FACULTY
OF APPLIED SCIENCES
UNIVERSITY
OF WEST BOHEMIA

DEPARTMENT OF
CYBERNETICS

Semestrální práce: Vektorový model pro vyhledávání informací

Pavel Ircing

ircing@kky.zcu.cz

Základní zadání – I.

Vytvořte v jazyce Python program, který implementuje vektorový model s tf-idf vahami pro vyhledávání informací.

Systém bude pracovat s testovací kolekcí, kterou je k dispozici na v Google Classroom (SZPJ_SP1_collection.zip)

Tato kolekce obsahuje:

- 3204 dokumentů v adresáři **`documents`**
  - jde o "hlavičky" článků z časopisu Communications of the ACM ze 70. let – časopis se v té době zabýval hlavně různými počítačovými algoritmy
  - dokumenty obsahují název článku, často jména autorů a někdy i abstrakt a klíčová slova – rozhodně se ale nelze spolehnout na úplně pevnou strukturu
- 30 „vývojových" témat pro vyhledávání v souboru *query_devel.xml*
  - začátek každého tématu je uvozen tagem `<DOC>`, konec tagem `</DOC>`. Identifikátor tématu je mezi tagy `<DOCNO>` a `</DOCNO>`
- soubor s posouzením relevance (relevance judgments) **`cacm_devel.rel`** obsahující výčet dokumentů, které jsou pro dané téma relevantní – např. řádek:

```
4 Q0 CACM-1749 1
```

značí, že dokument `CACM-1749` je relevantní pro téma 4 – druhý a čtvrtý sloupec můžete bez obav ignorovat

Základní zadání – II.

Výstup programu bude soubor, ve kterém je pro každý zpracovávaný dotaz vygenerován seznam 100 dokumentů, které jsou dle algoritmu nejpodobnější danému dotazu (tj. nejrelevantnější). Seznam je seřazen sestupně dle skóre podobnosti – např:

```
1       CACM-1938       0.24284285661
1       CACM-2319       0.230932347805
1       CACM-1657       0.227669418393
1       CACM-2371       0.19772801487
…
1       CACM-1647       0.0912735727289
1       CACM-0866       0.0905080935677
2       CACM-3078       0.0930908542064
2       CACM-2434       0.084884674854
…
```

- čili formát „`ID` tématu `<tabelátor> ID dokumentu <tabelátor> sk`óre podobnosti"

(vzorový soubor **`vzor_vystupu_vyhledavaciho_programu.txt`**)

Máte k dispozici také skript **`compute_score.py`**, který na základě porovnání tohoto výstupu se souborem **`cacm_devel.rel`** automaticky vyhodnotí střední průměrnou přesnost (Mean Average Precision – *MAP*)

Detaily implementace

- návrh metod pro předzpracování dokumentu a dotazu je zcela na vás – použijte, co uznáte za vhodné

- konkrétní varianta implementace vektorového modelu je také volitelná a to jak z hlediska toho, jaké knihovny Pythonu použijete, tak i konkrétních formulí pro výpočet *tf* a *idf*

- soubor dotazů, který máte k dispozici, slouží pro vyhodnocení úspěšnosti jednotlivých postupů a výběru těch nejvhodnějších. Aby se však odhalilo případné přílišné „naladění“ na konkrétní soubor dotazů, reálná úspěšnost se většinou zjišťuje pomocí další sady dotazů, která není při vývoji systému k dispozici – tzv. **evaluačních datech**. Tak to uděláme i v našem případě – úspěšnost na evaluačních datech vyhodnotím s použitím vašich systémů sám.
  - prosím berte na vědomí, že čísla témat se budou lišit od *development* dat – pozor na správné parsování souboru s dotazy !

Odevzdání semestrální práce

Jako výsledek své práce odevzdejte:

- program spolu s návodem, jak jej použít pro vyhledávání – kolekci dokumentů předpokládejte neměnnou, měnit se mohou pouze zadávané dotazy. I u těch však bude samozřejmě zachován formát souboru.
- krátkou dokumentaci, která kromě výše zmíněného návodu bude obsahovat i popis použitých metod předzpracování dat, vyzkoušených variant vektorového modelu, případně dalších zajímavostí.
- program bude považován za úspěšně fungující, pokud jeho výstup pro **vývojová** (devel) data – tj. ta, která máte k dispozici – dosáhne hodnoty MAP alespoň **0.3**

Termín odevzdání a bodování:

- termín odevzdání: **7.4.2024**
- maximální možný počet bodů: **10 + „bonus za umístění“:**
  - všechny v termínu odevzdané programy budou seřazeny podle hodnoty MAP dosažené na **evaluačních** datech (tj. těch, která nemáte k dispozici)
  - autor nejlepšího systému získá jako bonus navíc **9** bodů, autor druhého nejlepšího **8.5**, atd. Čili v případě, že práci *v termínu* odevzdají všichni, kdo předmět prokazatelně studují, získá autor systému na posledním místě 0.5 bonusového bodu.

- penalizace za pozdní odevzdání či odevzdání nesprávně fungující práce: **2** body