Richtlinie zum „Software-Sprint“

SemanticDocuments – Semantische Informationen aus PDF-Dokumenten

Schlussbericht

Zuwendungsempfänger:

Michael Volz

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS24S14 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Immens viele Abläufe in unserer Gesellschaft basieren auf dem Austausch und der Bearbeitung von Dokumenten – ein Großteil davon weiterhin in Papierform. Selbst bei digitalen Dokumenten fehlen häufig strukturierte Informationen und die Einhaltung von Standards für Barrierefreiheit. In Folge wird behinderten Menschen hier ein essentieller gesellschaftlicher Zugang verwehrt und bei Beteiligten entsteht hoher Mehraufwand, weil Aufgaben bewältigt werden müssen, die unter anderen Umständen automatisierbar wären (z.B. Klassifizieren und Einsortieren von Dokumenten, Identifizieren von Metadaten, Extraktion tabellarischer Daten, Filtern anhand von Metadaten wie Vorgangsnummer, Sachbearbeiter*in).

Ziel des Projekts war es diesen Missstand anzugehen und sowohl Individuen als auch Organisationen zu helfen, Abläufe zu verbessern und Automatisierungspotenzial zu nutzen. Besonderes Augenmerk lag dabei auf Personen, die auf Assistenztechnologien wie Screenreader angewiesen sind. Es sollte eine Software entwickelt werden, um Informationen aus digitalen und eingescannten Dokumente nicht nur "rudimentär", sondern tatsächlich barrierefrei (und semantisiert) zugänglich zu machen. Herkömmliche und bisher genutzte OCR-Programme ermöglichen dies nicht.

Dieses Ziel sollte durch die Kombination existierender Technologien und Open-Source-Software erreicht werden. So war geplant, existierende Machine-Learning-Modelle für die Analyse der Seitenstruktur von Dokumenten oder zur Klassifizierung von Dokumenten zu verwenden. Für die Erkennung einzelner Elemente wie Tabellen und Listen sollte existierende Open-Source-Software zum Einsatz kommen. Ein Hauptziel des Projektes war die Integration der zahlreichen vorhandenen Open-Source-Werkzeuge in eine Komplettlösung zur semantischen Auszeichnung von Dokumenten.

Die Software sollte verschiedene Ausgabe-Formate wie barrierefreie PDF/UA, json, scholary HTML, markdown und org-mode unterstützen.

Die wichtigsten Meilensteine waren:

1. Evaluation der Bibliotheken und Tools, auf die SemanticDocuments aufbaut
2. Gegebenenfalls Anpassung von einzelnen Komponenten
3. Sammlung/Erstellung von Trainings- und Testdatensätzen
4. Erstellung der Software-Pipeline für die Erkennung und Rekonstruktion von strukturierten Daten und Fließtext
5. User Testing (Klassifizierung von Dokumentenart, Layout, Metainformationen, Priorisierung und Hierarchisierung von Informationen)
6. Implementierung des Ausgabemoduls zur Erstellung von PDF/UA-Dokumenten mit eingebetteten Metadaten
7. User Testing: Güte der Barrierefreiheit
8. In Abhängigkeit des Projektverlaufs: Integration in vorhandene Projekte: z.B. Nextcloud Flow, paperless-ngx, OpenPaperwork, ContentMine
9. In Abhängigkeit des Projektverlaufs: Tool zur Korrektur der erkannten Daten und zum Nachtrainieren der eingesetzten Machine-Learning-Modelle

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Auf die Zielsetzung und den Fokus des Projekts wurde oben bereits eingegangen. Zusammengefasst möchte SemanticDocuments das gesamtgesellschaftliche Problem nicht-semantisierter Dokumente und deren Verarbeitung adressieren, für alle Menschen nutzbar sein, einen besonderen Fokus auf Screen-Reader-Nutzer:innen legen und so der Allgemeinheit von Nutzen sein.

In einer Gesellschaft, in der wir täglich Dokumente austauschen und bearbeiten, ist es notwendig, dass diese auch von allen benutzt werden können. Unsere Software hilft einerseits bei der Erstellung von barrierearmen Dokumenten, vor allem in Kontexten, in denen die Verbesserung der bisherigen Lösungen nicht schnell genug möglich ist. Zudem hilft sie Betroffenen bei der Verarbeitung von Dokumenten, die sie in barrierebehafteter Form erhalten haben. SemanticDocuments will so einen wichtigen Beitrag zur gesellschaftlichen Teilhabe leisten, die ohne Barrierefreiheit nicht realisierbar ist.

