Applied Text as Graph (ATAG)

Kuczera, Andreas
andreas.kuczera@mni.thm.de
Technische Hochschule Mittelhessen, Deutschland
ORCID: 0000-0003-1020-507X

Stand der Forschung

Zahlreiche Konferenzen, Workshops, Artikel und Blogbeiträge in den letzten Jahren beschäftigten sich mit der Frage, was Text eigentlich ist. Dabei hat sich der Eindruck verfestigt, dass unser Verständnis von Text viele Interpretationen zulässt und daher ständig im Wandel ist. Dies spiegelt sich auch in den TEI-Richtlinien und den verschiedenen Wegen wider, wie sie angewendet werden. Dabei können Textedierende Module und Elemente sorgfältig anpassen, um zu einem Set von Kodierungsrichtlinien zu gelangen, das mit ihrer Interpretation und ihrem Forschungsinteresse am Quelltext übereinstimmt. Dennoch unterscheidet sich oft die Art und Weise, wie die Textdaten auf einem Computer gespeichert werden, vom intellektuellen Verständnis des Textes des Editors. Das bedeutet, dass textliche Merkmale, die nicht natürlich in das Hierarchiemodell von XML passen, nur mit Hilfe von Umwegen oder zusätzlicher (vokabularspezifischer) Codierung adäquat digital dargestellt werden können. Je mehr zusätzliche Codierung erforderlich ist, desto komplizierter wird es, den Text zu kodieren, zu verarbeiten oder abzufragen. Auch in der Computerlinguistik und in den Computational Literary Studies wird das Problem der konkreten Verbindung von Theorie und Datenmodellierung diskutiert (Pichler/Reiter 2022; Bode 2023). Es ist davon auszugehen, dass die digitale Editionswissenschaft an der Schnittstelle von Wissenschaft und Technik mit demselben Problem en kämpft, weil Forschende immer nur lose und implizit wissen, von welchen Zeichen sie sprechen, wenn sie Interpretationen und Schlussfolgerungen äußern. Dies führt zu einer elementaren informationellen Ungenauigkeit, die sich auf alle Interpretationsebenen ausbreitet und die Arbeit hemmen. Nicht zuletzt deswegen werden in den digitalen Editionswissenschaften seit einiger Zeit die Vorteile von Standoff-Formaten und Text as Graph diskutiert. Zuletzt haben Bleeker et al. (2022) ihren Vorschlag zu einer Text as Graph Markup Language (TagML) vorgestellt und dem bisher führenden Standard TEI-XML gegenübergestellt. Standoff-Formate werden in der Computerlinguistik sehr häufig verwendet und sind ein sehr robustes Format. Allerdings kann bei ihnen in der Regel bereits a n notierte r Text später nicht mehr geändert werden. Bei TagML ist dies möglich, allerdings wird TagML bisher kaum produktiv eingesetzt. Das ursprünglich von Desmond Schmidt (Schmidt 2016) entwickelte und in (Neill/Kuczera 2019) vorgestellte Standoff-Property-System (SPO) vereint einen s tandoff-basierten Ansatz mit einer Labeled-Property-Graphdatenbank. Darauf aufbauend wird hier das Konzept Applied Text as Graph (ATAG) vorgestellt, das bereits im Rahmen des DFG-geförderten Projekts zum Liber Epistolarum der Hildegard von Bingen (https://liberepistolarum.mni.thm.de/home) eingesetzt wird (Kuczera 2020) . Die Software der Publikationsumgebung (vgl. Abb. 1) steht auf GitHub zur Nachnutzung zur Verfügung (https://github.com/digicademy/graph-dse).

