Varianten durch einfachen Klick an, so dass Editoren die Übersicht behalten, wenn Sie sich auf bestimmte Differenztypen konzentrieren wollen.

Methoden

Der Variance-Viewer verwendet für die Berechnung von Differenzen zwischen zwei Texten eine Implementierung[6] des Algorithmus von Myers und fügt dann Nachbearbeitungen zur Differenzierung verschiedener Typen von Änderungen hinzu. Die Kategorien sind frei konfigurierbar (s. Abbildung 1 links unten für einen Auszug aus der Konfigurationsdatei). Die Nachbearbeitung prüft für jede gefundene Änderung, ob die Bedingungen für einen der definierten Typen vorliegen und ordnet sie dann dem entsprechenden Typ zu. Die Änderungen werden auf Wortebene berechnet und zusätzlich die für die Änderung verantwortlichen Buchstaben identifiziert, so dass beides hervorgehoben werden kann, wobei zusätzliche Leerzeichen auch wortübergreifend gefunden werden. Alle nicht zugeordneten Typen werden einem Default-Typ (z. B. „Inhalt“ bzw. „Content“) zugeordnet, wobei noch zwischen einfachen und komplexen Änderungen unterschieden werden kann (einfache Änderungen unterscheiden sich nur in einem Buchstaben). Die Ergebnisse können in TEI ausgegeben werden, indem das „app“-Tag mit speziellen Attributen für die Änderungstypen benutzt wird. Weiterhin können sie visuell präsentiert werden, wobei den Typen verschiedene Farben zugeordnet werden und bei Bedarf jeder Typ auch ausgeblendet werden kann, um die Übersicht zu verbessern. Das Programm präsentiert beide Texte in einer synoptischen Darstellung, wobei zur Gewährleistung einer zeilenäquivalenten Darstellung in einem Dokument freier Platz auf Abschnittsebene hinzufügt wird, falls das notwendig ist.

Den Umgang mit den Typen erläutern wir an zwei philologischen Anwendungsprojekten, in denen der Variance-Viewer eingesetzt wurde: Die Analyse der Änderungen in den Schriften von Richard Wagner im Projekt RWS[7] und die Analyse der verschiedenen Auflagen von Drucken im Narragonien digital Projekt[8].

Im RWS-Projekt liegen die Texte als TEI-Dokumente vor. Bei der Analyse der Varianzen sind nicht nur textuelle Änderungen interessant, sondern auch Änderungen bzgl. der Formatierung, die in TEI im Element „rend“ hinterlegt sind. Daher wird dieses genauer analysiert. Insgesamt sind folgende Typen von Änderungen durch projektspezifische Regeln definiert (vgl. Abbildung 1):

- Satzzeichen (Punctuation): Die Änderung bezieht sich nur auf ein Satzzeichen (. , ; - ? ! usw.).
- Grapheme (Graphemics): Die Änderung bezieht sich nur auf bestimmte Schreibweisen (y i; u v; s #; ss ß; Groß/Kleinschreibung; th t; usw.).
- Abkürzungen (Abbreviation): Die Änderung bezieht sich nur auf Abkürzungen (z. B. Dr. Doktor; Hr. Herr Herrn; usw.).
- Typographie (Typography): Die Änderung ist keine inhaltliche, sondern bezieht sich auf das Layout oder die Typographie und wird in dem TEI-Attribut „rend“ mit entsprechenden Werten spezifiziert (kursiv; gesperrt; usw.).
- Inhalt (Content): Alle übrigen Änderungen, die keiner der obigen Kategorien zugeordnet werden können einschließlich Hinzufügen oder Löschen sowie Änderungen, bei denen mehr als eine Änderung der obigen Typen gleichzeitig vorkommt.

Im Narragonien-Projekt liegen die Drucktexte als Plain Text Dateien vor. Hier werden folgende Typen von Änderungen unterschieden (vgl. Abbildung 2):

- Grapheme (mit anderer Liste von Buchstabenersetzungen wie im RWS-Projekt).
- Abkürzungen (mit anderer Bedeutung als im RWS-Projekt; hier sind es meist einzelne Buchstaben mit Unter- oder Überstrichen, die expandiert werden).
- Leerzeichen im Wort, die ein Wort in zwei oder mehrere Wörter auftrennen. (Separation). Diese Option ist technisch aufwändiger, weil nicht einzelne Wörter sondern Wortgruppen miteinander verglichen werden müssen.
- Inhaltsänderungen mit nur einem Zeichen Unterschied (OneDifference), die nicht in der Graphem-Liste enthalten sind und anders bewertet werden als komplexere Änderungen.
- Inhalt (Content): Alle übrigen Änderungen.

Erfahrungen

Das Tool wurde in beiden Projekten erfolgreich eingesetzt, und dabei auch für die Verarbeitung sehr langer Dokumente genutzt. Im Folgenden zeigen wir zwei Screenshots aus dem RWS- und dem Narragonien-Projekt. Dabei ist besonders hervorzuheben, dass der Rest-Typ „Content“, der alle sonst nicht speziell erkannten Typen von Änderungen beinhaltet, nur noch ca. die Hälfte der Änderungen ausmacht, während die andere Hälfte spezielleren Typen zugeordnet werden konnte. Wenn das Ziel die Feinanalyse bestimmter Änderungstypen ist, können auch iterativ weitere Typen definiert und der Analysealgorithmus damit erneut ausgeführt werden.

