Vektorový model vyhledávání dokumentů

Daniil Drozdov
ČVUT–FIT
drozddan@cvut.cz

24. května 2024

1 Úvod

Tento report věnován semestrálnímu projektu předmětu BI–PYT B232. Cílem projektu je implementovaní aplikace, která umožní vyhledávání v kolekci dokumentů s využitím vektorového modelu.

2 Způsob řešení

Projekt zahrnuje:

1. **Tokenizace textu**: rozdělení textu na slova;
2. **Odstranění stop slov**: filtrace běžných slov, které nejsou užitečné pro analýzu;
3. **Stemming**: redukce slov na jejich základní tvar;
4. **Výpočet TF-IDF**: transformace textu na TF-IDF skóre, které hodnotí význam slova v dokumentu ve srovnání s jeho frekvencí v korpusu;
5. **Kosinová podobnost**: výpočet podobnosti mezi dvěma vektory v prostoru TF-IDF.

3 Implementace

Architektura aplikace:

- **Zpracování a uložení TF-IDF:** Aplikace je navržena jako skript, který se spouští z příkazové řádky. Nezahrnuje uživatelské rozhraní, ale spoléhá na databázi SQLite pro ukládání a zpracování textových dat.
- **Vyhledávání a zobrazení filmů:** Je to desktopová aplikace s grafickým uživatelským rozhraním, která umožňuje uživatelům interaktivně prohlížet a vyhledávat filmy, a zobrazovat podobné filmy na základě kosinové podobnosti.

Použité knihovny:

- **SQLite3:** Pro manipulaci s databází a ukládání dat.
- **NLTK:** Pro tokenizaci, odstranění stop slov a stemming. Tokenizace rozděluje text na slova, odstranění stop slov filtruje běžná slova, která nejsou užitečná pro analýzu, a stemming redukuje slova na jejich základní tvar.
- **scikit-learn:** Pro výpočet TF-IDF, který hodnotí význam slova v dokumentu ve srovnání s jeho frekvencí v korpusu, a pro výpočet kosinové podobnosti mezi vektory.
- **Numpy:** Pro matematické operace, zejména při výpočtu norm a kosinové podobnosti.
- **Tkinter a customTkinter:** Pro vytvoření a správu grafického uživatelského rozhraní.
- **Pillow (PIL):** Pro práci s obrázky v GUI.

Požadavky na spuštění aplikace:

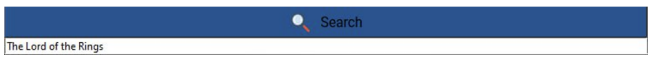
- **Python:** Kód vyžaduje Python 3.x.
- **Externí knihovny:** Jak bylo zmíněno, aplikace vyžaduje instalaci několika externích knihoven, které mohou být nainstalovány pomocí pip (např. `pip install numpy nltk scikit-learn pillow customtkinter`).
- **Databáze:** Pro spuštění je nutné mít databázi SQLite s předpřipravenou strukturou tabulek a potřebnými daty.

4 Příklady vstupu a výstupu

V aplikaci lze interagovat dvěma základními způsoby:

1. **Vyhledávání filmu pomocí SQL dotazu:** Uživatel zadá název filmu do vyhledávacího pole a aplikace zobrazí seznam filmů odpovídajících hledání, včetně názvu, žánru a data vydání.
2. **Výběr filmu a zobrazení podobných filmů:** Kliknutím na film v seznamu se zobrazí detailní popis a aplikace vypočítá seznam podobných filmů využívající kosinovou míru podob-

nosti na základě TF-IDF vektorů, což umožňuje uživatelům prozkoumat doporučené filmy s podobným obsahem.



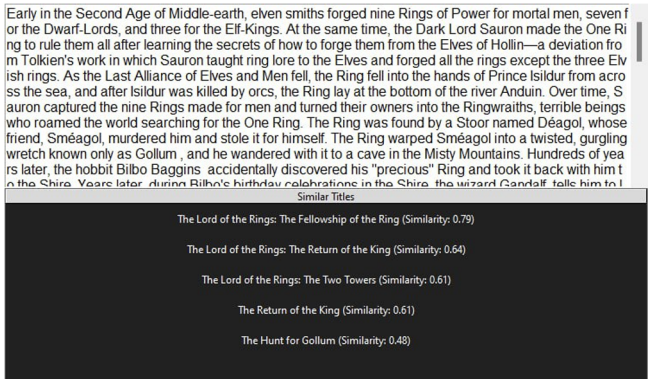
Obrázek 1: Vyhledávání filmu

Title	Genre	Release Date
The Lord of the Rings: The Two Towers	Fantasy Adventure, Adventure, Epic, Action/Adver	2002-12-05
The Lord of the Rings	Fantasy Adventure, Sword and sorcery films, Anim	1978-11-15
The Lord of the Rings: The Fellowship of the Ring	Fantasy Adventure, Adventure, Epic, Fantasy, Film	2001-12-10
The Lord of the Rings: The Return of the King	Fantasy Adventure, Adventure, Epic, Action/Adver	2003-12-17

Obrázek 2: Vystup seznamu filmů

Title	Genre	Release Date
The Lord of the Rings: The Two Towers	Fantasy Adventure, Adventure, Epic, Action/Adver	2002-12-05
The Lord of the Rings	Fantasy Adventure, Sword and sorcery films, Anim	1978-11-15
The Lord of the Rings: The Fellowship of the Ring	Fantasy Adventure, Adventure, Epic, Fantasy, Film	2001-12-10
The Lord of the Rings: The Return of the King	Fantasy Adventure, Adventure, Epic, Action/Adver	2003-12-17