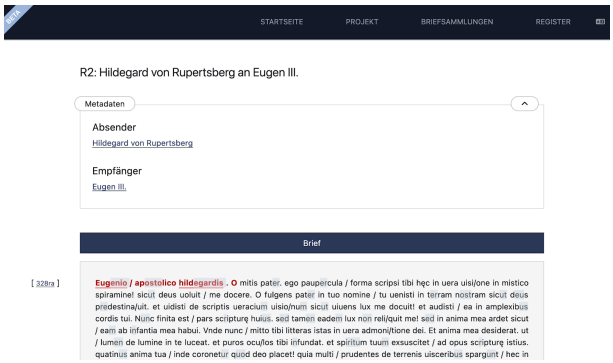

Abb 1: Der Anfang des Briefs R2 der Handschrift R des Liber Epistolarum der Hildegard von Bingen. URL: https://liberepistolarum.mni.thm.de/id/300dcfe1-9f1a-4e21-914d-4730fd85f1d2 (abgerufen am 6.7.2023).

Auf Grundlage dieser Publikationsumgebung wurde im folgenden die Webseite des DFG-Projekts zu den Sozinianischen Briefwechseln (https://gepris.dfg.de/gepris/projekt/3245185) von einer Typo3-basierten Webseite auf eine graphbasierte Publikationsumgebung umgestellt. Dabei wurden die in TEI-XML vorliegenden Briefe mit Hilfe eines TEI2json-Konverters (https://gitlab.rlp.net/adwmainz/digicademy/ sbw/tei2json) in das SPO-JSON-Format konvertiert und konnten so sehr einfach in die graphbasierte Publikationsumgebung eingespielt werden.

Auch wenn auf der Publikationsseite alles reibungslos funktioniert, haben sich im Hildegard-Projekt im Verlauf des Projekts doch einige Herausforderungen ergeben. So werden im Editionssystem Codex (vgl. Abb. 2) Texte, die bearbeitet werden, komplett in den Browser geladen. Bei Briefen mit einer Länge bis zu zehn DINA4-Seiten (nur als Größenvergleich) ist das noch realisierbar. Bei langen Kapiteln mit vielleicht 40 Seiten stößt das System an seine Grenzen, da die Bearbeitung bei größeren Textlängen sehr träge und langsam wird.

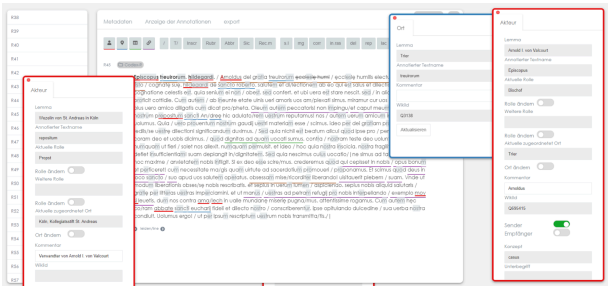

Abb. 2: Das graphbasierte Editionssystem Codex (Neill/Kuczera 2019).

Applied Text as Graph (ATAG)

Mit *Applied Text as Graph* (ATAG) schlagen wir ein neues Konzept von Text as Graph vor, das nicht nur denselben Grad notwendiger Flexibilität mitbringt, wie man ihn bei TagML findet, sondern durch seine Anlehnung an SPO auch eine performantere technische Umsetzung in digitalen Editionsprojekten ermöglicht.

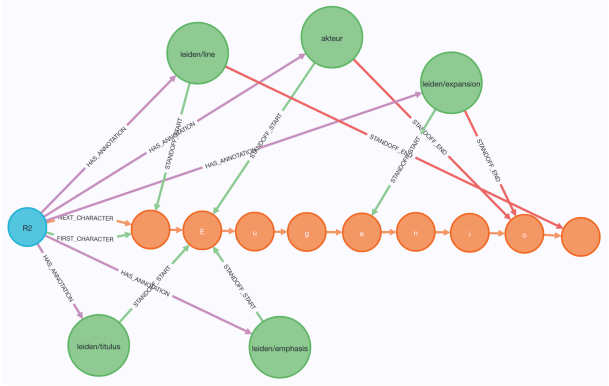

Abb 3.: Das Graphmodell des ersten Wortes des Briefes R2 aus der Handschrift R des Liber Epistolarum der Hildegard von Bingen. Der Grundtext ist normalisiert, die Annotation leiden/expansion zeigt an, dass im Rahmen der Transkription die Zeichen ‘enio’ ergänzt wurden (Quelle: Autor).

