Eingereicht: 04.12.2019 Angenommen: 25.04.2020 Publiziert: 09.05.2020

OZeAN
Online Zeitschrift zur Antiken Numismatik

Jahrgang 2 (2020), S. 5–19

Eine Pipeline zur Digitalisierung tabellenbasierter Fundmünzdaten aus PDF-Dokumenten

by Timo Kissinger

doi: https://doi.org/10.17879/ozean-2020-2778

Kontakt: Timo Kissinger, M. A., Akademie der Wissenschaften und der Literatur | Mainz, Digitale Akademie, Geschwister-Scholl-Str. 2, 55131 Mainz, E-Mail: Timo.Kissinger@adwmainz.de

Herausgegeben im Auftrag der Forschungsstelle Antike Numismatik der Westfälischen Wilhelms-Universität Münster
von Achim Lichtenberger, Katharina Martin und Ulrich Werz
http://ozean-numismatik.de/
ISSN 2690-4490

Eine Pipeline zur Digitalisierung tabellenbasierter Fundmünzdaten aus PDF-Dokumenten

Timo Kissinger

Zusammenfassung: Dieser Aufsatz beschreibt ein Verfahren, das es ermöglicht, textbasierte Fundmünzdaten zu extrahieren und in RDF zu übertragen. Entwickelt wurde das Verfahren im Rahmen einer Masterarbeit. Als Grundlage dienen die Publikationen des Projektes »Die Fundmünzen der Römischen Zeit in Deutschland (FMRD)«. In dem Beitrag wird auf die Herausforderung einer solchen Digitalisierung eingegangen und anschließend anhand eines Beispielmünzkomplexes ein Lösungsweg angeboten. Dieser Lösungsweg stellt eine aus mehreren Skripten bestehende Pipeline dar, die es ermöglicht, aus einem PDF die Daten als Text auszugeben, über das Zwischenformat CSV zu modifizieren und anschließend als RDF auszugeben.

Schlagwörter: Fundmünzen, Semantic Web, Linked Open Data, Python, Datenextraktion

Abstract: This paper describes a method to extract text-based coin find data and transfer it to RDF. The method was developed as part of a master thesis. It is based on the publications of the project »Die Fundmünzen der Römischen Zeit in Deutschland (FMRD)«. The paper deals with the challenge of such digitisation and then offers a solution based on an example coin complex. This solution represents a pipeline consisting of several scripts, which makes it possible to output the data from a PDF as text, to modify it via the intermediate format CSV and then to output it as RDF.

Key words: Coin find data, Semantic Web, Linked Open Data, Python, Data extraction

Dieser Aufsatz ist eine Zusammenfassung der Masterarbeit »Die Fundmünzen der römischen Zeit in Deutschland: Ein beispielhaftes Verfahren zur textbasierten Datenextraktion und Auszeichnung von Münzdaten für das Semantic Web«. Die Masterarbeit wurde im Rahmen des Studienganges »Digitale Methodik in den Geistes- und Kulturwissenschaften«[1] in Mainz im Jahr 2019 geschrieben.

Grundlage für die Arbeit stellen die Fundmünzdaten des Projektes »Die Fundmünzen der Römischen Zeit in Deutschland (FMRD)«[1] dar. Das Projekt wurde 1953 gegründet und war ursprünglich bei der Römisch-Germanischen Kommission (RGK) angesiedelt. Ab 1986 bis zu seinem Auslaufen 2009 wurde es an der Akademie der Wissenschaften und der Literatur in Mainz betreut. Ziel des Projektes war es, alle römischen (bzw. antiken) Fundmünzen aufzunehmen, die innerhalb Deutschlands gefunden wurden. Von diesem Gedanken angesteckt folgten dann die Länder Kroatien, Luxemburg, Niederlande und Slowenien mit eigenen Bänden unter der Schirmherrschaft des FMRD-Projektes.

Das FMRD-Projekt publizierte insgesamt 48 Bände mit weit über 300.000 Fundmünzen[3]. Aus dieser Menge an Daten wurde ein Beispieldatensatz ausgewählt, der exemplarisch in einer Masterarbeit zu bewältigen ist. Dieser Beispieldatensatz besteht aus den 1.157 Einzelfunden des Fundkomplexes FMRD IV 3/2 3006,1 der Domgrabung/des Liebfrauen-Areals[4]. Innerhalb der Arbeit ist eine Pipeline entwickelt worden, die es ermöglicht, Fundmünzdaten aus einem PDF auszulesen, über

1 https://www.digitale-methodik.uni-mainz.de/
2 http://www.fda.adwmainz.de/index.php?id=425
3 Wigg-Wolf – Tolle – Kissinger 2019; DOI 10.5281/zenodo.2596095.
4 Radnoti-Alföldi 2006, 119–206.

die Zwischenformate TXT, CSV und XML zu strukturieren und anschließend in das Endformat RDF zu konvertieren. Die Pipeline verfügt über drei Hauptsäulen: der Datenextraktion, dem Datenmodell und der Datentransformation. In diesem Aufsatz wird der Aufbau der Pipeline mit den drei Säulen beschrieben, aber auch eine beispielhafte Abfrage an die Daten gestellt. Der Teil der Abfrage dient dem Zweck einer Visualisierung der Daten, um ein Beispiel für eine Verwendung dieser aufzuzeigen.

Datenextraktion

Das Ausgangsformat für die Pipeline sind die FMRD-Bände als PDF. Da der zeitliche Umfang einer Masterarbeit begrenzt ist, wurde die Domgrabung als Beispieldatensatz ausgewählt. Die Einführung der PDF (Portable Document Format) im Jahr 1993 ermöglichte es, Texte im digitalen Format einheitlich abzubilden, egal welches Betriebssystem und welche Hardware genutzt wird. Dadurch ist das PDF zum heute weitverbreitetsten Austauschformat geworden. Aber auch eben durch die Unveränderlichkeit des PDFs entstehen Probleme, an die Daten in Form einer Extraktion heranzukommen. Vor allem sind Tabellen in PDFs nicht einheitlich formatiert. So gibt es verschiedene Arten der Formatierung, die eine Extraktion erschweren. Es können beispielsweise keine Trennlinien vorhanden sein, die Spalten oder Zeilen können ineinander übergehen, der Tabelleninhalt kann sich über mehr als eine Seite erstrecken oder es gibt leere Zellen, die das Einlesen der Tabellenstruktur erschweren[5]. All diese Problematiken treffen auch auf den hier verwendeten FMRD-Beispieldatensatz zu.

Ein PDF kann grundsätzlich auf drei Arten entstanden sein:

1. Das ›True‹ bzw. digital erschaffene PDF, welches aus z. B. Textverarbeitungsprogrammen wie Word generiert wurde und somit auf Code beruht.
2. Das gescannte PDF, welches nur ein Bild repräsentiert und über keinen Code verfügt.
3. Das durchsuchbare PDF, welches auf einem gescannten PDF beruht, das mit OCR (Optical Character Recognition) bearbeitet wurde. Ein zusätzlicher Textlayer wird dabei dem Bildlayer hinzugefügt, der die bei der Texterkennung gefundenen Texte und die Dokumentstruktur enthält[6].

Es ist möglich, unstrukturierte Daten mit ***Extract, Transform, Load*** (ETL) strukturiert auszugeben[7]. Aber auch der ETL-Prozess benötigt manuelle Eingriffe. Daten aus dem Internet zu gewinnen bzw. zu extrahieren ist heute bereits ein gängiges Verfahren und wird mit dem Begriff Web-Scraping beschrieben[8]. Mit sogenannten Wrappern lassen sich die Daten aus Tabellen auf Webseiten über die HTML-Struktur bzw. XML-Struktur automatisiert beziehen. Es gibt bereits eine Fülle von Literatur und Software zu dieser Thematik[9]. Die Extraktion von tabellarischen Daten aus PDF-Dokumenten ist hingegen weniger oft behandelt worden und erfreut sich erst in jüngerer Zeit einer wachsenden Aufmerksamkeit[10].

