VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ANALÝZA POSTOJŮ ČESKÝCH A SLOVENSKÝCH UŽIVATELŮ NA ZÁKLADĚ DAT ZE SOCIÁLNÍCH SÍTÍ A WEBOVÝCH DISKUSÍ

ANALYSIS OF THE ATTITUDES OF CZECH AND SLOVAK USERS BASED ON DATA FROM SOCIAL NETWORKS AND WEB DISCUSSIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **DUŠAN SLÚKA**
AUTHOR

VEDOUCÍ PRÁCE **doc. RNDr. PAVEL SMRŽ, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

155976

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Student:	**Slúka Dušan**
Program:	Informační technologie
Název:	**Analýza postojů českých a slovenských uživatelů na základě dat ze sociálních sítí a webových diskusí**
Kategorie:	Umělá inteligence
Akademický rok:	2023/24

Zadání:

1. Prostudujte rozhraní služby Facebook a dalších sociálních sítí
2. Navrhněte a implementujte systém, který dokáže pravidelně získávat, indexovat a analyzovat stahovaná data
3. Vytvořte systém pro automatickou klasifikaci shromažďovaných dat, analýzu trendů a vizualizaci výsledků
4. Demonstrujte vytvořený systém na vhodně zvolených příkladech.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- dle doporučení vedoucího

Při obhajobě semestrální části projektu je požadováno:

- funkční prototyp řešení

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Smrž Pavel, doc. RNDr., Ph.D.**
Vedoucí ústavu:	Černocký Jan, prof. Dr. Ing.
Datum zadání:	1.11.2023
Termín pro odevzdání:	9.5.2024
Datum schválení:	9.11.2023

Abstrakt

V tejto bakalárskej práci sa rieši problematika extrakcie a analýzy dát získaných zo sociálnych sietí s cieľom porozumenia verejnej mienky k rôznym sociálnym témam. Cieľom je systematické kategorizovanie a interpretovanie obsahov. Problém je vyriešený prostredníctvom platformy pre extrakciu názorov a automatickej klasifikácie dát, čo umožňuje tvorbu tematických podkategórií a triedenia do nich. Výsledkom práce je systém, ktorý analyzuje sociálne siete a poskytuje hlbší náhľad do verejnej mienky o sociálnych témach. Systém umožňuje organizáciám lepšie pochopiť dynamiku online diskurzu. Prínosom tejto práce je poskytnutie nového nástroja pre analýzu sociálnych otázok, ktorý môže slúžiť akademickej sfére aj organizáciám z praxe.

Abstract

This bachelor's thesis deals with the issue of extraction and analysis of data obtained from social networks to understand public opinion on various social topics. The goal is systematic categorization and interpretation of contents. The problem is solved through a platform for opinion extraction and automatic data classification, which allows the creation of thematic subcategories and sorting into them. The result of the work is a system that analyzes social networks and provides deeper insight into public opinion on social topics. The system enables organizations to better understand the dynamics of online discourse. The benefit of this work is the provision of a new tool for the analysis of social issues, which can serve the academic sphere as well as organizations from practice.

Kľúčové slová

Analýza sociálnych tém, Extrahovanie názorov zo sociálnych sietí, Automatická klasifikácia dát, Vytváranie podkategórií tém, Umelá inteligencia, Veľké jazykové modely

Keywords

Analysis of social themes, Extraction of opinions from social networks, Automatic data classification, Creating subcategories of themes, Artificial intelligence, Large language models

Citácia

SLÚKA, Dušan. *Analýza postojů českých a slovenských uživatelů na základě dat ze sociálních sítí a webových diskusí.* Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Pavel Smrž, Ph.D.

Analýza postojů českých a slovenských uživatelů na základě dat ze sociálních sítí a webových diskusí

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením doc. RNDr. Pavla Smrža, Ph.D. Uviedol som všetky literárne zdroje, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Dušan Slúka
8. mája 2024

Poďakovanie

Ďakujem doc. RNDr. Pavlovi Smržovi, Ph.D. za jeho podporu počas výkonu mojej bakalárskej práce. Oceňujem tiež jeho trpezlivosť a odborné vedenie.