Die Grundlagen sind:

- Ein linearer Text (Textstream) bestehend aus Zeichen und Leerzeichen. Dieser lineare Text wird als Textstück bezeichnet.
- Die Zeichen (und Leerzeichen) eines Textstückes werden in Zeichenknoten (Orange Knoten in Abb. 3) abgebildet, die untereinander mit NEXT-Kanten verbunden sind.
- Jedes Textstück beginnt mit einem Textknoten (Blauer Knoten in Abb. 3). Von diesem Textknoten geht eine FIRST_CHARACTER-Kante und eine NEXT-Kante zum ersten Zeichen der Zeichenkette und eine LAST_CHARACTER-Kante zum letzten Zeichen der Zeichenkette.
- Die kleinste Granularität ist die Zeichenebene, deren Reihenfolge mit NEXT-Kanten festgehalten wird.
- Jeder Zeichenknoten ist über eine UUID eindeutig identifizierbar und über eine persistent stabile und auflösende URI über das Internet adressierbar.
- Die Ketten von Zeichenknoten können mit Annotationsknoten annotiert werden (Grüne Knoten in Abb. 3).
- Die Annotationsknoten sind über FIRST_CHARACTER- und LAST_CHARACTER-Kanten mit den Zeichenknoten verbunden und machen die Reichweite einer Annotation explizit.
- Eine Annotation kann mit einem weiteren Textstück verbunden werden, in dem z.B. eine alternative Lesart oder ein Kommentar enthalten ist. Damit ergibt sich ein Netzwerk von Texten und Annotationen mit Texten.
- Der Grundtext (auf den Begriff Basistext wird bewusst verzichtet, da keinerlei Hierarchie hergestellt werden soll) und die Annotationen bilden die Grundlage dieses Textmodellierungssystems.
- Das Ende einer Zeichenknotenkette kann mit einer NEXT-Kante mit dem ersten Zeichenknoten einer weiteren Zeichenknotenkette verbunden werden und gibt damit eine mögliche Leserichtung wieder.
- Das System macht keine Vorgaben, wie z.B. die Metadaten von Texten festgehalten werden. Diese könnten beispielsweise gemäß der gut dokumentierten TEI-Richtlinien in einem Metadatenknoten festgehalten werden, der mit dem Textknoten verbunden ist. Oder die Metadaten werden als Properties direkt im Textknoten abgespeichert.
- Das System macht keine Vorgaben, was ein „Zeichen“ definiert. Diese Definition findet im Projekt im Anwendungszusammenhang statt.
- Festzuhalten bleibt also, dass es lineare Textstücke gibt, die Annotationen haben können und diese Annotationen können wiederum einen Text haben, der Annotationen hat.
- Es ergibt sich ein Netzwerk von Texten.

Das Modell geht davon aus, dass es keine z.B. Diskontinuitäten, keine Nichtlinearitäten oder Löschungen im Text gibt, sondern betrachtet sie als konstitutive Schritte auf dem Weg zu einem lesbaren Text. Die Textphänomene wie z.B. Diskontinuitäten, Löschungen oder Nichtlinearitäten werden in den Annotationen verzeichnet. Im Unterschied zu TEI-XML, wo bei der Verwendung des <choice>-Elements keine Vorgabe für das Lesen des Textes vorgegeben wird, ist bei ATAG die Entscheidung für einen Basistext konstitutiv. Allerdings kann der Nutzer selbst entscheiden, was in den Basistext und was in die Annotationen kommt. Darüber hinaus ermöglicht die Verwendung eines Labeled-Property-Graphen (LPG) die Verwendung von weiteren Properties in den Zeichen- und Annotationsknoten. Damit kann sehr flexibel modelliert werden.