Um eine Datenextraktion bei den FMRD-Daten durchzuführen, wurde zunächst überprüft, ob es bereits eine Software gibt, die es ermöglicht, die Tabellendaten aus den FMRD-Bänden korrekt als digitale Tabelle, wie beispielsweise CSV, zu extrahieren. Im Folgenden sollen zwei Beispiele aus diesem Softwarebereich beschrieben werden.

Tabula[11] ist eine Software die speziell dafür entwickelt wurde, um tabellarische Daten in einem PDF erkennen und extrahieren zu können. Die Open-Source-Software läuft auf Linux, Mac und Windows. Der Nutzer kann dem Programm ein PDF übergeben. Nun steht ihm zur Auswahl, ob er die Tabellen im PDF manuell mit einem Rahmen belegen will oder die Tabellen automatisch erkannt werden sollen.

5 Yadav et al. 2018, 2021.
6 Yadav et al. 2018, 2021 f.
7 ETL-Prozess, URL: https://de.wikipedia.org/wiki/ETL-Prozess (aufgerufen am 23.04.2019).
8 Web-Scraping, URL: https://en.wikipedia.org/wiki/Web_scraping (aufgerufen am 23.04.2019).
9 Beispielsweise Vidya 2014, 76–79.
10 Liu et al. 2008, 1312; DOI 10.1145/1458082.1458255.
11 URL: https://tabula.technology/.

Aufgrund der schieren Menge von 48 Bänden wird zukunftsorientiert die automatische Suche an dieser Stelle bevorzugt. Die erkannten oder markierten Bereiche kann sich der Nutzer nun als CSV-Datei pro erkannter Tabelle bzw. Seite ausgeben lassen. Das Ergebnis hängt jedoch davon ab, wie gut Tabula die vorhandene Tabellenstruktur erkennt. Nach dem Einlesen des PDF kann die Tabelle unter »Autodetect Tables« erkannt werden. Tabula markiert die erkannten Tabellen rot.

Dabei ergaben sich im Beispieldatensatz bereits Fehler in der Form, dass Teile des Einführungstextes und der Anmerkungen ebenfalls markiert wurden. Des Weiteren wurden nicht alle Münzzeilen markiert. Auch kam es vor, dass wenn andere Textelemente wie Überschriften oder Literaturangaben auf der gleichen Seite wie die Münzdaten vorhanden waren Tabula die Münzdaten nicht erkannte. Anschließend wurde die Tabelle unter »Preview & Export Extracted Data« betrachtet und als CSV ausgegeben. Das Ergebnis war jedoch nicht zufriedenstellend. Die exportierte CSV enthielt zehn Spalten anstatt der inhaltlich korrekten acht Spalten. Die Daten wurden nicht korrekt erkannt und nicht in ihrer richtigen Spalte ausgegeben. Somit müssten die Daten den Spalten nachträglich korrekt zugeordnet werden, was beispielsweise mit regulären Ausdrücken möglich wäre.

Eine andere Software die Tabellen in PDFs erkennen und extrahieren kann ist pdftohtml. Das Wissenschaftszentrum Berlin für Sozialforschung stellte 2017 in ihrem Data Science Blog den Blogeintrag »Data Mining OCR PDFs — Using pdftabextract to liberate tabular data from scanned documents«[12] ein Paket von Tools zusammen, die es ermöglichen, Tabellen aus PDF-Dokumenten zu extrahieren. Pdftohtml ist ein Teil des Softwarepakets poppler-utils zur PDF-Bearbeitung und ursprünglich für Linux entwickelt worden[13]. Jedoch gibt es auch Versionen für die Nutzung auf Windowsplattformen[14]. Pdftohtml ist in der Lage, aus einem durchsuchbarem PDF über den Befehl –xml ein XML-Dokument zu erzeugen. Die Daten aus dem PDF sind in Text-Boxen im XML vorhanden. Zur Visualisierung des Ergebnisses kann das Tool pdf2xml-viewer[15] verwendet werden. Das Ergebnis der Extraktion sieht dem ursprünglichen PDF sehr ähnlich. In vielen Fällen innerhalb des Beispieldatensatzes sind die Spalten korrekt wiedergegeben. Vor allem wurden auch leere Werte innerhalb der Spalten erkannt und in das XML übernommen. Jedoch finden sich sehr oft Text-Boxen die Daten aus mehreren Spalten als eine einzige erkannt haben. Die Spaltenstruktur der Daten ist also auch nicht von pdftohtml korrekt erfasst worden.

Doch was ist der Grund, warum diese Softwarebeispiele die Tabellendaten aus den FMRD-Bänden nicht klar erkennen können? Der Hauptgrund dafür ist, dass die Daten ursprünglich in Microsoft Word erfasst wurden. Dabei sind die Daten nicht in eine Tabelle geschrieben worden, sondern sie wurden mit Tabstopps eingerückt. Es fehlen also klare Spaltentrenner wie ein senkrechter Strich, der die Werte in den Spalten klar voneinander trennt. Die aus den Worddokumenten erzeugten PDFs können aus diesem Grund nicht über die korrekte Syntax verfügen, die es bräuchte, um die Spalteninhalte klar voneinander getrennt als CSV oder HTML auszugeben. Dieser Fehler erzeugt des Weiteren Folgefehler. So wurde bei den Tabellen öfter mit Zeilenumbrüchen gearbeitet. Durch das Fehlen der Spaltentrenner rutschen daher manchmal die Inhalte mehrere Spalten und Zeilen ineinander und erzeugen somit fehlerhafte Daten. Als Beispiel sei hier Fundmünze 525 der Einzelfunde der Trierer Domgrabung genannt.

[12] Konrad 2017; URL: https://datascience.blog.wzb.eu/2017/02/16/data-mining-ocr-pdfs-using-pdftabextract-to-liberate-tabular-data-from-scanned-documents/ (aufgerufen am 12.03.2019).

[13] https://poppler.freedesktop.org/.

[14] Hubers 2013; URL: https://blog.alivate.com.au/poppler-windows/ (aufgerufen am 14.03.2019).

[15] https://github.com/WZBSocialScienceCenter/pdf2xml-viewer.

Abb. 1: Fundmünze 525 der Einzelfunde der Trierer Domgrabung

Abb. 1a zeigt einen Screenshot aus dem PDF mit den Daten der Münze. In **Abb. 1b** ist das Ergebnis einer Extraktion der Daten als CSV dargestellt. In diesem Beispiel ist erkennbar, dass durch den Zeilenumbruch sich das »RLMT« der Spalte der Konkordanz bei der Extraktion zur CSV in die Spalte der Referenzangaben verirrt hat.

Zwar bieten viele Programme die Möglichkeit die Daten mit regulären Ausdrücken zu bereinigen. Doch ist der Variantenreichtum der FMRD-Daten so groß, dass es mit nur wenigen regulären Ausdrücken nicht getan ist, um die Daten korrekt wiederzugeben. Als Lösung für dieses Problem wurde daher ein Skript geschrieben, dass auf einer Vielzahl von regulären Ausdrücken aufbaut und die Daten, nach manueller Prüfung, korrekt abbildet. Dieses Skript deckt fast alle Varianten der Spaltenwerte des Beispieldatensatzes der Trierer Domgrabung ab. Das Skript konnte nach manueller Zählung bis auf 33 Zeilen, was einem Fehlerquotienten von 2,85 % entspricht, alle Daten korrekt wiedergeben. Die 33 fehlerhaften Zeilen sind teils bedingt durch die oben genannten Zeilenumbrüche. Es lässt sich bisher keine wiederkehrende Struktur bei diesem Fehler erkennen und somit maschinell beheben. Aber auch Tippfehler, die bei der Erfassung der Münzen für die Bände entstanden sind, sind Teil dieser 33 fehlerhaften Zeilen. In anderen Fundmünzkomplexen und Fundmünzbänden werden noch neue Varianten der Tabellenwerte auftauchen. Diese neuen Varianten können dem Skript jedoch hinzugefügt werden und dieses somit erweitert werden.