Obsah

Zoznam obrázkov

Kapitola 1

Úvod

V dnešnej dobe je internet zaplavený množstvom dát. Väčšina používateľov upriamuje svoju pozornosť na sociálne siete, kde vyjadrujú svoje názory a postoje na širokú škálu tém. Jednou z hlavných sú sociálne témy, ktoré ovplyvňujú široké množstvo populácie. Preto sociálne siete ako Facebook a Reddit sú vhodným miestom pre nájdenie a analýzu postojov k týmto témam. Na Facebooku sa do konverzácií zapájajú väčšinou staršie ročníky. Čo sa týka Redditu, platforma je priamo stavaná na vyjadrovanie názorov a jej užívatelia sú všeobecne mladší. Ako je zistené v prieskume [4]

Motiváciu k práci som našiel vo využití analýzy. Vie nám priblížiť rozdelenie jednotlivých postojov do kategórií, a tým lepšie interpretovať v čom spočíva záujem. Sociálne témy a ich výsledky či riešenia ovplyvňujú spoločenské, politické aj ekonomické procesy. Je dôležité pre bežných ľudí, ale aj skupiny ľudí ktorých sa témy týkajú, aby si mohli predstaviť v akých aspektoch sa témy popularizujú a čo je obsahom obáv.

Pri malom množstve názorov je možné využiť len manuálnu analýzu. No pre objektívnejší a širší pohľad na tému je vhodné aby existoval systém, ktorý vie diskusie analyzovať s menším množstvom úsilia.

Cieľom tejto práce je navrhnúť a implementovať systém, ktorý využíva metódy umelej inteligencie pre analýzu textových dát získaných zo sociálnych sietí. Sociálne diskusie sa odohrávajú na viacerých miestach v sociálnej sieti. Systém by preto mal byť schopný automaticky vytvárať podkategórie tém a roztrieďovať jednotlivé postoje do týchto kategórií. Dôležité je aby vedel spracovať širokú škálu sociálnych tém a vizuálne reprezentovať zistenia v užitočnej forme.

Práca je štruktúrovaná do niekoľkých hlavných kapitol. Po tomto úvode nasleduje kapitola zaoberajúca sa prehľadom teórie. Tretia kapitola oboznamuje o dátach, ktoré boli v systéme využité. Približujeme ch získavanie, štruktúru a rozdelenie. Štvrtá kapitola popisuje metodológiu a použité technológie pre návrh. Piata kapitola je o implementácii systému.

Kapitola 2

Teoretický prehľad

V tejto kapitole sa nachádzajú teoretické základy a metódy. Tie tvoria náš systém ktorí analyzuje názory. Je určený na efektívne spracovanie a analýzu názorov získaných zo sociálnych sietí. Zameriavame sa na interdisciplinárny prístup, ktorý prepája poznatky z oblasti informatiky, umelej inteligencie a vedy o údajoch. Cieľom je poskytnúť pohľad o teórii použitej pri vývoji systému.

2.1 Hypertext Markup Language

HTML (Hypertext Markup Language) je štandardný značkovací jazyk určený na tvorbu a štruktúrovaného obsahu na internete. Jeho vývoj a špecifikácie sú aktuálne pod kontrolou organizácie WHATWG[1], skratka pre Web Hypertext Application Technology Working Group. Táto organizácia bola založená pracovníkmi najpopulárnejších webových prehliadačov. Z čoho vyplýva že nad organizáciou majú kontrolu spoločnosti ako Google, Mozilla, Apple a Microsoft.

HTML slúži na vytváranie štruktúrovaného obsahu, ktorý sa na internete zobrazuje a môže byť prehliadaču predávaný rôznymi spôsobmi. Či už ide o generovanie serverovou aplikáciou na základe požiadavky užívateľa, alebo klientskou aplikáciou generujúcou HTML za procesu používania. Základným stavebným blokom sú značky, ktoré dávajú špecifický význam a štruktúru. V týchto značkách sa nachádza konkrétna informačná hodnota poskytovaná stránkou[7].

V kontexte našej práce je HTML dôležité z dvoch dôvodov. Po prvé, systém je reprezentovaný ako internetová stránka, pričom HTML je základom vytvárania jej štruktúrovaných obrazoviek. Po druhé, pre efektívne sťahovanie názorov zo sociálnych sietí je kľúčové rozumieť ako sú dáta usporiadané v HTML. Tieto názory a informácie sa často nachádzajú vnorené v značkách. Pre ich extrakciu je nevyhnutné identifikovať a spracovať tieto dáta pomocou unikátnych identifikátorov. Detailnejšie informácie o tom ako tento proces funguje a ako využívať XPath pre navigáciu a výber dát v HTML dokumentoch, vysvetľujem v príslušnej sekcii o XPath 2.4.