Die Ergebnisse des Projekts stehen offen und kostenfrei zur Verfügung. Damit soll sichergestellt werden, dass sie, völlig abseits eines profitorientierten Angebots, möglichst vielen Menschen zu Gute kommen.

Das Projekt richtet sich sowohl an Personen und Organisationen im beruflichen, institutionellen und privaten Kontext. All jene, die daran interessiert sind, Dokumentenverarbeitung zugänglicher und/oder effizienter zu gestalten sind Teil der Zielgruppe. Ein besonderer Fokus liegt dabei auf Menschen mit Seheinschränkungen oder dem Bedarf nach Komplexitätsreduktion, da hier die Verfügbarkeit quasi roher, und gleichzeitig semantischer "getaggter" Daten den Zugang überhaupt erst ermöglicht. Anwendungsfälle, bei denen das Projekt hilft, sind beispielsweise:

- Dokumente in Papierform oder nicht-zugänglicher digitaler Form aufbereiten und zugänglich machen

- Dokumentenablage nach inhaltlichen Kriterien
- Indizieren und durchsuchbar machen von Dokumenten, sowohl nach Volltext als auch nach Metainformationen wie Datum (in verschiedenen Formaten), Rechnungsnummer, Ansprechperson
- Einbinden in Automatisierungsprozesse
- Erzeugen von barrierefreie PDF/UA-Dokumenten bzw. Ergänzung bestehender Dokumente um die logische Struktur
- Extrahieren von Teilen des Dokuments, Reduktion auf relevante Informationen

Ausführliche Darstellung der Ergebnisse

Ich hatte während der Projektphase Unterstützung von weiteren Personen, die auf konzeptueller Ebene und beim user testing mitgewirkt haben. Unser Hauptergebnis ist die von uns veröffentlichte Software "semdoc". Diese ist ein Kommandozeilen-Tool, das sich leicht in Benutzer:innen-Anwendungen integrieren lässt und Eingabedokumente mit fehlenden Barrierefreiheits-Funktionen analysiert und als barriereärmere PDF/UA-Dokumente wieder ausgibt. Es ist in Python geschrieben und steht unter der GPLv3-Lizenz.

Die Software besteht aus drei großen Komponenten-Gruppen: Eingabe-Komponenten, Verarbeitungs-Komponenten und Ausgabe-Komponenten. Die Eingabe-Komponenten erlauben das Einlesen von digitalen und eingescannten PDF-Dokumenten sowie von Bitmap-Dateien.

Die Verarbeitungs-Komponenten führen jeweils einen Schritt auf dem Weg zu einem barrierearmen Dokument durch. Solche Schritte sind beispielsweise die Analyse der Dokumentenstruktur (Identifikation einzelner Elemente als Überschriften, Textabsätze, Grafiken, Tabellen und so weiter), die Erkennung von Text oder das Verstehen von Tabellen. Für viele dieser Aufgaben wurde auf bereits vorhandene Open-Source-Lösungen zurück gegriffen und diese integriert. Für andere haben wir eigene Lösungen geschrieben.

Ein Herzstück der Software stellt die so genannte Erkennungs-Pipeline dar. Sie ist dafür verantwortlich, dass die einzelnen Verarbeitungs-Komponenten korrekt ineinander greifen und einfach zu implementieren sind. Sie ruft die einzelnen Komponenten in der richten Reihenfolge auf, übergibt Zwischen-Ergebnisse an spätere Analyse-Schritte und sorgt dafür, dass die Ergebnisse korrekt zusammengesetzt werden.

Schließlich sind die Ausgabe-Komponenten für das Erzeugen verschiedener Dateiformate verantwortlich – je nach Anforderung an Weiterverarbeitung oder Speicherung der Ergebnisse. Hier ist der meiste Aufwand in die Entwicklung der PDF/UA-Ausgabe geflossen. Diese muss mehrere Schritte auf dem Weg zu einem standardkonformen und barrierearmen Dokument durchführen. So müssen beispielsweise die erkannten Metadaten in das PDF-Dokument geschrieben werden und der logische Aufbau des Dokumentes muss als Baumstruktur von so genannten tags in der PDF-Datei ablegt werden. Auch muss die Komponente die Grafik- und Schrift-Anweisungen, die für die visuelle Repräsentation des Dokumentes verantwortlich sind, verstehen und jede Anweisung mit einem Verweis auf die entsprechende Stelle im Struktur-Baum versehen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die während der Projektphase erstellte Software erzeugt aus unstrukturierten oder eingescannten Dokumenten PDF-Dateien, die dem PDF/UA-Standard entsprechen. Dies ermöglicht Benutzer:innen