Abbildung 3 zeigt das ATAG-Graphmodell des ersten Wortes des Briefes R2 aus der Handschrift R des Liber Epistolarum der Hildegard von Bingen. Abb. 1 zeigt den gleichen Brief auf der Webseite. Der Grundtext ist normalisiert, die Annotation leiden/expansion zeigt an, dass im Original die Zeichen ‘enio’ fehlen. In Abb. 3 sind auch die

weiteren Annotationen zu sehen, die diesen Textbereich betreffen.

Als zweites Beispiel wird eine Textstelle aus einem Brief des Sozinianerprojekts vorgestellt.

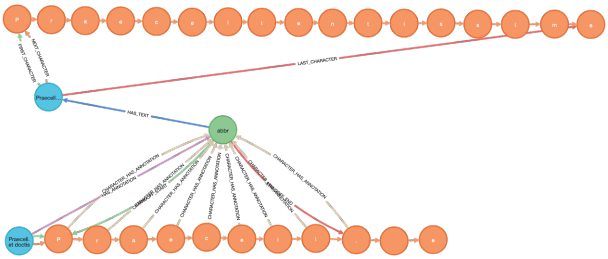

Abb. 4.: Das erste Wort des Briefes von Johannes Müller (Hamburg) an Stanisław Lubieniecki (Hamburg) vom 5. Januar 1665 (https://sozinianer.m-ni.thm.de/view/ed_m3l_ldy _ vkb) abgerufen am 15.7.2023.

Links unten in Abb. 4 befindet sich der blaue Textknoten des Briefes, von dem die Zeichenkette “Praecelli.” startet. Die Zeichen von “P” bis “.” sind von der Annotation *abbr* (grüner Knoten in Abb. 4) umfasst. Vom Annotationsknoten geht eine Kante zum oberen blauen Textknoten, von dem die Zeichenknotenkette mit der normalisierten Fassung des Wortes „Praecellentissime“ startet. Hier wird gut sichtbar, wie sich das Muster zu größeren Strukturen zusammensetzt. Festzuhalten bleibt auch, dass im Sozinianerprojekt im Grundtext die Originalschreibweise festgehalten wird, während die Normalisierung in der Annotation liegt. Im Hildegardprojekt ist hingegen der Grundtext normalisiert und die Ergänzungen annotiert. Diesbezüglich ist das Textmodell agnostisch. Ein Projekt muss allerdings zu Beginn die Entscheidung treffen, wie der Text modelliert werden soll, und sich anschließend daran halten. Prinzipiell sind die verschiedenen Varianten über Graphalgorithmen inneinander überführbar. Daher wird hier auch keine Position in der langen Diskussion eingenommen, was genau ein „Text“ ist, den diese Definition liegt bei den Nutzenden.

Webbasierter Editor für ATAG

Wie bereits oben erwähnt, sind standoff-basierte Textformate schon lange bekannt und werden umfangreich genutzt. Allerdings besteht bei vielen Ansätzen das Problem, dass der Basistext später nicht geändert werden kann, da sich die Indizes sonst verschieben. Mit ATAG wird das Ändern von bereits annotiertem Text möglich.

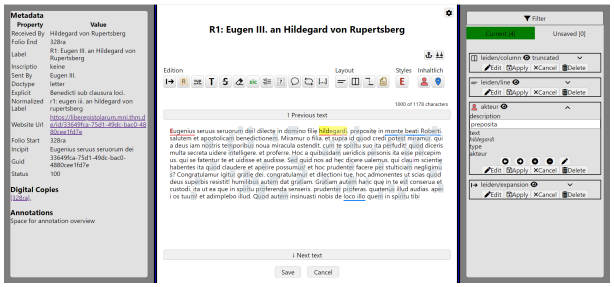

Abb. 5.: Screenshot des ATAG-Editors, der gerade im Rahmen einer Masterarbeit entwickelt wird.

