Information retrieval:

Vektorový model

Popis projektu:

Cílem projektu je implementace vektorového systému ukládání dat (tj. preprocessing těchto dat a jejich indexování) spolu s možností dotazování z webové aplikace. V našem projektu jsme využili kolekci sta knih z projektu Gutenberg.

Dotaz se obecně může skládat z kolekce termů a jejich vah. Pro komfortní dotazování ze strany uživatele jsme se rozhodli implementovat webové rozhraní takovým způsobem, že uživatel samotné termy a jejich váhy nezadává, ale pouze si vybere jednu z nabízených knih a její obsah je následně použit jako dotaz (tj. po vyhledání jsou ve výsledné kolekci sestupně seřazené knihy dle jejich podobnosti k uživatelem vybrané knize).

Způsob řešení:

Preprocessing:

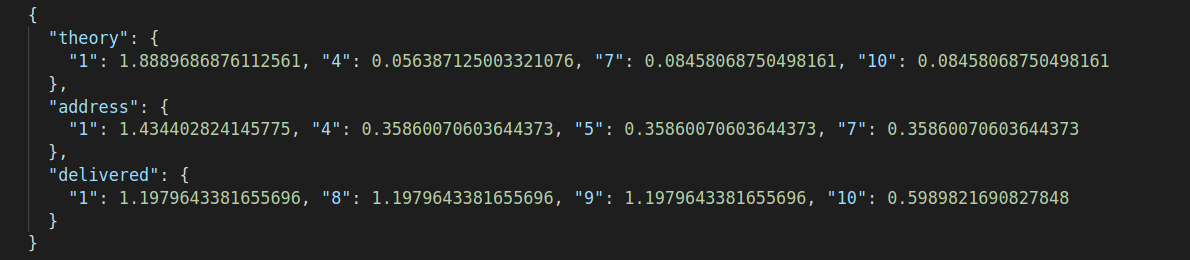
V první fázi se všechny dokumenty musí zpracovat a připravit pro rychlé vyhledávání. To zahrnuje odstranění takzvaných stop words, což jsou slova, která mají malý informační význam. Typicky jsou to třeba předložky, spojky, zájmena atd. Dále jsou všechna slova „lematizována“ tj. jsou převedena do základního tvaru (např. anglická slova speaking a speaks jsou převedena na speak). Poté je již možné spočítat váhu jednotlivých slov v daných dokumentech. To probíhá ve dvou fázích. V první se projdou všechny dokumenty a zjistí se maximální výskyt konkrétního slova v jednom dokumentu. V druhé fázi se spočítá relativní váha slov pro každý dokument podle následujících vzorců:

Díky tomuto vzorci získáme normalizované váhy termů – v intervalu (0; 1>. V dalším kroku spočítáme ještě inverzní frekvenci termu ti v kolekci dokumentů.

Následně tyto hodnoty vynásobíme a dostaneme váhu termu, s níž budeme dále pracovat.

Tu uložíme do souboru v následujícím JSON formátu (viz obrázek):

*"název\_termu": {"id\_dokumentu": váha\_termu, "id\_dokumentu": váha\_termu, …}*



Vyhledávání:

Na úvodní stránce webové aplikace si uživatel může vybrat jeden z nabízených dokumentů. Po potvrzení výběru se mu zobrazí stránka, kde je zobrazeno pět nejpodobnějších dokumentů, které jsou nalezeny pomocí kosinové podobnosti:

Implementace:

První fáze preprocessingu je implementována v Pythonu s pomocí několika ověřených knihoven. Těmi jsou:

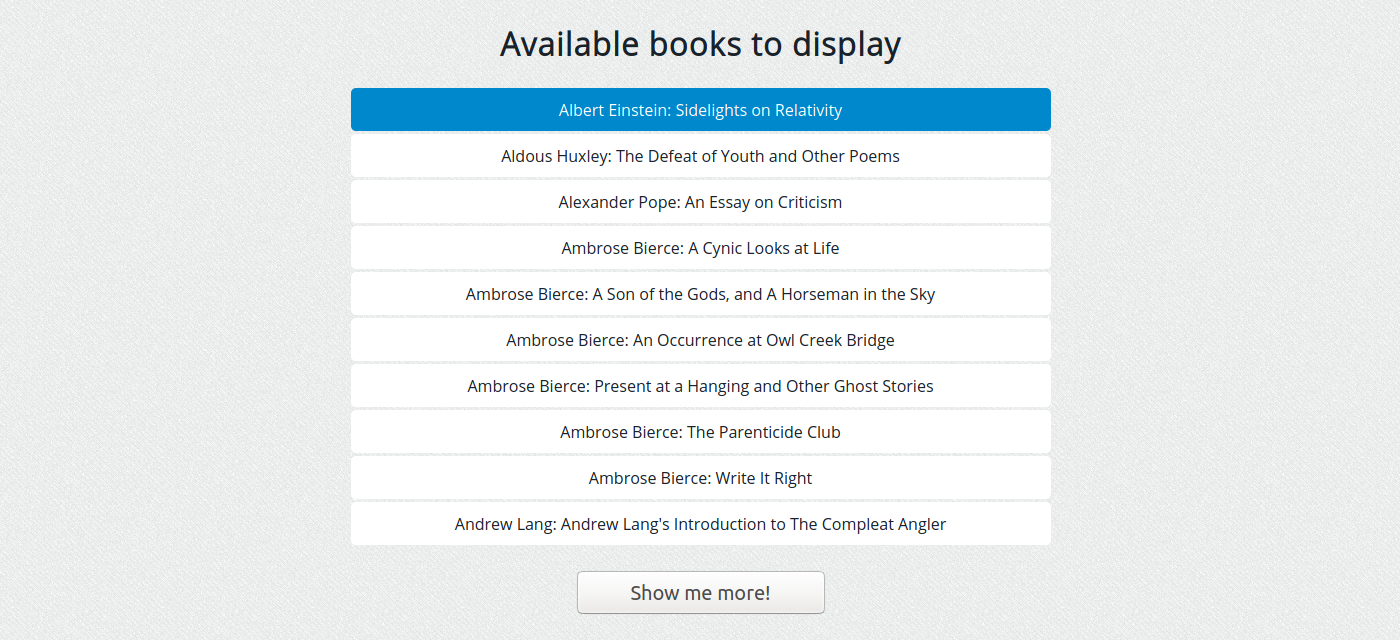
* [NLTK](https://www.nltk.org/) (Natural Language Toolkit), což je knihovna pro práci s lidským jazykem; v našem programu je využita pro odstranění stop words a lematizaci
* SQLite3; jedná se o knihovnu pro persistentní uložení dat; je součástí standartní Python knihovny
* Další knihovny?

Samotný výpočet vah jednotlivých termů a celý zbytek aplikace je napsaný v jazyce C++. V něm jsme využili následující knihovny:

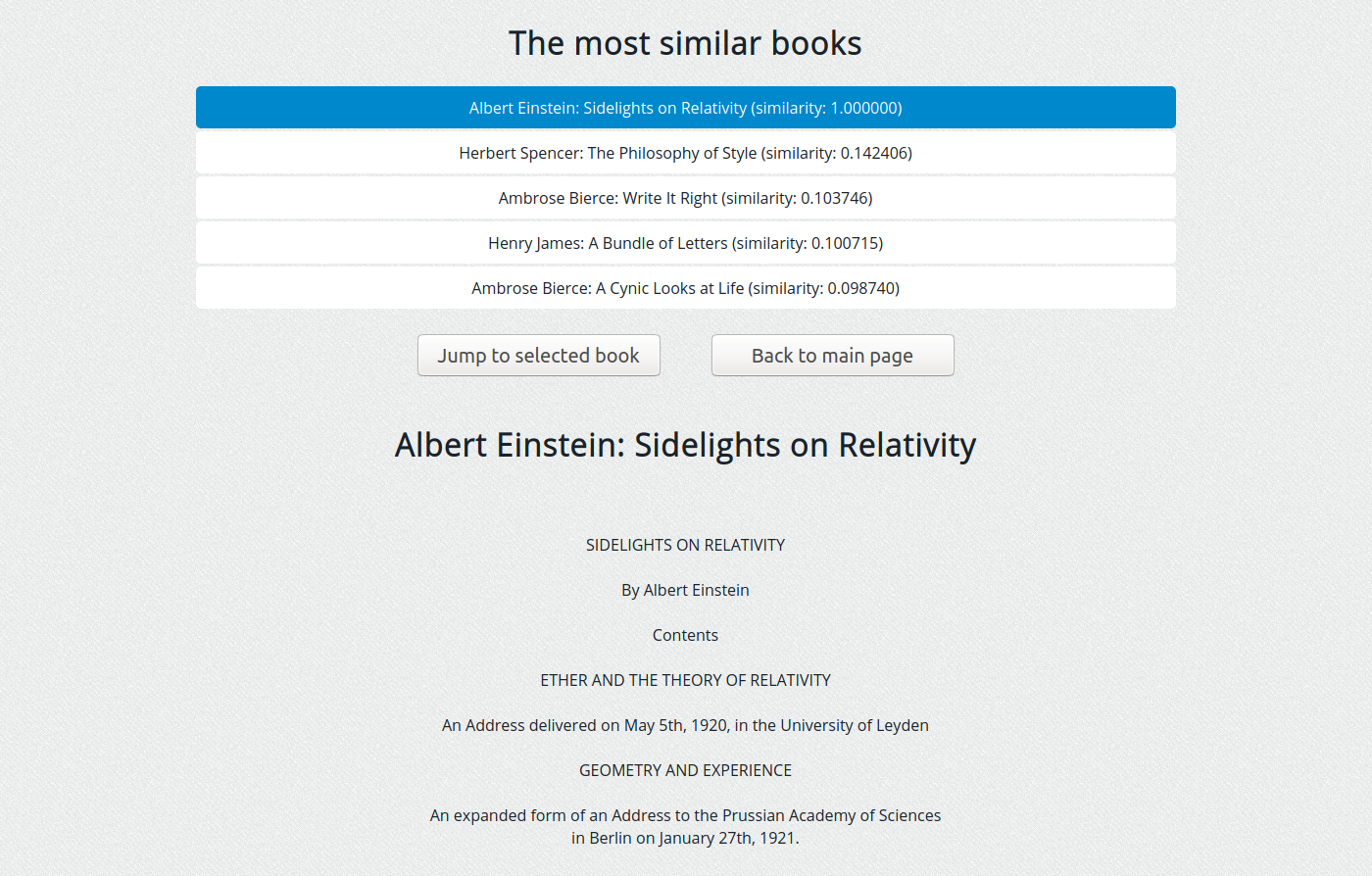
* [SQLiteCpp](https://github.com/SRombauts/SQLiteCpp), což je knihovna pro jednodušší práci s databází SQLite, kterou využíváme pro ukládání dat
* [Wt](https://www.webtoolkit.eu/wt); knihovna pro vytváření webových aplikací
* + další knihovny, dopsat popis

Příklad výstupu:

Na obrázku níže vidíme úvodní obrazovku celé webové aplikace. Zde je možné vybrat jeden z deseti nabízených dokumentů a následně potvrdit svůj výběr kliknutím na tlačítko „Show me more!“.



Na dalším obrázku již vidíme výsledek akce z úvodní obrazovky. V horní části stránky je zobrazeno pět nejpodobnějších dokumentů vybranému na úvodní stránce. Nechybí zde ani informace, jak jsou si dokumenty podobné. Je umožněno na kterýkoliv z nich přejít, případně se vrátit zpět na úvodní stránku. V druhé části stránky můžeme vidět samotný obsah vybraného dokumentu.



Experimentální sekce:

Diskuze:

Závěr: