CMP Rádce
Instalační příručka klasifikátoru

August 24, 2021

Abstract

Tento dokument slouží jako instalační příručka a manuál klasifikátoru pro CMP Rádce. Obsahuje pokyny pro nasazení webové služby včetně inicializace prázdné databáze. Dále popisuje REST API rozhraní webové služby a způsob importu počátečních dat pro trénink klasifikátoru.

1 Nasazení klasifikátoru na serveru

V této části jsou popsány kroky nutné pro úspěšné nasazení klasifikátoru. Postup je testován na systému Ubuntu 20.04, ale měl by být dostatečně generický. Předpokladem je instalace Java JRE ve verzi min. 8, raději však 11 (testováno na OpenJDK 11 64bit) a MySQL (testováno na verzi 8).

1.1 Příprava databáze

V podsložce */data/db_scripts* existuje soubor *ddl.sql*, který postačí spustit nad prázdnou instancí databáze. Tento skript vytvoří nové schéma *cmpradce*.

Co skript neobsahuje je **tvorba nového uživatele** pro tuto databázi - tento krok je nutné zajistit manuálně. Uživatel by měl mít práva pro běžné CRUD operace s daty.

1.2 Spuštění webové služby

Webová služba je dodána jako standalone aplikace obsahující i samotný webový server. Pro úspěšné spuštění je potřeba pouze nainstalované JRE. Spuštění probíhá následovným příkazem:

```
java -jar cmpradce.jar
```

Ve výchozím stavu bez argumentů server nenastartuje. K tomut potřebuje přiložený konfigurační soubor *application.properties* na stejné úrovni, jako samotný JAR soubor webové služby. S přiloženým konfiguračním souborem server startuje na portu 5555 a pokouší se připojit na testovací databázi (*127.0.0.1/cmpradce* s uživ. jménem *cmpradce* a heslem *cmpradce*), která s největší pravděpodobností

nebude na cílovém systému existovat (a zejména kombinace uživ. jména a hesla ani není takto doporučována). Proto je potřeba tyto argumenty (port a argumenty pro připojení k databázi) nastavit úpravou konfiguračního souboru.

Pro úspěšné nastartování webové služby je potřeba běžící a připravená databáze - viz krok 1.1.

2 Počáteční trénink klasifikátoru

Pokud máte vlastní tréninková data, je možné tento krok přeskočit. V opačném případě doporučujeme po prvním spuštění provést prvotní trénink klasifikátoru s využitím dat získaných z časové osy a dobového stavu diskuzního fóra. K tomu lze použít data z podložky */data/train* předané formou cesty na disku jako argument pro REST endpoint */importData* (více v části 4.1). Klasifikátor lze následně "dotrénovat" pomocí inkrementálních synchronizací.

3 Vlastní Stop slova

Klasifikátor ve výchozím stavu pracuje se zabudovaným (v praxi dosti omezeným) seznamem stop slov. Externě lze dodat seznam nový skrze konfigurační soubor *application.properties* a klíč `stopwords.path` mající jako hodnotu cestu k souboru obsahující stop slova (jedno na řádek) - např. takto:

```
stopwords.path=stopwords.lst
```

Jako vzor lze použít dodaný *stopwords.lst*. Že jsou stop slova z externího souboru úspěšně načtena si lze všimnout z logu serveru během procesu učení (hledejte *StopWordsList*).

3.1 Změna stop slov po natrénování

V praxi může nastat případ, kdy po natrénování zjistíme, že nalezená klíčová slova jednotlivých kategorií obsahují také stop slova, která nebyla dříve na seznamu. Tento nežádoucí stav lze vyřešit rozšířením seznamu stop slov a kompletním přetrénováním klasifikátoru.