Momentan entsteht im Rahmen einer Masterarbeit ein webbasierter Editor für ATAG. Die Herausforderungen liegen hier vor allem in der Visualisierung der zahlreichen Annotationsebenen und -möglichkeiten. Es müssen sinnvolle Nutzungsszenarien entwickelt werden, die auf die jeweiligen Bedürfnisse der Nutzer zugeschnitten sind. Das Textmodell von ATAG ist sehr eng verwandt mit SPO, die beiden sind direkt ineinander überführbar. SPO wird in den Publikationsumgebungen des Sozinianer-Projekts und des Hildegard-Projekts schon produktiv eingesetzt. Mit einem Algorithmus kann aus SPO-Texten ATAG erstellt werden. ATAG bildet dann die Grundlage für das Edieren oder das Auswerten der annotierten Texte. Sind die Änderungen im Graph gespeichert, können die SPO-Angaben in den Text- und Annotationsknoten neu berechnet werden. Damit können die Änderungen auch direkt in die Publikationsumgebung übernommen werden. Festzuhalten ist, dass die Wahl von Zeichenknoten als kleinste Einheit dem leichteren Management im webbasierten Editor geschuldet ist. Abbildung 4 zu „Praecell.“ zeigt die direkt Verbindung zwischen Zeichenknoten und Annotationsknoten über HAS_ANNOTATION-Kanten. Damit ist für jedes Zeichen unmittelbar feststellbar, mit welchen Annotationen es verbunden ist. Diese Information erleichtert die Darstellung im webbasierten Editor. Es lassen sich ausgehend von den Zeichenknoten aber jederzeit Annotationen erstellen, die dann die Tokenebene abbilden und weitere Zusatzinformationen, wie z.B. das Lemma des Tokens, enthalten können. Grundlage ist aber immer der Grundtext mit den Zeichenknoten.

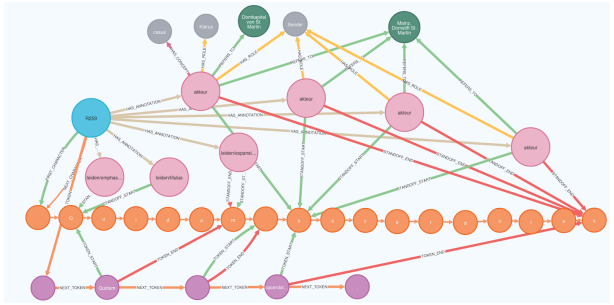

Abb. 6.: Ein Textstück aus dem Brief 259 des Manuskripts R des Liber Epistolarum der Hildegard von Bingen mit Zeichen- (orange) und Tokenebene (fliederfarben).

Abschließend wird in Abbildung 5 der Brief R259 mit einer sehr stark annotierten Stelle, gezeigt bei dem der Zeichenkette “sacerdotes” neben den verschiedenen Layoutannotationen noch vier Akteursannotationen zugeordnet werden, ohne dass das Modell an Klarheit verliert. Die fliederfarbenen Knoten am unteren Rand des Bildes zeigen die Tokenknoten, die jeweils mit der Zeichenebene verbunden sind. Hier wird deutlich, wie das Modell unerschiedliche Granularitätsstufen gemeinsam modellieren kann.

Für einen Test zur Performance wurden die über 200.000 Volltextregesten der Regesta Imperii Online in Neo4j importiert und in ATAG umgewandelt. Der Prozess dauerte auf einer Standardinstallation von Neo4j Desktop ca. 100 Minuten.

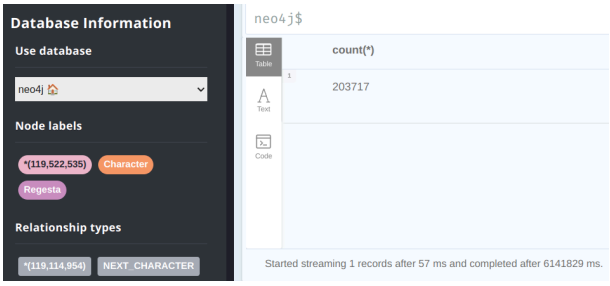

Abb. 7.: Ergebnis des Imports der 203717 Regesten der Regesta Imperii in ATAG.