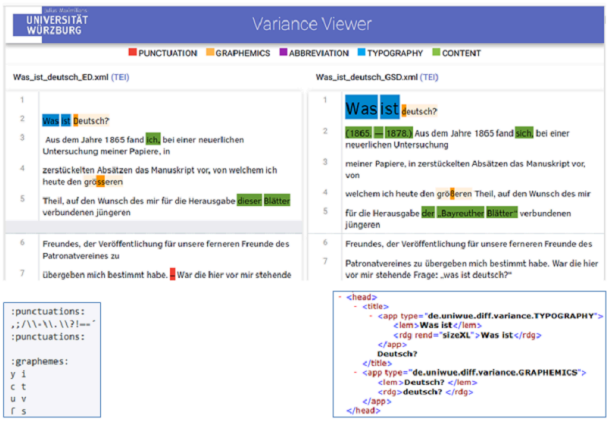

Abbildung 1: Vergleich zweier Texte aus dem Schriften-Verzeichnis von Richard Wagner mit Hervorhebung der Änderungstypen in verschiedenen Farben (Erläuterungen der Typen im Text; Auszug aus Konfigurationsdatei links unten). Die Texte liegen im Format TEI vor, wobei TEI-Attributwerte auf CSS abgebildet wurden, um die Darstellung unterschiedlicher Typen sichtbar zu machen, und die gefundenen Differenzen mit ihren Typen können auch als TEI exportiert werden (Auszug für die erste Zeile s. rechts unten).

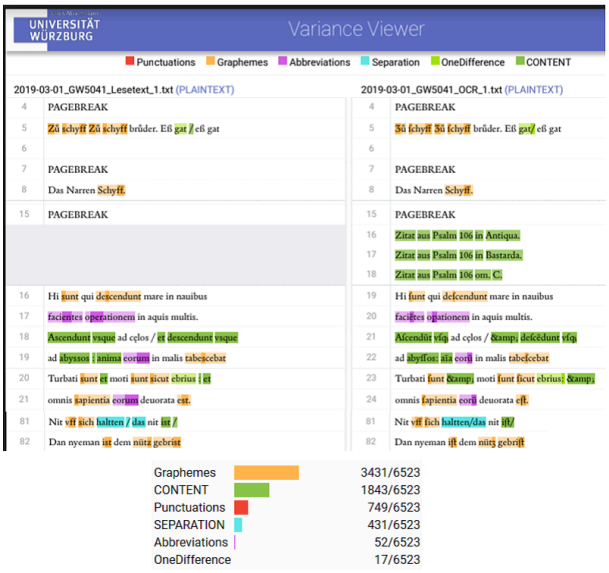

Abbildung 2: Vergleich des edierten Lesetextes der Narrenschiff-Ausgabe GW5041 (links) mit dem Ergebnis der OCR auf dem Originaltext einer anderen Druckausgabe (rechts), wobei die Änderungen sowohl OCR-Fehler als auch Normalisierungen der Schrift im Lesetext umfassen. Dies ist durch Hervorhebung der Änderungstypen in verschiedenen Farben leichter nachvollziehbar (Erläuterung der Typen im Text). Unten eine Statistik, die die 6.523 gefundenen Änderungen nach Änderungstypen aufschlüsselt. Der Gesamttext umfasste 150 Seiten mit 4.200 Zeilen, 26.000 Wörtern und 121.000 Zeichen und die zugehörige Konfigurationsdatei („Settings“) ca. 100 Zeilen. Für diese Analyse brauchte der Variance-Viewer in dem serverseitig ausgeführten Demo-Modus im Web ca. 25 Sekunden (für intensive Nutzung sollte der Open-Source Code lokal installiert werden).

Die bisherigen Erfahrungen zeigen, dass noch eine Reihe von relativ einfachen Erweiterungen wünschenswert sind, wobei zu erwarten ist, dass in weiteren Projekten weitere Aspekte hinzukommen:

- Gelegentlich enthält ein Wort mehrere Änderungen (z. B. mehrere Grapheme und/oder Satzzeichen). Wenn es mehrere Änderungen desselben Typs gibt, werden diese dem Typ zugeordnet (z. B. „Schiff“ in Zeile 5 in Abbildung 2 mit zwei graphemischen Differenzen). Wenn es jedoch Änderungen unterschiedlicher Klassen sind, werden diese als ein nicht näher differenzierter Unterschied („Content“) betrachtet (z. B. „ist /“ und „iſt/“ in Zeile 81 mit zwei verschiedenen Typen von Änderungen bezüglich Leerzeichen und Graphem). Hier wäre eine Mischklasse aus den jeweiligen Ursprungsklassen wünschenswert. Das Problem lässt sich teilweise durch vorherige Normalisierung lösen, indem z. B. alle Graphem-Änderungen vorab normalisiert werden und sich dann manche komplexe Fehler zu einfachen Fehlertypen reduzieren.
- Es gibt Ausnahmen zu den Regeln, in denen die Nutzer die Änderungsklasse manuell ändern und ggf. kommentieren können sollten. Bisher zeigt der Viewer nur das automatisch generierte Ergebnis der Regelauswertung an, er sollte um eine Editierfunktion erweitert werden. Dies ist z. B. hilfreich, wenn bei automatischer OCR von verschiedenen Ausgaben eines Werkes zwischen OCR-Fehlern und Textvarianten unterschieden werden soll.

