

Masterarbeit Tagebuch

Sven Burkhardt

Inhaltsverzeichnis

1 Vorlage dd.mm.yyyy

Kurzbeschreibung

Hier kurze Zusammenfassung einfügen.

Erledigte Aufgaben

Aufgabentitel

✓ Aufgabenbeschreibung.

Ergebnis: Ergebnisbeschreibung.

Nächste Schritte

☐ Beschrieb der nächsten Aufgabe.

Offene Fragen

☐ Beschreibung der offenen Frage.

2 Überblick Hilfskript CHatGPT_{API}TranskribusXMLtoJSONv3.py

Dieses Skript verarbeitet XML-Dateien aus einer definierten Ordnerstruktur und sendet deren Inhalt an die OpenAI-API zur automatischen Analyse und Extraktion relevanter Metadaten. Das generierte Ergebnis wird als JSON-Datei gespeichert.

Das Skript ist speziell auf historische Dokumente (z. B. aus dem Männerchor Murg Corpus, 1925-1945) zugeschnitten. Es analysiert und strukturiert die Daten, um relevante Informationen wie Autoren, Empfänger, Orte, Ereignisse und Zeitangaben zu extrahieren.

3 Voraussetzungen Einrichtung

3.1 API Key setzen

Bevor das Skript ausgeführt wird, muss der OpenAI-API-Schlüssel als Umgebungsvariable gesetzt werden. Dies kann dauerhaft in der `/.zshrc`-Datei erfolgen:

Alternativ kann das Skript mit VS Code gestartet werden, indem es aus dem Terminal mit folgendem Befehl aufgerufen wird:

3.2 Python-Bibliotheken installieren

Folgende Bibliotheken werden benötigt:

Falls weitere Module fehlen, kann ein allgemeiner Installationsbefehl helfen:

4 Funktionsweise

Das Skript ist in folgende Hauptbestandteile gegliedert:

4.1 Basis-Einstellungen

- Zähler zur Erfassung der verarbeiteten Dateien und Token-Kosten.
- API-Schlüssel wird aus der Umgebungsvariable geladen.
- Pfad-Definitionen für Input- und Output-Verzeichnisse.

4.2 Verzeichnisstruktur durchlaufen

Das Skript iteriert über alle 7-stelligen Ordner im Basisverzeichnis (`base_input_directory`). In diesen Ordnern sucht es nach Unterverzeichnissen.

Falls kein page-Ordner gefunden wird, wird die Verarbeitung dieser Akte übersprungen.

4.3 Verarbeitung der XML-Dateien

Jede XML-Datei im page-Ordner wird eingelesen und analysiert:

- Seitennummer extrahieren: Die Dateinamen haben das Muster `p001.xml`, `p002.xml` usw.
- XML-Daten einlesen: Nutzung von `xml.etree.ElementTree` zur Extraktion von Transkribus-Metadaten und TextEquiv-Daten.
- Fehlermanagement: Falls eine Datei nicht geparkt werden kann oder keinen verwertbaren Text enthält, wird sie übersprungen.

4.4 Strukturierung der Daten in JSON

Die extrahierten Informationen werden in einem JSON-Format gespeichert. Dazu gehören:

- Metadaten (Dokument-ID, Seiten-ID, Bild- und XML-URL)
- Autor und Empfänger mit Name, Rolle und zugehöriger Organisation
- Erwähnte Personen, Organisationen, Ereignisse und Orte
- Dokumentart (z. B. Brief, Protokoll, Rechnung)
- Dokumentformat (z. B. Handschrift, maschinell, mit Unterschrift, Bild)

4.5 API-Anfrage an OpenAI

Ein Prompt wird erstellt, um die Inhalte durch die OpenAI-API analysieren zu lassen. Dabei wird explizit vorgegeben:

- Die historische Relevanz der Dokumente (1925-1945).
- Die Aufgabenstellung (Identifikation von Dokumenttypen, Metadaten und Inhalten).
- Die genaue Formatierung der JSON-Antwort.

5 Fehlerbehandlung

Das Skript enthält mehrere Mechanismen zur Fehlerbehandlung:

- Fehlermeldungen beim XML-Einlesen (try-except beim Parsen)
- Fehlermeldungen bei API-Anfragen (try-except um den OpenAI-Aufruf)
- Fehlermeldungen bei JSON-Speicherung (try-except beim Schreiben der Datei)
- Logging von Problemen (print(f» Fehler beim Parsen der API-Antwort: e»))

Falls Fehler auftreten, werden sie ausgegeben und das Skript setzt die Verarbeitung der nächsten Datei fort, anstatt komplett abzubrechen.

6 Fazit

Dieses Skript automatisiert die Verarbeitung von XML-Dokumenten, extrahiert deren Inhalte und strukturiert die Daten in JSON-Format, das von OpenAI-API analysiert wird. Die Ergebnisse werden gespeichert und abschließend statistisch ausgewertet. Durch Fehlerbehandlung und Logging wird sichergestellt, dass auch bei Problemen das Skript robust bleibt.