Dieses Skript ist Teil einer Reihe von Skripten, die zusammen eine Pipeline bilden **(Abb. 2)**. Die verwendete Programmiersprache ist Python. Die Pipeline liest das PDF ein, erzeugt ein neues PDF, das einen bestimmten Fundmünzkomplex eines Bandes repräsentiert, erzeugt ein Textdokument, bereinigt die Daten darin und erzeugt eine CSV-Datei und übergibt die Münzdaten als XML einem Webservice, der die Daten als RDF erzeugt und ausliefert. Die Angaben der Münzen werden zunächst wie

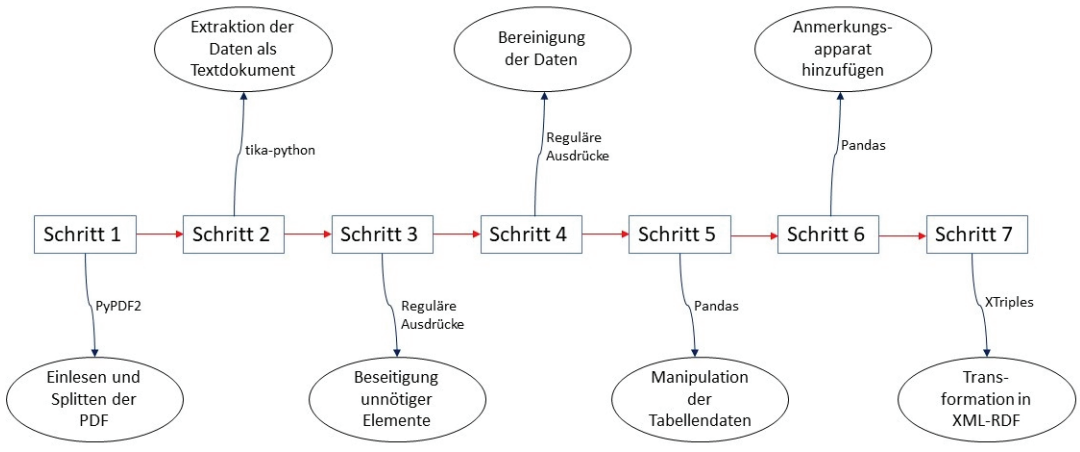

Abb. 2: Die Schrite der Pipeline

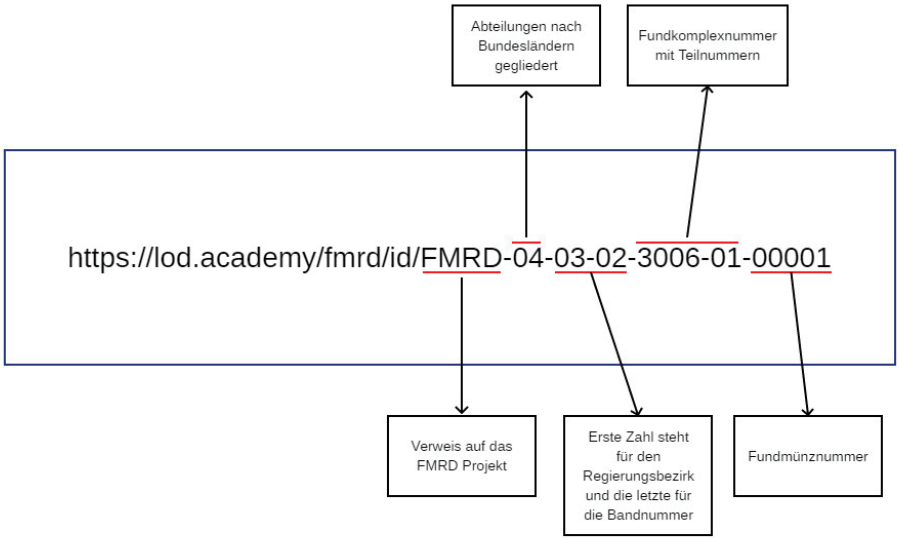

Abb. 3: URI mit Erläuterung der ID

im Band übernommen. Eine Vereinheitlichung der Datierungsangaben, die Anreicherung mit Nomisma-IDs etc. ist durch Folgearbeiten o. ä. möglich, jedoch im Rahmen der Masterarbeit aus Zeitgründen nicht umsetzbar.

Als erster Schritt der Pipeline wird mit PyPDF2[16] das PDF eingelesen und eine Splittung der gewünschten Seiten vorgenommen. Eine Splittung der einzelnen Fundmünzkomplexe ermöglicht ein einzelnes Bearbeiten eben dieser Komplexe. Zumal die Masterarbeit sich auf eben nur einen Komplex fokussiert. PyPDF2 ist eine Python-Bibliothek die es erlaubt, skriptbasiert PDF-Dateien zu bearbeiten und beispielsweise bestimmte Seiten eines PDFs zu separieren. Das somit erzeugte PDF mit den Fundmünzen der Domgrabung wird im zweiten Schritt an die Python-Bibliothek tika-python[17] übergeben. Ein paar Eckdaten zu Tika selbst: Tika ist ursprünglich ein auf Java basierendes Framework namens Apache Tika. Es wurde entwickelt, um Metadaten und Text von unterschiedlichsten Dateiformaten zu erkennen und zu extrahieren[18]. Tika-python nutzt den Tika REST Server[19], um die Funktionalität von Apache Tika für Python nutzbar zu machen.

Die mit tika-python erzeugte Textdatei erlaubt es nun in weiteren Schritten die Tabellendaten zu gewinnen. Dafür wird in einem dritten Schritt, der der Entfernung unnötiger Daten dient, zuerst der einleitende Text, der mit den Tabellendaten in das Textformat extrahiert wurde, mit Hilfe von regulären Ausdrücken erfasst und entfernt. Das gleiche gilt für den Anmerkungsapparat unterhalb der Tabellendaten. Jedoch wird dieser nicht einfach entfernt, sondern als CSV weggeschrieben, da die Anmerkungen in einem späteren Schritt noch einmal Verwendung finden. Die somit von unnötigen Elementen bereinigte Textdatei kann nun in einem vierten Schritt mit regulären Ausdrücken strukturiert werden. Im fünften Schritt spielt die Python Bilbiothek Pandas[20] eine große Rolle. Pandas ist eine Bibliothek, die es ermöglicht, vor allem tabellarische Daten zu manipulieren und zu analysieren. So sind die Prägeherren nicht als separate Spalte in den FMRD-Bänden erfasst worden, sondern immer als Überschrift über den zugehörigen Münznummern angegeben worden. Diese Überschriften können nun mit Pandas über ihren Index als separate Spalte

16 https://pythonhosted.org/PyPDF2/.
17 https://github.com/chrismattmann/tika-python.
18 Apache Tika, URL: https://en.wikipedia.org/wiki/Apache_Tika (aufgerufen am 07.04.2019).
19 Tika REST Server, URL: https://wiki.apache.org/tika/TikaJAXRS (aufgerufen am 07.04.2019).
20 https://pandas.pydata.org/

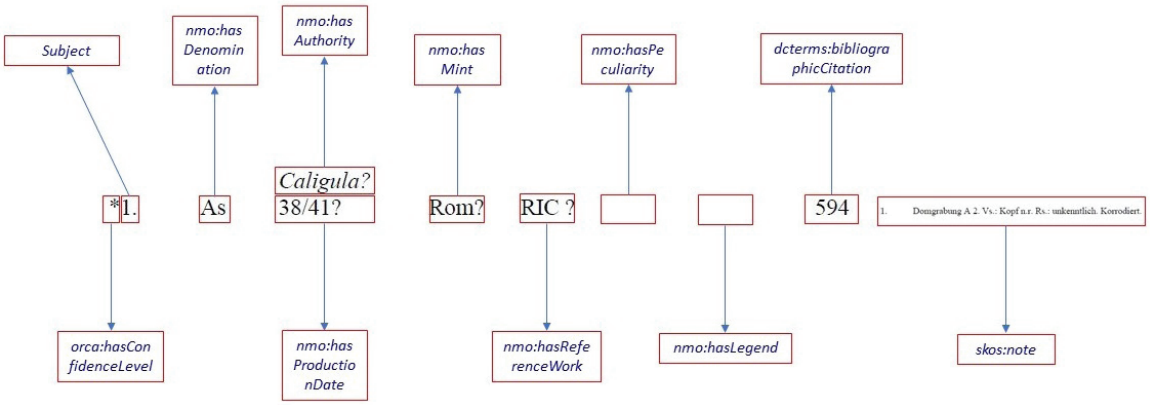

Abb. 4: Visualisierung des Datenmodells

erfasst werden. In den FMRD-Tabellen sind die Werte typengleicher Münzen mit Anführungszeichen angegeben worden. Diese Anführungszeichen wurden nun mit ihren reellen Werten ersetzt.