Upozornění: berte prosím v potaz, že původní tréninková sada se může lišit od té, nad kterou byl aktuální klasifikátor reálně natrénován, pokud již byla provedena synchronizace dat (dotrénování novými daty). Kompletní sadu tréninkových dokumentů je tedy potřeba před přetrénováním s novými stop slovy aktualizovat. Nejsnáze toho lze dosáhnout exportem dokumentů z databáze (obsahuje všechny dokumenty, které kdy byly použity pro trénink klasifikátoru). K tomuto existuje REST endpoint popsaný v části 4.7. Postup je tedy následující:

1. Export všech dokumentů skrze endpoint popsaný v 4.7.
2. Zastavení webové služby.

3. Spuštění skriptu *truncate_all_tables.sql* z podsložky */data/db_scripts* nad databází klasifikátoru (příp. je možné množinu vyprázdněných tabulek upravit s respektem ke vzájemným vazbám).
4. Úprava souboru se stop slovy.
5. Spuštění webové služby.
6. Natrénování klasifikátoru dle 4.1, kde cesta k trénovací sadě bude shodná s cestou, kam byly výše data vyexportována.

4 Popis REST API

V této části popisujeme detailněji jednotlivé REST endpointy a jejich argumenty.

4.1 Počáteční import dat: `/rest/importData/{loadKeyWords}`

Slouží pro import dat z dané cesty na disku. Data jsou očekávána v následující struktuře:

- Kategorie 1
  - Textový dokument 1
  - Textový dokument 2
- Kategorie 2
  - Textový dokument 3
  - Textový dokument 4
  - Textový dokument 5

Ukázku dat lze najít v podložce */data/train*.

4.1.1 Parametry

- Metoda: POST
- Parametr *loadKeyWords*: říká, zda mají být při importu detekována také klíčová slova kategorií (doporučeno TRUE).
- Konzumuje: Plain text (cesta na disku, ze které data importovat).

4.1.2 Odpověď

HTTP 200, pokud import proběhl v pořádku.

4.2 Výpis kategorií: `/rest/categories`

Výpis aktuálně naimportovaných kategorií.

4.2.1 Parametry

- Metoda: GET
- Produkuje JSON

4.2.2 Odpověď

Ukázka:


```
{
   "categories":[
      {
         "name":"4 - Období domácí péče (život po CMP)",
         "id":1
      },
      {
         "name":"1 - Období šoku (první setkání s CMP)",
         "id":2
      },
      {
         "name":"2 - Období akutní péče (neurologické oddělení městské nemocnice)",
         "id":3
      },
      {
         "name":"3 - Období následné péče (rehabilitační ústav)",
         "id":4
      }
   ]
}
```


4.3 Klíčová slova pro jednotlivé kategorie: `/rest/keywords-per-categories`

Výpis aktuálně identifikovaných klíčových slov pro aktuálně naimportované kategorie. Mapování na kategorie lze provést přes atribut *categoryId* korespondující s atributem *id* u výpisu kategorií.

4.3.1 Parametry

- Metoda: GET
- Produkuje JSON

4.3.2 Odpověď

Ukázka:


```
{
   "keyWordsPerCategories":[
      {
         "keyWords":[
            "jí",
            "charity",
            "bytě",
            "maminky",
            "sem"
         ],
         "categoryId":1
      },
      {
         "keyWords":[
            "volali",
            "odváželi",
            "jdu",
            "jedou",
            "nezhroutit"
         ],
         "categoryId":2
      },
      {
         "keyWords":[
            "tohle",
            "vydržet",
            "poradit",
            "neochoty",
            "pevně"
         ],
         "categoryId":3
      },
      {
         "keyWords":[
            "stres",
            "zařizuje",
            "tipy",
            "zabezpečila",
            "jakám"
         ],
         "categoryId":4
      }
   ]
```

```
}
```

4.4 Predikce kategorie: `/rest/categorize`

Provede predikci kategorie (klasifikaci) daného textu.

4.4.1 Parametry

- Metoda: POST
- Konzumuje *application/octet-stream* (text ke klasifikaci)
- Produkuje JSON

4.4.2 Odpověď

Ukázka:

```
{
   "categoryName":"4 - Období domácí péče (život po CMP)",
   "categoryId":1
}
```

4.5 Datum poslední synchronizace: `/rest/lastSync`

Vrací datum poslední synchronizace, nebo 0, pokud žádná ještě neproběhla.