Zusammenfassung und Ausblick

Der vorgestellte Variance-Viewer ermöglicht die Feindifferenzierung und Klassifikation von Textvarianten mittels selbstdefinierter Typen. In verschiedenen Ausgaben von literarischen Texten treten oft zahlreiche „technische“ Varianzen auf, die sich auf Satzzeichen, Leerzeichen, Buchstabenvarianten und ggf. auch auf das Layout oder die Typographie beziehen, die von eigentlichen inhaltlichen Änderungen zu trennen sind. Hier werden bei Verwendung eines einfachen Diff-Werkzeugs häufig so viele Änderungen angezeigt, dass der Überblick verloren geht. Ein Filtern bzw. Hervorheben bestimmter Typen von Varianzen erleichtert die philologische Arbeit beträchtlich. Wichtig ist, dass die Typen von Varianzen abhängig von den Fragestellungen und individuellen Interessen des jeweiligen Philologen leicht konfiguriert werden können. Der vorgestellte Varianz-Viewer erfüllt diese Anforderungen und hat sich in zwei größeren philologischen Projekten bewährt. Er ist Open-Source, webbasiert, leicht zu installieren und zu bedienen. Perspektiven der Weiterentwicklung umfassen eine einfachere oder sogar automatische Definition der Varianztypen sowie funktionelle Erweiterungen:

- Aus technischer Sicht sollte für die Definition von Varianztypen ein Editor bereitgestellt werden, so dass deren Definition im Vergleich zur bisherigen Konfigurationsdatei noch weiter vereinfacht wird. Dazu kann eine Regelsprache bereitgestellt werden oder ein Lernverfahren, dem einige Beispiele präsentiert werden und der das Muster dann selbständig erkennt.

- Die häufigsten Typen von Varianzen können auch durch Lernverfahren vollautomatisch erkannt werden (ohne vorgegebene Varianztypen), indem alle vom Diff-Algorithmus gefundenen Varianten auf gemeinsame Muster hin analysiert werden.
- Eine umfassende Änderung wäre die Weiterentwicklung des relativ einfachen Tools zur Erkennung komplexerer Änderungen wie Transpositionen und zur Visualisierung der Änderungen, auch von mehreren Werken, in Graphen, ggf. durch Übernahme entsprechender Funktionalitäten z. B. aus CollateX oder Stemmaweb.

Fußnoten

1. Web-Link vom Variance-Viewer: http://variance-viewer.informatik.uni-wuerzburg.de ; Code Open-Source unter: https://github.com/cs6-uniwue/Variance-Viewer
2. http://faustedition.net
3. https://collatex.net
4. https://stemmaweb.net
5. https://github.com/tla/stemmatology
6. Java-diff-utils: https://code.google.com/archive/p/java-diff-utils . Die Software wird von Google gehostet und wurde von 2009-2013 von verschiedenen Autoren entwickelt (s. https://code.google.com/archive/p/java-diff-utils/source/default/commits).
7. Richard Wagner Schriften (RWS): Historisch-Kritische Gesamtausgabe: http://www.musikwissenschaft.uni-wuerzburg.de/forschung/richard-wagner-schriften .
8. „Narragonien digital“: http://kallimachos.de/kallimachos/index.php/Narragonien .

Bibliographie

Andrews, Tara L (2014): Analysis of variation significance inartificial traditions using Stemmaweb, in *Digital Scholarship in the Humanities, 31(3)*.

Bulatovic, Natasa / Gnadt, Timo / Romanello, Matteo / Schmitt, Viola / Stiller, Juliane / Thoden, Klaus (2016): Usability von DHTools und Services, in *DARIAH Working Papers*: https://wiki.de.dariah.eu/download/attachments/14651583/AP1.2.3_Usability_von_DH-Tools_und-Services_final.pdf .

Haentjens Dekker, Ronald / van Hulle, Dirk / Midell, Gregor / Neyt, Vincent / van Zundert, Joris (2014): Computer-supported collation of modern manuscripts: CollateX and the Beckett Digital Manuscript Project, in *Digital Scholarship in the Humanities, 30(3), 452-470*.

Jänicke, Stefan / Geßner, Annette / Büchler, Marco / Scheuermann, Gerik (2014): 5 Design Rules for Visualizing Text Variant Graphs, in *DH*: https://www.informatik.uni-leipzig.de/~stjaenicke/5_Design_Rules_for_Visualizing_Text_Variant_Graphs.pdf .

Myers, Gene (1986): An O(ND) difference algorithm and its variations, in *Algorithmica, 1, 251-266*.