Im fünften Schritt wird zudem für jede Fundmünze eine eigene ID angelegt. Mit der ID ist es möglich, jeder Fundmünze des FMRD-Projektes eine eigene, feste Adresse zu geben (siehe **Abb. 3**). Die Angabe FMRD gibt hier an, dass die Münze aus einem FMRD-Band stammt. Ausblickend kann so eine Unterscheidung zu anderen Fundmünzprojekten Europas vorgenommen werden. So bedeutet z. B. die ID FMRD-04-03-02-3006-01-00001, dass es sich um den FMRD Band IV (also Rheinland-Pfalz) 3/2 (3 entspricht dem Regierungsbezirk Trier und 2 dem zweiten Band aus der Trierer Reihe) mit dem Fundkomplex 3006,1 (3006 steht für den Komplex der Trierer Domgrabung und 1 für die Einzelfunde) und der Fundmünze 1 handelt.

Das Ergebnis aus Schritt fünf wird als CSV gespeichert. Diese Tabelle wird mit dem Anmerkungsapparat anschließend in Schritt 6 in einem neuen CSV-Dokument zusammengefügt. Mit Pandas wird der Anmerkungsapparat dabei als weitere Spalte ergänzt. Beim Speichern werden leere Zellen mit NaN (Not a Number) ersetzt. Dies erfüllt die Funktion, dass die Zellen nicht einfach leer sind, sondern als undefinierter bzw. nicht darstellbarer Wert ausgezeichnet werden. Somit erhält jede Zelle einen Wert und es wird damit deutlich gemacht, dass die Zellen nicht fehlerhaft leer sind, sondern absichtlich hier nichts vorhanden ist. Auch lassen sich bei Abfragen diese NaN-Werte gezielt ausgeben.

Datenmodell

Im Folgenden wird ein Datenmodell beschrieben, das aufzeigt wie die Tabellenspalten in Form von Ontologien zu RDF-Triples umgewandelt werden können. Ontologien stellen ein Vokabular bereit, um mit Subjekt-Prädikat-Objekt-Beziehungen (RDF-Triples) bestimmte Dinge auszuzeichnen. Dinge können dabei, wie hier, das abstrakte Konzept einer Münze sein. Von zentraler Bedeutung bei der späteren Implementierung des Modells sind die Spalten des FMRD-Komplexes, bzw. der CSV-Datei, da diese in RDF überführt werden. Die für Münzen führende Ontologie ist von Nomisma.org und wird in dieser Arbeit am meisten verwendet. Daher seien hier ein paar Eckdaten der Ontologie und dem Bezug zu Fundmünzen genannt: Die Ontologie von Nomisma.org wurde speziell für die Modellierung von Münzdaten entwickelt. Dabei wurde bisher das Augenmerk besonders auf Münzbestände in Sammlungen gelegt. Fundmünzen wurden bisher nur geringfügig von der Ontologie berücksichtigt. Zukünftig sollen Münzen aus archäologischen Kontexten jedoch stärkere Beachtung bei Nomisma finden[21]. Nichtsdestoweniger lassen sich bereits

[21] Siehe hierzu beispielsweise den Workshop »Coins in Context« vom 24. bis 25. September 2018 in Oxford: https://www.greekcoinage.org/coins-in-context.html (aufgerufen am 25.05.2019).

jetzt schon Fundmünzdaten mit der Ontologie auszeichnen. Das Modell wurde für Fundmünze Nummer eins aus der Trierer Domgrabung entwickelt (**Abb. 4**).

Als Grundlage dient hier ein Screenshot aus dem PDF mit der ersten Fundmünze. Aus dem Anmerkungsapparat wurde die entsprechende Anmerkung der Tabelle hinzugefügt. Bevor im Folgenden auf die einzelnen Spalten in ihrer Reihenfolge von links nach rechts eingegangen wird, muss zunächst die Spalte der Münznummer separiert betrachtet werden. Die Münznummer stellt für jeden Münzkomplex einen einmaligen Identifier dar, der für Fundmünze Nummer eins dieses Aussehen hat: ***https://lod.academy/fmrd/id/FMRD-04-03-02-3006-01-00001***. Diese URI ist das Subjekt aller RDF-Tripples. D. h., dass alle weiteren Spalten der Tabelle diese URI mit weiteren Informationen zu dem Prägeherrn, der Datierung, etc. anreichern. Die Prädikate sind nach einem Key-Value Prinzip aufgebaut. So findet sich für die Ontologie von Nomisma.org immer ein nmo gefolgt von einem Doppelpunkt und dann dem eigentlichen Prädikat. Die erste Spalte enthält einen Stern. Dieser gibt an, dass der Bandbearbeiter die Münze im Original gesehen hat. Die Angabe, ob ein Bearbeiter eine Münze im Original in der Hand hatte, gibt Informationen darüber, ob die Münze nach neuesten numismatischen Methoden (jedenfalls zum Zeitpunkt der Erfassung der Münze im FMRD-Band) erfasst wurde oder ob diese aus Literaturangaben übernommen wurde. Letzteres stellt beispielsweise oft Probleme dar, wenn sich Informationen in älterer Literatur finden lassen, die beispielsweise nur angeben, dass »Münzen aus dem 3. Jahrhundert n. Chr. gefunden wurden«. Für diese Spalte wurde die Ontologie ***Ontology of Reasoning, Certainty and Attribution*** (ORCA) gewählt. ORCA ist eine Ontologie, die speziell dafür entwickelt wurde, um auszudrücken, wie unsicher bzw. sicher die Angabe von Informationen ist. Die Spalte wurde mit dem Prädikat ***hasConfidenceLevel*** ausgezeichnet22. Dieses Prädikat gibt an, wie sicher die Information ist. In der CSV, bzw. im RDF wurde der Stern durch ***orca:DoxasticKnowledge oder orca:DubitativeKnowledge*** ersetzt. Dem Prädikat hasConfidenceLevel werden somit Objekte der beiden oben genannten Klassen zugeschrieben. Das Objekt des Triples orca:DoxasticKnowledge (Münze lag dem Bearbeiter als Original vor) und orca:DubitativeKnowledge (Münze lag dem Bearbeiter nicht vor) gibt somit die Zweifelhaftigkeit der Bestimmung wieder. So lassen sich in einer Abfrage beispielsweise alle im Original vorhanden Münzen mit der Suche nach orca:DoxasticKnowledge ausgeben.

Das erste Prädikat, das ein numismatisches Konzept darstellt, ist ***hasAuthority*** und entstammt der Ontologie von Nomisma.org. Die Angabe des Prägeherren ist im Band als Überschrift notiert worden, findet sich aber in der CSV als Tabellenspalte wieder. Die verwendeten Prädikate müssen semantisch die Spalten der Tabelle wiederspiegeln. Das Prädikat ***hasAuthority*** wurde hier gewählt, da es mit seiner Beschreibung die münzverausgabende Person, Organisation, etc. wiederspiegelt[23]. Da diese Angabe der FMRD-Bände nicht nur den Prägeherren wie Caligula, sondern auch z. B. die Kelten enthalten kann, ist das Prädikat ***hasAuthority*** bestens geeignet, um den Inhalt der Spalte zu beschreiben. Mit hasDenomination wird das Nominal, als nächste Spalte, beschrieben[24]. Das Prädikat hasProductionDate zeichnet hingegen das Prägedatum einer Münze[25].