von Screenreadern, den Inhalt von Dokumenten deutlich besser zu erfassen. Beispielsweise können sie sich bei Bedarf nur die Überschriften und ihre jeweiligen Hierarchie-Ebenen vorlesen lassen und so einen schnellen Überblick gewinnen. Oder sie können eine Tabelle zellenweise navigieren und sich an beliebigen Punkten die aktuelle Spalten-Überschrift ausgeben lassen.

Das PDF/UA-Format hat über den Gebrauch mit Assistenz-Software hinaus auch in anderen Kontexten Vorteile. So beinhaltet es eine Reihe von Metadaten über das Dokument, die Dokumenten-Verwaltungs-Systeme für die Indizierung, Suche und Ablage von Dokumenten heran ziehen können. Auch entspricht es den Anforderungen an langfristige Dokumenten-Archivierung.

Dadurch, dass das erstellte Programm als Open-Source-Software veröffentlicht wurde, steht es der Zielgruppe und der weiteren Open-Source-Community unbegrenzt zur Verfügung. Sowohl die Weiterentwicklung des Projekts selbst als auch die Anpassung an spezifische Bedürfnisse oder die Verwendung innerhalb eines größeren Systems ist uneingeschränkt möglich. Wir werden das Projekt weiter verbessern und neue Funktionen einbauen. Darüber hinaus sind wir bestrebt, die Benutzer:innen-Basis zu verbreitern und stehen Interessierten offen und hilfsbereit zur Verfügung.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Es gab den Versuch, eine rudimentäre Variante der Layout-Erkennung ohne Machine-Learning-Modell durchzuführen. Dies hätte den Vorteil gehabt, mit deutlich weniger Rechenleistung auszukommen und schneller zu sein. Statt der Modelle kamen Computer-Vision-Algorithmen zum Einsatz. Dies funktionierte bei einfachen Dokumenten mit reinem Text-Inhalt ausreichend gut, aber kam bei mehr Komplexität schnell an die Grenzen. Ich habe diese Funktion dann wieder aus der Software raus genommen und habe mich auf machine-learning-gestütze Verfahren konzentriert. Dennoch haben diese Versuche eine produktive Rolle dabei gespielt, das Grundgerüst der Software zu schreiben und eine geeignete Architektur für die Pipeline zu entwickeln.

Auch die Suche nach einer geeigneten Bibliothek für die PDF-Ausgabe beinhaltete mehrere Versuche, eine bestimmte Lösung zu benutzen, die nicht zum Erfolg führten. Manchmal war erst nach einer etwas näheren Beschäftigung klar, ob die Bibliothek geeignet war. Auch hierbei führten die erfolglosen Versuche aber am Ende zu einer geeigneten Lösung. Ähnliches lässt sich über die Suche nach dem passendsten Machine-Learning-Modell zur Layout-Erkennung sagen, wobei quantitativ der Aufwand hierbei geringer war.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Der Sourcecode des erstellen Tools ist als Open-Source-Software unter der GPLv3-Lizenz veröffentlicht und auf https://codeberg.org/InfiniteAccess/SemanticDocuments frei verfügbar. Informationen zur Software und unseren weiteren Ergebnissen haben wir auf der Webseite https://semdoc.infiniteaccess.eu/ veröffentlicht.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die semantische Analyse des Seiteninhalts benötigte deutlich mehr Aufwand als gedacht. Während der Evaluationsphase wurde klar, dass viele Bibliotheken und Tools, die wir benutzen wollten, für die Zieldokumente keine ausreichend guten Ergebnisse lieferten. Dies hatte zur Folge, dass wir mehr

Software evaluiert haben als geplant und dass wir mit der Integration der ausgewählten Bibliotheken mehr Aufwand hatten.