Nach dem Importprozess hatte die Datenbank knapp 120 Millionen Knoten und knapp 120 Millionen Kanten. Auf die Geschwindigkeit der Queries hatte dies jedoch keine Auswirkungen.

Zusammenfassung

Mit ATAG, oder Applied Text as Graph können Text und Annotationen strukturiert modelliert werden. Im Kern basiert ATAG auf drei Hauptelementen: einem Textknoten, der den Beginn eines Textblocks kennzeichnet, einer Kette von individuellen Zeichenknoten, die den Text ausmachen, und Annotationen, die mit einem anderen Textknoten verknüpft werden können. Eine der flexiblen Eigenschaften von ATAG besteht darin, dass es zwar eine grundlegende Zeichenkette erfordert, jedoch keine Beschränkungen dafür auferlegt, was diese Kette enthalten sollte. Bemerkenswert ist, dass die Basiskette von Zeichen in ATAG dynamisch ist. Sie kann auch nach der Annotation bearbeitet oder modifiziert werden. Dies ist besonders nützlich in kollaborativen Umgebungen oder wenn Informationen aktualisiert werden. Darüber hinaus ist ATAG darauf ausgelegt, eine nahezu unbegrenzte Anzahl paralleler und sich überlappender Annotationen zu verarbeiten, ähnlich den bestehenden Standoff-Systemen, die in der Textanalyse verwendet werden. In Bezug auf zukünftige Arbeiten in ATAG gibt es mehrere Richtungen. Das Framework erforscht die Integration der Standards der Text Encoding Initiative (TEI), um seine Fähigkeiten weiter zu bereichern. Darüber hinaus wird an der Entwicklung eines webbasierten Editors gearbeitet, der die Interaktion mit ATAG benutzerfreundlicher macht. Zusammen mit dem generischen Publikationssystem stehen dann alle für eine graphbasierte digitale Edition notwendigen Komponenten zur Verfügung.

Bibliographie

Bleeker, Elli, Ronald Haentjens Dekker, Bram Buitendijk. 2023. “Texts as Hypergraphs: An Intuitive Representation of Interpretations of Text”, Journal of the Text Encoding Initiative, Issue 14 | April 2021-March 2023, Online since 08 June 2022, connection on 13 March 2023. URL: http://journals.openedition.org/jtei/3919 ; DOI: https://doi.org/10.4000/jtei.3919

Bode, Katherin. 2022. “Doing (Computational) Literary Studies”, New Literary History 53.4-54.1 (2022-23): 531-558.

Kuczera, Andreas. 2022. “TEI Beyond XML – Digital Scholarly Editions as Provenance Knowledge Graphs.” In Tara Andrews, Franziska Diehr, Thomas Efer, Andreas Kuczera and Joris van Zundert (eds.): Graph Technologies in the Humanities - Proceedings 2020, published at http://ceur-ws.org/Vol-3110 , 2022. http://ceur-ws.org/Vol-3110/paper6.pdf .

Neill, Iian, Andreas Kuczera. 2019. “The Codex – an Atlas of Relations.” In Die Modellierung des Zweifels – Schlüsselideen und -konzepte zur graphbasierten Modellierung von Unsicherheiten. Hg. von Andreas Kuczera / Thorsten Wübbena / Thomas Kollatz. Wolfenbüttel 2019. (= Zeitschrift für digitale Geisteswissenschaften / Sonderbände, 4) text/html Format. DOI: 10.17175/sb004_008.

Pichler, Axel, und Nils Reiter. 2022. “From Concepts to Texts and Back: Operationalization as a Core Activity of Digital Humanities.” Journal of Cultural Analytics, vol. 7, no. 4, Dec. 2022, doi:10.22148/001c.57195.

Schmidt, Desmond. 2016. “Using standoff properties for marking-up historical documents in the humanities.“ In it – Information Technology 58 (2016), H. 2, S. 63–69. DOI: 10.1515/itit-2015-0030