Mit dem Prädikat ***hasMint*** wird die Prägestätte benannt, in der die Münze geprägt wurde[26]. Mit ***hasReferenceWork*** wird hingegen das Referenzwerk ausgezeichnet[27]. Ein Referenzwerk kann beispielsweise der RIC sein. Im RIC sind die

22 http://vocab.deri.ie/orca#hasConfidenceLevel (Stand 9. November 2012).
23 http://nomisma.org/ontology#hasAuthority (Stand 27. September 2018).
24 http://nomisma.org/ontology#hasDenomination (Stand 27. September 2018).
25 http://nomisma.org/ontology# hasProductionDate (Stand 27. September 2018).
26 http://nomisma.org/ontology#hasMint (Stand 27. September 2018).
27 http://nomisma.org/ontology#hasReferenceWork (Stand 27. September 2018).

Münzen des Römischen Kaiserreichs nach Typen erfasst. Manche Münze weist etwas Ungewöhnliches auf, also eine Besonderheit. Dies könnte ein Gegenstempel oder ein Loch sein. Die Ontologie von Nomisma.org bietet hierfür das Prädikat ***hasPeculiarity*** an[28]. Vor allem die Prägestätte bestreffende Inschriftenangaben auf der Münze sind ebenfalls vom FMRD-Projekt in einer gesonderten Spalte erfasst worden. Für Legenden (auch wenn es sich hier nur um einen Ausschnitt der gesamten Münzlegende handelt) findet sich bei Nomisma.org das Prädikat ***hasLegend***[29]. Die nächste im LOD-Modell auszuzeichnende Spalte ist die der Konkordanz. In dieser Spalte wird innerhalb der FMRD-Bände auf eine Referenz zur entsprechenden Münze in z. B. einem Archiv hingewiesen. Bei der Ontologie von Nomisma.org ist jedoch kein Prädikat vorhanden, welches die Konkordanz wiederspiegeln könnte. Aus diesem Grund muss für die Konkordanz eine andere Ontologie herangezogen werden. Die Ontologie der ***Dublin Core Metadata Initiative*** (DCMI)[30] bietet für bibliographische Angaben eine weit verbreitete Ontologie. Diese wurde entwickelt, um Dokumente und andere Objekte über Metadaten im Internet zu beschreiben und zu vernetzen[31]. Mit dem Prädikat ***bibliographicCitation*** lässt sich die Angabe der Konkordanz auszeichnen[32]. Die bibliographische Angabe der Konkordanz entspricht exakt der im FMRD-Band gelisteten Münze. Auch für die Spalte der Anmerkungen musste auf eine andere Ontologie als die von Nomisma zurückgegriffen werden. Nomisma.org bietet zwar Prädikate wie hasAppearance für das Erscheinungsbild einer Münze oder ***hasFindspot***, um die Umstände des Auffindens der Münze zu beschreiben, doch dies reicht nicht aus, um die Spalte Anmerkungen passend zu beschreiben. So sind dort diverseste Angaben über die Münze, wie z. B. über das Erscheinungsbild, die Fundumstände oder die Inschrift gebündelt vorhanden. Es ist daher nötig, ein Prädikat zu finden, das semantisch dem deutschen Wort ›Anmerkung‹ ähnelt und die Spalte am treffendsten beschreibt. Gefunden wurde das Prädikat in der Ontologie SKOS[33]. Mit note bietet es ein Prädikat, welches dem deutschen Wort ›Anmerkung‹ entspricht[34]. Über diese Prädikate ist es möglich, ein LOD-Modell für die Fundmünzkomplexe der FMRD-Reihe zu erstellen. Im Kapitel der Datentransformation wird dieses Modell eine bedeutende Rolle spielen, da es in implementierter Form auf den ganzen Fundmünzkomplex angewandt wird.

Datentransformation

Für die Datentransformation ist das oben beschriebene Datenmodell wie auch das Ergebnis der Datenextraktion von zentraler Bedeutung. Das Datenmodell wurde beispielhaft auf eine Münze manuell in XML-RDF angewandt und wird nun für den gesamten Fundkomplex der Einzelfunde der Trierer Domgrabung mit 1.157 Münzen übertragen. So ist es möglich, die Daten aus dem CSV-Zwischenformat der Datenextraktion nach XML bzw. XML-RDF zu transformieren. In der Pipeline wird nun das siebte Skript ausgeführt. Über Pandas werden die Spalten der CSV-Datei in ein XML Dokument mit folgendem Schema umgewandelt:

```
<row>
  <cell role="Spaltenname der CSV">Wert</cell>
                    .
                .
                    .
</row>
```

[28] http://nomisma.org/ontology#hasPeculiarity (Stand 27. September 2018).
[29] http://nomisma.org/ontology#hasLegend (Stand 27. September 2018).
[30] http://www.dublincore.org/specifications/dublin-core/dcmi-terms/.
[31] DCMI, URL: https://en.wikipedia.org/wiki/Dublin_Core (aufgerufen am 28.05.2019).
[32] http://dublincore.org/specifications/dublin-core/dcmi-terms/2012-06-14/?v=terms#bibliographicCitation (Stand 14. Juni 2012).
[33] https://www.w3.org/2009/08/skos-reference/skos.html.
[34] https://www.w3.org/2009/08/skos-reference/skos.html#note und https://www.w3.org/TR/skos-reference/#notes (Stand 18. August 2009).

Dies ist nötig, da der Webservice XTriples[35] ein XML-Dokument zur Erzeugung des Serialisierungsformats RDF-XML benötigt. XTriples wurde an der Digitalen Akademie in Mainz entwickelt. Dieser Service ist speziell dafür geschaffen worden, um automatisiert aus XML-Dokumenten Serialisierungsformate wie RDF erzeugen zu können. Der Webservice ermöglicht es, nach der Übergabe eines XML Dokumentes mit sogenannten Statements aus den XML Tags RDF Tags zu erzeugen. Hierbei wird die URI einer XML-Ressource bzw. einer Dateneinheit in dieser Ressource als Subjekt angesehen. Diesem Subjekt können nun über Prädikate kontrollierter Vokabulare Objekte zugewiesen werden. Somit ist es XTriples möglich, semantische Aussagen von XML-Datenbeständen mit Hilfe kontrollierter Vokabulare zu generieren[36]. Der Webservice ist in der Lage verschiedene XML-Datenbestände zu crawlen. Über einen Request wird XTriples die auf XPATH/XQuery basierende Konfiguration mit den Vokabularen und den Statements übergeben. Über einen Response wird dann das Extraktionsergebnis zurückgegeben. Dabei stehen neben RDF-XML verschiedene weitere Serialisierungsformate, wie Turtle, JSON oder SVG, zur Verfügung[37].

Über die Bibliothek Requests[38] lässt sich ein HTTP-Request über Python an XTriples stellen[39]. Dieser enthält die Statements für die Umwandlung in RDF, wie hier ausschnitthaft angegeben:

```
<xtriples>
  <configuration>
    <vocabularies>
      <vocabulary prefix="dcterms" uri="http://
purl.org/dc/terms/"/>
      <vocabulary prefix="fmrd" uri="https://lod.
academy/fmrd/id/"/>
      <vocabulary prefix="nmo" uri="http://
nomisma.org/ontology#"/>
      <vocabulary prefix="orca" uri="http://vocab.
deri.ie/orca#"/>
      <vocabulary prefix="rdf" uri="http://www.
w3.org/1999/02/22-rdf-syntax-ns#"/>
      <vocabulary prefix="rdfs" uri="http://www.
w3.org/2000/01/rdf-schema#"/>
      <vocabulary prefix="skos" uri="http://www.
w3.org/2004/02/skos/core#"/>
      <vocabulary prefix="tei" uri="http://www.
tei-c.org/ns/1.0"/>
      <vocabulary prefix="void" uri="http://rdfs.
org/ns/void#"/>
    </vocabularies>
    <triples>
      <statement>
        <subject prefix="fmrd">//cell[@role =
'Muenznummer']</subject>
        <predicate prefix="rdf">type</predicate>
        <object type="uri"
prefix="skos">Concept</object>
      </statement>
                    .
                    .
                    .
    </triples>
  </configuration>
  <collection uri="http://www.digitale-akademie.
de/fileadmin/user_upload/fmrd/muenzdaten_XML.
xml">
    <resource uri="{//table/row}"/>
  </collection>
</xtriples>
```

Das Pythonskript führt den Request als Teil der Pipeline vollautomatisch aus und speichert die Münzdaten des Response im Endformat RDF. Wie im Codeausschnitt zu sehen, sind die verwendeten Vokabulare in ***vocabularies*** eingebunden worden. Das Datenmodell findet sich in den Statements wieder. Unter ***collection*** findet sich die URL zu der XML-Datei.