4.5.1 Parametry

- Metoda: GET

4.5.2 Odpověď

"0", pokud ještě v minulosti synchronizace neproběhla, nebo timestamp poslední (úspěšné) synchronizace.

4.6 Synchronizace dat: `/rest/syncData`

Provede synchronizaci dat na základě nových dat dodaných ve formě JSON objektu. Ten obsahuje pole dokumentů (struktura níže) s jejich externím ID (*postId*), textem (*postContent*) a kategorií (ID a název - *categoryId* a *categoryName*). Atributy *postContent* a *categoryName* jsou povinné. ID dokumentu a kategorie mohou chybět (kategorie je následně dohledávána na základě jména - v případě, že je ID kategorie přiloženo, je kontrolována shoda s uloženou hodnotou). V případě, že synchronizátor najde pod ID dokumentu již existující dokument, aktualizuje jeho obsah. V případě, že synchronizátor nenalezne danou kategorii, vytvoří ji. Na závěr dojde k přepočítání klíčových slov pro jednotlivé kategorie. Synchronizace běží asynchronně.

Ukázka vstupního JSON objektu:

```
{
   "objects":[
      {
         "_comment":"aktualizuje dokument s externim ID 999, nebo vytvori novy s timto ID",
         "postId":999,
         "postContent":"Obsah",
         "categoryId":4,
         "categoryName":"3 - Období následné péče (rehabilitační ústav)"
      },
      {
         "_comment":"vlozi novy dokument a necha DB vygenerovat vlastni ID",
         "postContent":"Obsah",
         "categoryId":4,
         "categoryName":"3 - Období následné péče (rehabilitační ústav)"
      },
      {
         "_comment":"ID kategorie bude dohledavano dle nazvu, nebo vytvori novou",
         "postContent":"Obsah",
         "categoryName":"3 - Období následné péče (rehabilitační ústav)"
      }
   ]
}
```

4.6.1 Parametry

- Metoda POST
- Konzumuje *application/json* (nová data k synchronizaci - viz výše)

4.6.2 Odpověď

Vrací vždy HTTP 200, pokud nebyl synchronizátor uzamčen. Jinak HTTP 503 při uzamčení předchozím pokusem (tzn. stále běží poslední synchronizace, příp. zhavarovala - detaily v logu serveru a je nutný restart služby).

4.7 Export dat: `/rest/export`

Provede export všech kategorií a jejich dokumentů z databáze na definované místo ve struktuře vhodné pro opětovný import. Cílové místo, kam budou data exportována, určuje zaslaný JSON objekt, resp. atribut *destination*. Webová služba smaže veškerý již příp. existující obsah v daném umístění!

Ukázka vstupního JSON objektu:

```
{
   "destination":"/tmp/cmpradcetrain"
}
```

4.7.1 Parametry

- Metoda POST
- Konzumuje *application/json* (JSON objekt s destinací - viz výše)

4.7.2 Odpověď

Vrací vždy HTTP 200, pokud nebyl exportér uzamčen. Jinak HTTP 503 při uzamčení předchozím pokusem (tzn. stále běží poslední export, příp. zhavaroval - detaily v logu serveru a je nutný restart služby).

A Tréninková data z CMP Rádce - úprava tříd

Pokud bude potřeba upravit třídy v dodané tréninkové sadě, lze k tomu využít následující postup:

1. V podsložce *data* existuje soubor *cmp-radce.cz.xlsx* - jde o export původních dat z CMP Rádce a k nim přiřazené kategorie, které postačí upravit v Excelu.
2. V podsložce *data* existuje Python 3 skript *cmp-radce.cz-export-xlsx.py*, který vyexportuje upravený XLSX soubor *cmp-radce.cz.xlsx* do podadresáře *MODIFIED*. Sem je následně možné nasměrovat klasifikátor pro natrénování (viz sekce 4.1).