Falls du Fragen hast oder Anpassungen benötigst, kannst du einfach den relevanten Abschnitt im Code ändern!

7 Tagging der JPGEs im AppleFinder 26.10.24

Kurzbeschreibung

Überlegung: JPEGs sollen bereits im Apple Finder mit Tags versehen werden, um eine effiziente, automatisierte Transkription der Chorunterlagen des Männerchors Murg zu ermöglichen. Geplant ist die Kombination von ChatGPT und Transkribus zur Erkennung unterschiedlicher Dokumententypen. Ein Tag-System, bestehend aus „**Maschinell**“ für maschinengeschriebene und „**Handschrift**“ für handschriftliche Dokumente, gewährleistet die gezielte Zuordnung zur jeweils geeigneten OCR-Software (*Maschinenschrift mit ChatGPT, Handschrift mit Transkribus "German Giant"*).

Dokumente, die sowohl maschinell erstellten Text als auch handschriftliche Elemente enthalten, werden entsprechend ihrer Hauptinformationsgehalt getaggt. Zusätzlich erhalten alle Dokumente mit Unterschriften den Tag „**Unterschrift**“, um eine gezielte Verarbeitung dieser Elemente sicherzustellen.

7.1 AppleFinder Tags

- **Handschrift**
- **Maschinell**
- **mitUnterschrift**
- **Bild**

Erledigte Aufgaben

Handschriften tagging

- ✓ taggen ● **Handschrift** in AppleFinder.
- ✓ taggen ● **Maschinell** in AppleFinder.
- ✓ taggen ● **Bild**
- ✓ taggen ● **mit Unterschrift**

Ergebnis: Handschriften, Maschinell, Bilder und alle handschriftlichen Unterschriften getaggt

Nächste Schritte

- ☐ Skripte schreiben, um maschinelle Text zu extrahieren ☐ Transkribus für Handschriftliches anschmeißen.
- ☐ Nach Gemeinsamkeiten in den Texten suchen, um automatisierte Abfrage für ChatGPT zu erstellen.
- ☐ Ggf. Aufteilung in unterschiedliche Korpora (Briefe handschr. Briefe Schreibmaschine, Zeitungsunterlagen.)
- ☐ Transkribus für Handschriften verwenden.

Offene Fragen

- ☐ Sollen die Bilder gelöscht werden?

8 JPG Datenbereinigung - leere Seiten löschen 25.10.2024

JPG Datenbereinigung

Alle JPGs ohne Inhalt, also beispielsweise Rückseiten, werden gelöscht. Regel: sobald etwas handschriftlich oder gedruckt auf einer Seite steht, bleibt es erhalten. Im Moment sind auch Bilder (Bsp. Postkarten) inbegriffen. Bilder mit Taggs

8.1 Anmerkung

Geschichte/Chronik/Gründung des Männerchors in Akte 323

Erledigte Aufgaben

JPG Datenbereinigung

- ✓ Alle JPGs ohne Inhalt, also beispielsweise Rückseiten, werden gelöscht.
- Ergebnis:** Reiner JPG Korpus mit Schriftgut, aber auch Bildern (bspw. Postkarten)

Nächste Schritte

- ☐ Nach Gemeinsamkeiten in den Texten suchen, um automatisierte Abfrage für ChatGPT zu erstellen.
- ☐ Ggf. Aufteilung in unterschiedliche Korpora (Briefe handschr. Briefe Schreibmaschine, Zeitungsunterlagen.)
- ☐ Transkribus für Handschriften verwenden.

Offene Fragen

- ☐ Sollen die Bilder gelöscht werden?
- ☐ Handschriftliche, maschinengeschriebene und gemischte Daten taggen? Ggf. erst später mit ChatGPT.
- ☐ Transkribus für Handschriften verwenden?

9 Datennormalisierung PDF zu JPEG 24.10.2024

Kurzbeschreibung

Heute zwei Python-Skripte zur Normalisierung der Akten geschrieben:

Erledigte Aufgaben

PDF zu JPG Konvertierung

✓ Skript *JPEG-to-PDF.py* geschrieben.

Ergebnis Alle PDF-Seiten in JPGs umgewandelt, Dateinamen mit Seitenzahlen formatiert.

Prüfung der Aktennummern

✓ Skript *Check-if-all-files-complete.py* geschrieben.

Ergebnis: Überprüft, ob Akten von 001 bis 425 vorhanden sind. Alle Akten sind vollständig in JPG umgewandelt.

Nächste Schritte

☐ Daten für OCR-Bereinigung vorbereiten, leere Seiten manuell entfernen.

Offene Fragen

☐ Handschriftliche, maschinengeschriebene und gemischte Daten taggen? Ggf. erst später mit ChatGPT.