Abfrage

Das RDF-Dokument kann nun auf einem Triplestore wie rdf4j[40] abgelegt und über die Abfragesprache SPARQL abgefragt und visualisiert werden. Eine Visualisierung ist an dieser Stelle nur ein Beispiel, was im Anschluss mit den Daten getan werden kann. Visualisierungen vereinfachen oft den Überblick über große und kom-

35 https://xtriples.lod.academy/index.html.
36 Schrade 2016, 233.
37 Siehe XTriples Dokumentation, URL: https://xtriples.lod.academy/documentation.html (aufgerufen am 26.06.2019).
38 https://2.python-requests.org/en/master/.
39 https://2.python-requests.org/en/master/.
40 https://rdf4j.eclipse.org/.

plexe Datenmengen. Aus diesem Grund eignen sie sich auch bestens dazu, große Fundmünzkomplexe abzubilden. Für die Fundmünzen der Trierer Domgrabung sind beispielhaft sechs Visualisierungen angefertigt worden. Grundlage für die Visualisierungen sind SPARQL-Abfragen. Mit den Abfragen werden alle Fundmünzen nach den Jahrhunderten, in denen sie geprägt wurden, und nach Nominalen gegliedert ausgegeben. Für die Einteilung nach Jahrhunderten wurde der Tabelle eine weitere Spalte mit den Prägedaten der Fundmünzen in Hundertjahreseinteilungen angelegt. Dies erleichtert die Visualisierung, da die Angaben der genauen Prägedaten sehr variantenreich sind und so unübersichtliche Visualisierungen erzeugen. Für eine beispielhafte Visualisierung reicht die Einteilung nach Jahrhunderten aus. Die Diagramme sollen nur das Potential einer weiteren Verwendung der Daten als RDF aufzeigen. Eine Einteilung in historische Phasen, z. B. nach den Perioden von Richard Reece, kann in weiteren Arbeiten geschehen. Als Beispiel sei hier die SPARQL-Abfrage für das 1. Jahrhundert n. Chr. des Fundmünzkomplexes der Trierer Domgrabung angegeben:

```
SELECT ?nominal (COUNT(?date) AS ?count)
WHERE {
  ?coin dcterms:date ?date .
  ?coin nmo:hasDenomination ?nominal .
  FILTER regex(str(?date), ,1\\. Jh\\. n\\. Chr\\.')
}
GROUP BY ?date ?nominal
ORDER BY ?date
```

Erstellt wurden die Diagramme mit der JavaScript Bibliothek sgvizler[41]. Für die Fundmünzdaten wurden Kuchendiagramme als Diagrammart gewählt. Sgvizler nutzt Google Charts[42], um Daten zu visualisieren. Nach einem Test der unterschiedlichen Visualisierungsformen von Google Charts hat sich ergeben, dass bei der Anzeige der Münzdaten beim Kuchendiagramm die Daten sich am übersichtlichsten darstellen lassen. Bei den anderen Darstellungsarten von Google Charts sind die Prozentangaben der Nominale zu eng und damit zu unübersichtlich angegeben. Die Abbildungen sind hier als Screenshots angegeben. Google Charts sind jedoch interaktiv. So sind bei vielen Nominalen die Legenden über zwei Spalten angegeben. Auch weichen die Farben der Nominale bei jeder neuen Abfrage in den Diagrammen voneinander ab. Die Nominalbezeichnungen sind vom Datensatz abhängig und daher bei KENOM ausgeschrieben und bei FMRD abgekürzt. Eine Livevisualisierung aus den Abfragen mit allen Features kann unter https://tkissing.pages.gitlab.rlp.net/fmrd_ma_website/sites/abfrage_und_visualisierung.html betrachtet werden. Auch können dort die Mengenangaben der einzelnen Nominale genauer betrachtet werden, indem der Mauszeiger über die Diagramme bewegt wird.

Als optischer Vergleich wurde der RDF-Datensatz von KENOM (»Kooperative Erschließung und Nutzung der Objektdaten von Münzsammlungen«) herangezogen. Dieser wurde unter der CC BY-NC-ND 4.0 auf Nomisma.org veröffentlicht[43]. KENOM ist ein virtuelles Münzkabinett, welches Bestände wissenschaftlicher Sammlungen online stellt[44]. Der Datensatz dient als rein optischer Vergleich, da ein archäologischer Kontext hier nicht gegeben ist. Das Ziel der Masterarbeit ist es, ein Verfahren zu entwickeln, das die FMRD-Daten extrahieren und als RDF zur Verfügung stellen kann. Die Visualisierung der Daten soll nur als Beispiel dienen, was mit dem RDF danach gemacht werden kann. Dies könnte eben ein Vergleich mit anderen Münzkomplexen sein. Tiefergehende Fundmünzforschung anhand dessen ist aber das Thema anderer Arbeiten oder gar Projektvorhaben. Die Visualisierungen der Münzdaten sehen folgendermaßen aus:

41 http://mgskjaeveland.github.io/sgvizler/.
42 https://developers.google.com/chart.
43 http://nomisma.org/datasets.
44 »Kooperative Erschließung und Nutzung der Objektdaten von Münzsammlungen«, s. https://www.kenom.de.

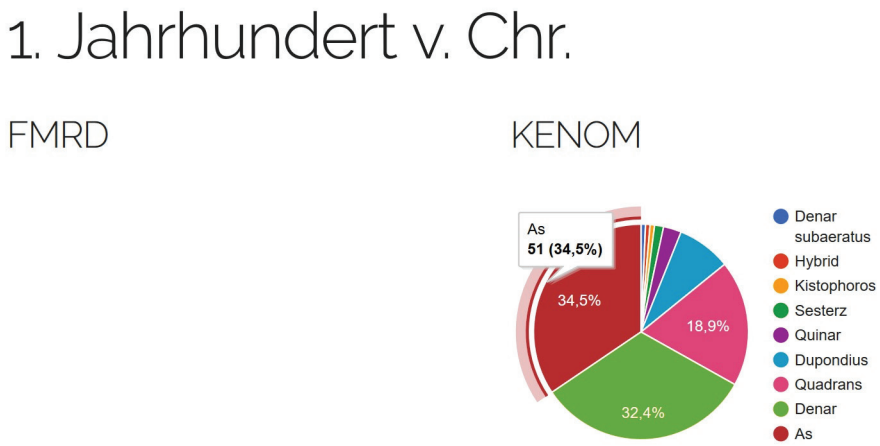

Abb. 5: Visualisierung 1. Jahrhundert v. Chr. (Gesamtmengen: FMRD: 0, KENOM: 148)

Die Fundmünzdaten der Trierer Domgrabung sind hier verkürzt als FMRD angegeben. Für das 1. Jahrhundert v. Chr., bzw. für die vorchristliche Zeit **(Abb. 5)**, zeigt sich, dass bei der Domgrabung keine Münzen vorhanden sind, im Gegensatz zum KENOM-Datensatz. Das Fehlen von Münzen eines bestimmten Zeitraumes kann ebenfalls Informationen liefern. Gerade in der umstrittenen Gründungsphase Triers[45]. So deuten im Bereich der Domgrabung beispielsweise keine Münzen darauf hin, dass hier eine keltische Vorbesiedlung vorhanden war. Bei KENOM dominiert hingegen der As das Kuchendiagramm mit 51 Exemplaren und 34,5 %.