Obrázek 3: Vyběr filmu ze seznamu



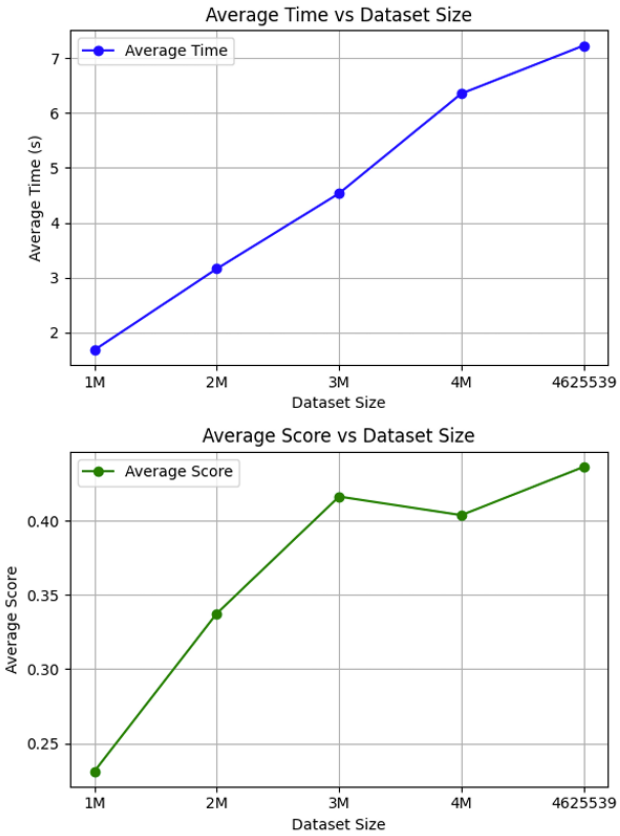
Obrázek 4: Zobrazení popisu filmu a seznamu podobných filmů

5 Experimentální sekce

Cílem experimentu bylo zjistit, jak velikost datasetu TF-IDF hodnot ovlivňuje výkon a přesnost v aplikaci pro vyhledávání podobných filmů.

Při pohledu na výsledné grafy můžeme vidět, jak se s rostoucí velikostí datasetu TF-IDF hodnot zvyšuje průměrná doba vyhledávání podobných filmů. To je očekávané, protože s větším množstvím dat roste výpočetní náročnost. Zajímavým zjištěním je, že průměrné skóre podobnosti filmů má tendenci růst s velikostí datasetu, což může naznačovat, že větší datové sady mohou poskytnout bohatší kontext pro nalezení relevantnějších doporučení.

Výjimkou je mírný pokles průměrného skóre při přechodu z 3 milionů na 4 miliony TF-IDF hodnot, což by mohlo být důsledkem náhodného výběru filmů pro testování.



Obrázek 5: Vysledky experimentů

6 Diskuze

Hlavním cílem bylo vytvoření funkčního prototypu, který umožňuje uživatelům prohlížet filmy a získávat doporučení na základě kosinové podobnosti jejich TF-IDF vektorů. Nicméně, během vývoje a testování aplikace bylo zjištěno, že existují určité nedostatky, které by mohly být vylepšeny:

- **Optimalizace paměti:** Vzhledem k tomu, že aplikace provádí operace s vysokými nároky na paměť, jako jsou výpočty TF-IDF a kosinové podobnosti pro velké množství dat, může být vhodné zvážit použití efektivnějších struktur dat nebo technik, které by snížily paměťové nároky.
- **Škálovatelnost a výkon databáze:** Databáze SQLite je výborná pro menší projekty a rychlý vývoj, ale může se stát úzkým místem při škálování na velké množství dat. Pro zvýšení škálovatelnosti by mohlo být vhodné přejít na robustnější systém řízení databází,

jako je PostgreSQL, které lépe zvládají větší objemy dat.

7 Závěr

V rámci semestrálního projektu jsem se zaměřil na vývoj aplikace, která umožňuje uživatelům prohlížet rozsáhlou databázi filmů a získávat doporučení založená na analýze kosinové podobnosti TF-IDF vektorů.

Projekt byl úspěšný v tom, že jsem dokázal vytvořit funkční prototyp, který splňuje základní požadavky specifikace. Uživatelé mohou efektivně vyhledávat filmy a zobrazit si informace o filmech a jejich podobnosti. Během testování aplikace jsem však narazil na několik problémů týkajících se výkonu při zpracování většího množství dat a škálovatelnosti databáze, což mě omezovalo, zejména při snaze o zpracování dotazů v reálném čase.

Přestože má projekt určité nedostatky, jsem s výsledkem celkově spokojen. Získal jsem cenné praktické zkušenosti v oblasti databázových systémů a zpracování textů, které budou pro mě užitečné v budoucích projektech.

Reference

[1] Codemy.com. Color and style our treeview - python tkinter gui tutorial 118. online. [cit. 2024–04–10] `https://www.youtube.com/watch?v=ewxT3ZEGKAA`.

[2] Dwayne Richard Hipp. Sqlite documentation. online. [cit. 2024–04–10] `https://www.sqlite.org/docs.html`.

[3] Scikit learn Development Team. Scikit-learn: Machine learning in python. online. [cit. 2024–04–07] `https://scikit-learn.org/stable/`.

[4] Edward Loper and Ewan Klein. Nltk :: Natural language toolkit. online. [cit. 2024–04–08] `https://www.nltk.org/`.

[5] NumPy Development Team. Numpy documentation. online. [cit. 2024–04–04] `https://numpy.org/doc/stable/`.