Auch die Ausgabe als PDF/UA-Dokument benötigte mehr Aufwand als geplant. Dies lag zum einen daran, dass die verfügbaren Hilfsbibliotheken, die uns bekannt waren, entgegen unseren Erwartungen nicht geeignet waren, die erforderlichen Operationen auf PDF-Dokumenten auszuführen. Es kostete Zeit, dies festzustellen und Alternativen zu suchen. Wir haben erwägt, eine entsprechende Bibliothek selbst zu schreiben, fanden dann aber doch noch ein Projekt, das wir benutzen konnten, um einen Teil der damit verbundenen Arbeit zu sparen. Dennoch war der reale Aufwand für diesen Meilenstein mehr als doppelt so hoch wie geschätzt. Da die Implementierung der Ausgabe als barrierefreie PDF hohe Priorität hatte, hatten wir damit frühzeitig begonnen und ausreichend Zeit reserviert, sodass der Meilenstein dennoch erfolgreich abgeschlossen werden konnte. In Folge konnten bei der Erkennungs-Pipeline jedoch nicht alle Integrationen und damit nicht alle Funktionen umgesetzt werden.

Neben diesen inhaltlichen Herausforderungen und Anpassungen gab es an manchen Stellen Verzögerungen im Zeitplan, die durch Ereignisse außerhalb des Projekts aufgetreten sind, wie private Herausforderungen der Teammitglieder. Da wir ein sehr kleines Team waren, ist es uns schwerer gefallen, solche Einflüsse abzufangen, als das bei größeren Teams oft möglich ist. Sie hatten zwar keine Auswirkungen auf den Aufwand der einzelnen Meilensteine oder die Fertigstellung des Projekts als ganzes, aber auf interne Terminierungen von Teilaufgaben, Zielformulierungen und Motivation. Insgesamt konnte dennoch allen Widrigkeiten gut begegnet werden. Dies lag auch an einer guten und flexiblen Zeitplanung des Teams und wurde durch den Rahmen der Förderung und die Begleitung durch die Open Knowledge Foundation sehr gut unterstützt.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Nicht zuletzt durch die bevorstehende In-Kraft-Tretung des Barrierefreiheitsstärkungsgesetzes zum 28. Juni 2025 bekommt die Erstellung von barrierefreien PDFs zur Zeit viel Aufmerksamkeit. Während der Projektphase wurden bei Software-Bibliotheken, die dazu benutzt werden, programmatisch PDF-Dokumente zu erstellen, Bestrebungen mit dem Ziel das PDF/UA-Format zu unterstützen voran getrieben. Dies ist zunächst sehr begrüßenswert, weil die besten Ergebnisse in Bezug auf Barrierefreiheit erzielt werden können, wenn von Anfang an daran gedacht wird und die verwendeten Werkzeuge dafür Unterstützung mitliefern. Auf unser Projekt gab es einerseits keine direkten Auswirkungen, da diese Bestrebungen die Erstellung von neuen Dokumenten betreffen und wir uns mit bereits existierenden Dokumenten beschäftigen. Auch auf technischer Ebene konnten wir nicht von solchen Entwicklungen profitieren. Auf der anderen Seite haben diese Entwicklungen uns aber herausgefordert, die Zielsetzung unseres Projektes kritisch zu hinterfragen und zu schärfen. Wir mussten uns immer wieder klar machen, dass es weiter Bedarf dafür gibt, existierende Dokumente zu verbessern – auch wenn glücklicherweise zukünftig immer mehr Dokumente von Anfang an barriereärmer erstellt werden.

Eine andere Entwicklung während der Projektphase war, dass mehrere Machine-Learning-Modelle zur Struktur-Erkennung von Dokumenten veröffentlicht wurden. Diese Technologie stellt einen zentralen Schritt in unserer Erkennungs-Pipeline dar und ist entscheidend für die Qualität der Ausgabe-Dokumente. Deswegen haben wir diesen Entwicklungen einige Aufmerksamkeit gewidmet um sicher zu stellen, dass wir die besten Modelle nutzen. Da die Qualität der Analyse eines Modells unter anderem stark von der Dokumentenart abhängt, mussten die Modelle in verschiedenen

Kontexten evaluiert werden. Dies hat mehr Zeit gebraucht, als ursprünglich gedacht und unseren Zeitplan verschoben. Für die Fähigkeit unserer Software, unterschiedliche Dokumentenklassen zu erkennen, war dies eine positive Entwicklung, aber wir konnten in Folge noch nicht alle Features implementieren, die auf der Roadmap standen.