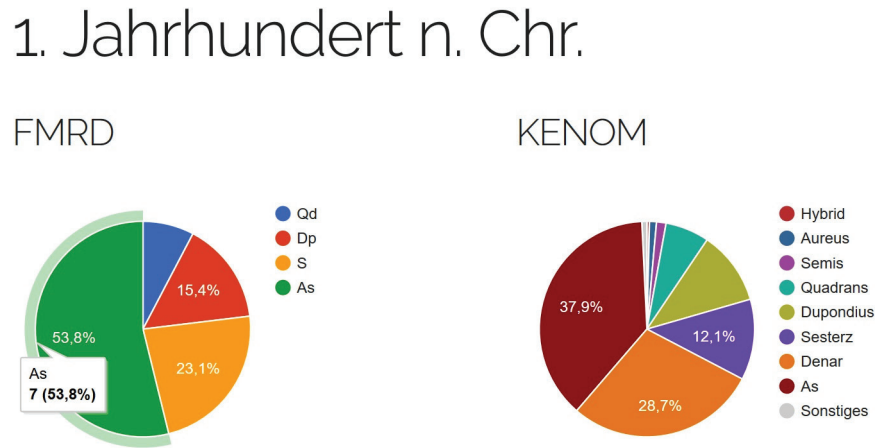

Abb. 6: Visualisierung 1. Jahrhundert n. Chr. (Gesamtmengen: FMRD: 13, KENOM: 1039)

Im 1. Jahrhundert n. Chr. **(Abb. 6)** tauchen auch bei der Domgrabung das erste Mal Münzen auf. Der As dominiert hier das Bild mit sieben Exemplaren und 53,8 %. Bei KENOM ist dies ebenfalls der As, jedoch mit 397 Exemplaren und 37,9 %.

[45] Siehe u. a. Morscheiser 2009.

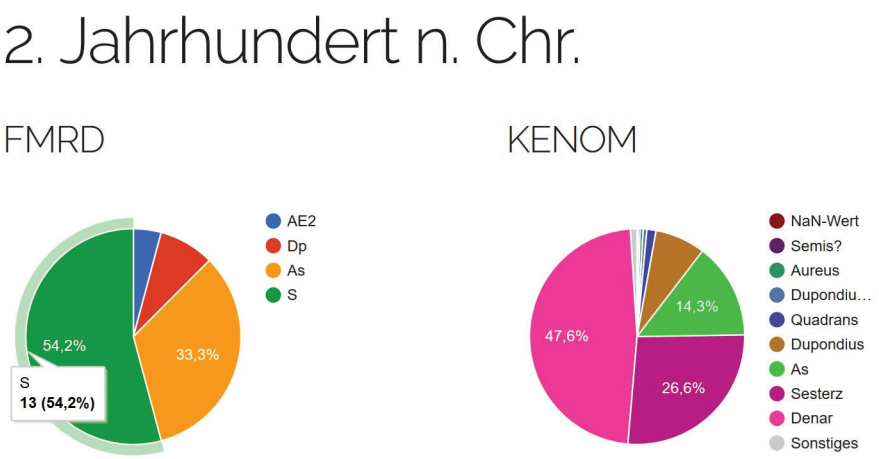

Abb. 7: Visualisierung 2. Jahrhundert n. Chr. (Gesamtmengen: FMRD: 24, KENOM: 2146)

Im 2. Jahrhundert **(Abb. 7)** überwiegt der Sesterz bei der Domgrabung mit 13 Exemplaren und 54,2 %, wobei dies bei KENOM der Denar mit 1.037 Stücken und 47,6 % ist. Es sind somit Münzen des 1. und 2. Jahrhunderts n. Chr. im Areal des Trierer Domes gefunden worden, jedoch ist ihre Anzahl noch relativ gering.

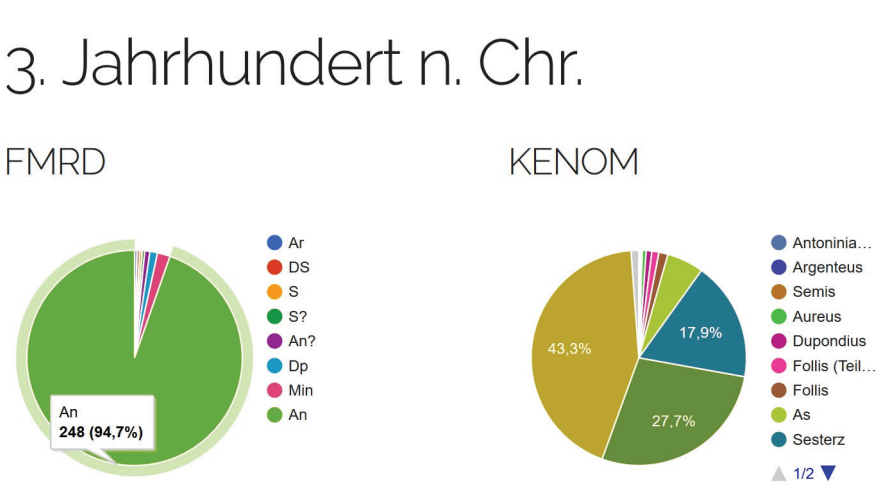

Abb. 8: Visualisierung 3. Jahrhundert n. Chr. (Gesamtmengen: FMRD: 262, KENOM: 2529)

Im 3. Jahrhundert n. Chr. **(Abb. 8)** ändert sich die Anzahl der Fundmünzen bei der Domgrabung schlagartig und der Antoninian dominiert hier das Diagramm mit 248 Stücken und 94,7 %. Bei KENOM ist das meist vertretene Nominal ebenfalls der Antoninian, jedoch mit nur 43,3 %, die aber 1.108 Stücke ausmachen.

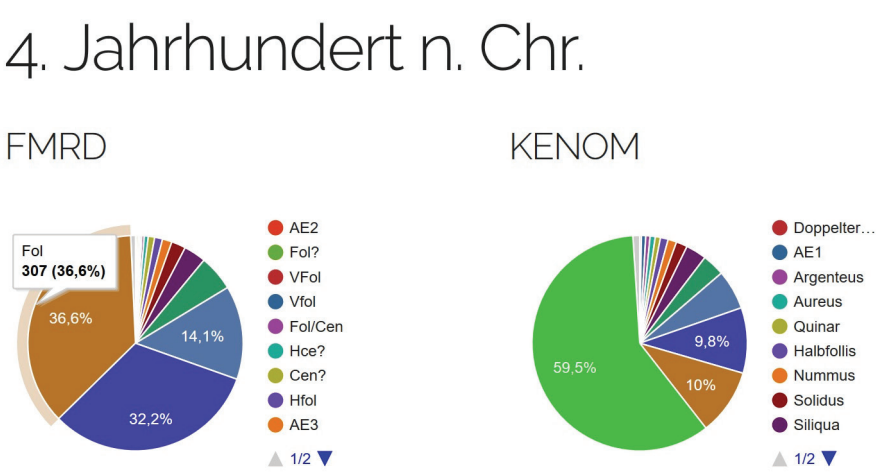

Abb. 9: Visualisierung 4. Jahrhundert n. Chr. (Gesamtmengen: FMRD: 832, KENOM: 765)

Das 4. Jahrhundert n. Chr. **(Abb. 9)** wird vom spätantiken Nominal des Follis dominiert. Es macht bei der Domgrabung mit 307 Exemplaren 36,6 % aus und bei KENOM 59,5 % mit 460 Exemplaren. Fundmünzen des 3. und 4. Jahrhunderts n. Chr. sind im Areal des Trierer Domes somit stärker vertreten als die ersten beiden Jahrhunderte.

