(Semi-)Automatic method of reconstructing the book collection based on source documents

1. Model do rozpoznawania pisma odręcznego

Cel: Automatyczne odczytanie rękopisów (starych inwentarzy) lub działań

Podejście:

- Zbieranie zeskanowanych dokumentów + ręczne transkrypcje jako dane treningowe.
- dla rękopisów potrzeba HTR (np. CRNN, TrOCR, Transkribus).
- Fine-tuning modelu na historycznych czcionkach i rękopisach.

Technologie:

- TrOCR (Transformers OCR, Microsoft) - bardzo dobre dla historycznych tekstów.
- PyTorch + CRNN - działa dobrze dla pisma odręcznego.
- Transkribus - bardziej gotowe narzędzie ale można też trenować własne modele

2. Oczyszczanie i strukturalizacja tekstu

Po przekształceniu rękopisów do postaci cyfrowej za pomocą HTR, otrzymamy surowy tekst, który prawdopodobnie będzie zawierać błędy i nie będzie posiadał odpowiedniej struktury.

Cel:

- Czyszczenie tekstu - usuwanie błędów, artefaktów i niepotrzebnych znaków.
- Strukturalizacja - podział tekstu na kategorie, np. tytuły książek, autorów, daty, miejsca wydania, itp.
- Normalizacja - sprowadzenie danych do jednolitego formatu (np. standardowe zapisy dat, nazwisk).

Podejście:

- Do oczyszczenia tekstu można zastosować reguły heurystyczne i modele do korekcji błędów:
    - RegExp – do usuwania zbędnych znaków
    - Spell-checking np. Pyspellchecker – do poprawienie błędnych słów (tutaj mogą pojawiać się problemy ze słowami w różnych językach)
- Podział tekstu na logiczne jednostki:
    - Tytuł
    - Autor
    - Data wydania
    - Miejsce wydania
    - Sygnatura katallogowa
    - Dodatkowe opisy

Do tego celu moim zdaniem warto użyć modeli NLP

    - NER (Named Entity Recognition) - do identyfikacji nazw własnych (autorzy, tutuły itp.)
    - Reguły heurystyczne – np. Jeżeli tekst zawiera cztery cyfry obok siebie to może być to rok wydania

- Normalizacja danych - według zadanych norm
    - Daty
        - 1874 -> 1874-01-01 jeżeli nie znamy miesiąca i dnia
        - 02.03.1888 -> 1888-03-02 jeżeli format się nie zgadza
    - Autor
        - J. Słowacki -> Juliusz Słowacki
    - Nazwy miejscowości
        - W-wa -> Warszawa - jeżeli stosowane są skróty

3. Model do klasyfikacji wpisów

Warto się zastanowaić w którym miejscu dokładnie należy wykonać ten krok. Tzn. Czy przed strukturalizacją aby łatwiej było nam rozpoznać tytuł, autora itp, czy po niej, żeby łatwiej było rozpoznać czym dany wpis jest

Cel:

- Odróżnienie właściwych wpisów od innych tekstów w dokumencie (Informacje o książce, przypis czy notatka)

Podejście:

- Klasyfikacja sekwencji tekstowych (czy dany wiersz to pełny opis książki?) - można wykożystać Fine-tuning dla DistilBert lub RoBERTa

4. Model do uzupełnienia brakujących informacji

Cel:

- Automatyczzne wypełnianie brakujących pól na podstawie kontekstu

Podejście:

- Uzupełnienie braków - niektóre wpisy mągą być niekompletne np “J. Słowacki Pan Tadeusz” (nie ma daty wydania)
    - Możemy wyszukiwać brakujące dane w bazach bibliotecznych biorąc pod uwagę to co wiemy o “bibliotekarzu”
    - Możemy używać modeli AI do predykcji w celu usupełniania wpisu

- Sentence-BERT – do wyszukwiania podobnych opisów
- Fine-tuning GTP – do generowania uzupełnień

5. Konwersja do katalogu

C**el**: Zamiana danych na gotowy katalog bibliograficzny

Po**dejści**e:

- Konwersja metadanych do formatu JSON lub XML
- Deduplikacja i standaryzacja
    - RapidFuzz do scalania duplikatów
    - Coś do obsługi wybranego wormatu