Das 5. Jahrhundert n. Chr. (Gesamtmengen: FMRD: 1, KENOM: 17) ist mit nur 1 Münze bei der Domgrabung vertreten – einer Siliqua. Aufgrund der geringen Fundmenge wird beim 5. Jahrhundert auf das Diagramm verzichtet. Bei KENOM sind es alleine 14 goldene Solidi. Wie sich zeigt, ist bei der Trierer Domgrabung das Übergewicht an verlorenen Münzen in der Spätantike. Dies könnte so interpretiert werden, dass das Gelände im Bereich des heutigen Domes in römischer Zeit vor allem in der Spätantike und vielleicht danach genutzt wurde. Da der Trierer Dom auf römischen Fundamenten steht, würde das Bild der Visualisierungen dazu passen und je nachdem, wie lange die Münzen nach ihrer Prägung im Umlauf waren. Die bisherigen Datierungsansätze deuten in das 4. Jahrhundert n. Chr.[46]. Eine genaue numismatische bzw. archäologische Analyse zur Datierung der römischen Strukturen oder der Umlaufzeit römischer Münzen soll an dieser Stelle jedoch ausbleiben, da sie nicht Ziel dieser Arbeit sind. So sei noch gesagt, dass im KENOM-Datensatz sammlungsbedingt natürlich mehr Münzen aus Edelmetall vertreten sind – wie es das 5. Jahrhundert zeigt.

Fazit

Diese Arbeit beschäftigt sich mit einem Verfahren, das es ermöglicht, textbasierte Fundmünzdaten in RDF umzuwandeln. Wie oben beschrieben, ist bisher keine Software vorhanden, die automatisiert Münzdaten aus PDF-Versionen der FMRD-Bände sauber extrahieren kann. Dass die Münzdaten in einer strukturierten Form extrahiert werden, ist von wesentlicher Bedeutung für die weitere Verarbeitung der Daten. Die Datenextraktion ist daher der erste wichtige Schritt im Aufbau einer Pipeline. Die hier vorgestellte Pipeline kann nach der Extraktion die Daten über informatische Methoden weiter modifizieren. So kann nicht nur eine CSV erzeugt werden, die die Spalten eines FMRD-Bandes wiedergibt, sondern diese Spalte können noch erweitert werden. So sind beispielsweise auch die Prägeherren bzw. prägende Autorität und der Anmerkungsapparat als separate Spalte erfasst worden.

Die Pipeline besteht aus verschiedenen Skripten, die aufeinander aufbauen. Das Ein-

[46] Weber 2010, 183–186.

lesen der PDF mit dem gesamten FMRD-Band, das Splitten der Seiten des entsprechenden Fundmünzkomplexes, das Extrahieren der Münzdaten in ein Textdokument, das Erzeugen einer CSV-Datei mit den korrekten Münzdaten, die Umwandlung der CSV in XML und die Erzeugung des RDF mittels des Webservices XTriples werden automatisiert durchgeführt. Es konnten jedoch nicht alle Fehler, die während der Extraktion entstanden sind, rein skriptbasiert gelöst werden. Manche Fehler in der Extraktion müssen manuell gelöst werden, da sie in kein Schema passen. Außerdem sind manche Dinge, wie die Seitenzahlen des Münzkomplexes im PDF-Band, noch manuell anzugeben. Während der Datenextraktion wird des Weiteren die Münznummer zu einer ID umgewandelt. So ist es möglich, für alle Münzen die im FMRD-Projekt aufgenommen wurden eine feste ID pro Münze zu erzeugen. Für einen ›Vorher-Nachher-Vergleich‹ sind an diesen Aufsatz drei Dateien angehängt. Die Datei text_Output.txt enthält die Extraktion als Textdatei, die CSV muenzdaten_mit_Anm_und_NaN.csv verfügt über alle Münzdaten in einer strukturierten Tabelle und die RDF-Datei muenzdaten_als_RDF.rdf stellt das Endergebnis mit den Münzdaten als RDF-Triples dar[47].

Dies ist ein erster Prototyp, wie er im zeitlichen Rahmen einer Masterarbeit entwickelt werden kann. Das Skript kann über diese Arbeit hinaus nun erweitert werden, indem die regulären Ausdrücke für andere Fundmünzkomplexe bzw. Bände ausgebaut werden. Auch ist es zukünftig sicher sinnvoll, die FMRD-Daten mit Normdaten von Nomisma.org skriptbasiert anzureichern. Jedoch muss hierfür zunächst bei älteren FMRD-Bänden die Spalte der Referenzwerke mit der inzwischen veralteten Literatur überarbeitet werden. Diese könnten beispielsweise mit einer Konkordanztabelle skriptbasiert abgeglichen werden und für den jeweiligen Münztyp das veraltete Referenzwerk durch das in der Forschung aktuelle Werk ersetzt werden[48]. Mit ergänzten Normdaten bzw. Nomisma-IDs für z. B. die Nominale könnten sich andere Münzkomplexe, wie beispielsweise von KENOM, besser mit den FMRD-Daten vergleichen lassen.

Bibliographie

Hubers 2013
T. Hubers, Poppler for Windows (2013); URL: https://blog.alivate.com.au/poppler-windows/ (Aufgerufen am 14.03.2019)

Konrad 2017
M. Konrad, Data Mining OCR PDFs — Using pdftabextract to liberate tabular data from scanned documents (2017); URL: https://datascience.blog.wzb.eu/2017/02/16/data-mining-ocr-pdfs-using-pdftabextract-to-liberate-tabular-data-from-scanned-documents/ (Aufgerufen am 12.03.2019)

Liu et al. 2008
Y. Liu et al., Identifying Table Boundaries in Digital Documents via Sparse Line Detection, in: Proceedings of the 17th ACM conference on Information and Knowledge Management (New York 2008) 1311–1320; DOI 10.1145/1458082.1458255

Morscheiser 2009
J. Morscheiser, Die Anfänge Triers im Kontext augusteischer Urbanisierungspolitik nördlich der Alpen (Wiesbaden 2009)

Radnoti-Alföldi 2006
M. Radnoti-Alföldi, Die Fundmünzen der römischen Zeit in Deutschland IV 3/2. Stadt und Reg.-Bez. Trier. Die Sog. Römerbauten (Mainz 2006); http://d-nb.info/981457207

Schrade 2016
T. Schrade, Geisteswissenschaftliche Fachdatenrepositorien im Semantic Web, in: DHd 2016. Modellierung, Vernetzung, Visualisierung. Konferenzabstracts (Leipzig 2016) 232–235

[47] Die CSV ist am besten mit LibreOffice Calc und der Pipe | als Trennzeichen zu öffnen.

[48] Weitere Informationen über den Inhalt der Masterarbeit können auf https://tkissing.pages.gitlab.rlp.net/fmrd_ma_website/ abgerufen werden.

Vidya 2014
V. L. Vidya, A Survey of Web Data Extraction Techniques, International Journal of Advance Research in Computer Science and Management Studies II,9, 2014, 76–79

Weber 2010
W. Weber, Dom und Liebfrauenkirche, in: K.-P. Goethert – W. Weber (Hrsg.), Römerbauten in Trier: Porta Nigra, Amphitheater, Barbarathermen, Thermen am Viehmarkt, Kaiserthermen, Basilika, Dom und Liebfrauenkirche, Römerbrücke. Führungsheft Burgen, Schlösser, Altertümer Rheinland-Pfalz 20 ²(Regensburg 2010) 181–199; http://d-nb.info/1002839785

Wigg Wolf – Tolle – Kissinger 2019
D. Wigg-Wolf – K. Tolle – T. Kissinger, Nomisma.org: Numismatik und das Semantic Web, in: DHd 2019. Digital Humanities: multimedial & multimodal. Konferenzabstracts (Frankfurt am Main) 2019, 188–192; DOI 10.5281/zenodo.2596095

Yadav et al. 2018
M. Yadav et al., Result extraction from searchable PDF, International Journal of Advance Research, Ideas and Innovations in Technology IV,2, 2018, 2021–2025