Príklad značiek v ktorých sa informácie nachádzajú:

- **<p>**: Základná značka pre prezentáciu textu užívateľovi.

[1] WHATWG

- **<h1>**, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`: Značky predstavujú nadpisy rôznych úrovní, od najdôležitejšieho (`<h1>`) po najmenej dôležitý (`<h6>`), a pomáhajú vytvárať štruktúrovaný obsah na stránke.
- **<ul>**, `<ol>`, `<li>`: Značky pre zoznamy, kde `<ul>` je neusporiadaný zoznam, `<ol>` je usporiadaný zoznam a `<li>` predstavuje jednotlivé položky zoznamu.
- **<table>**, `<tr>`, `<td>`: Značky pre vytváranie tabuliek, kde `<table>` definuje tabuľku, `<tr>` je riadok tabuľky a `<td>` je bunka tabuľky.
- ** **, `<span>`: Obecné kontajnery pre blokové (`<div>`) alebo riadkové (`<span>`) elementy, ktoré sa používajú na obsah a štylizovanie obsahu pomocou CSS2.2.

2.2 Cascading Style Sheets

CSS (Cascading Style Sheets), je jazyk používaný na opis vzhľadu a formátovania dokumentov, napísaných v značkovacom jazyku ako HTML. CSS umožňuje web dizajnérom a vývojárom vytvárať vizuálne atraktívne webové stránky s presným ovládaním rozloženia, farieb, písma a štýlov. S CSS sa dá špecifikovať ako budú webové elementy zobrazené na rôznych zariadeniach s rôznymi veľkosťami obrazoviek. Pomocou selektorov a deklarácií sa definujú pravidlá, čím sa oddelí obsah od dizajnu. Toto je prax odporúčaná v modernom web dizajne [3].

2.3 Laravel Framework

Laravel je moderný PHP framework, ktorý je navrhnutý pre vývoj webových aplikácií s využitím architektúry MVC[2] (Model-View-Controller). Laravel poskytuje vývojárom súbor nástrojov na zjednodušenie bežných úloh ako sú autentifikácia, smerovanie, manažment pripojení a práca s databázou. Jeho funkcionality umožňujú rýchly vývoj aplikácií s dôrazom na udržateľnosť kódu.

Laravel vyniká v integrovaní moderných vývojových praktík a návrhových vzorov. Zahŕňajúc objektovo-relačné mapovanie (ORM)[3] cez Eloquent, využívanie kompozitného softvéru cez Composer[4] a implementáciu závislostí cez služby kontajnerov. Laravel tiež podporuje vývoj softvéru s metodikami ako sú TDD (Test-Driven Development), umožňujúce vytvárať modulárne a štruktúrované aplikácie [1].

Vlastnosti Laravel Frameworku

Poskytuje rozsiahlu sadu funkcií, medzi ktoré patria:

- **Blade Templating Engine:** Intuitívny a flexibilný systém šablón, ktorý umožňuje separáciu logiky aplikácie od jej prezentácie.
- **Migrácie a Seeding:** Nástroje na verzovanie databázy a vkladanie testovacích dát. Zjednodušuje správu a vývoj databáz.

[2] MVC
[3] ORM
[4] Composer

- **Artisan Console:** Vstavaná konzolová aplikácia pre vykonávanie rôznych úloh a automatizáciu vývojových procesov.
- **RESTful Routing:** Expresívne definovanie ciest pre aplikácie, podporujúce REST webové služby.
- **Autentifikácia a Autorizácia:** Integrované riešenie pre správu užívateľských prístupov a oprávnení.

Výhody použitia Laravelu

- **Rýchly vývoj:** Laravel zrýchľuje proces vývoja tým že zjednodušuje bežné programátorské úlohy.
- **Bezpečnosť:** Poskytuje silné mechanizmy ochrany proti bežným bezpečnostným hrozbám ako sú SQL injekcie, XSS a CSRF.
- **Komunita a Zdroje:** Laravel má silnú a aktívnu komunitu, z čoho vyplývajú bohaté zdroje materiálov a podpory.

2.4 Adresovanie informácii na internetovej stránke

XPath (XML Path Language) je jazyk pre navigáciu a výber špecifických častí z XML dokumentov. Môže byť použitý aj pre HTML2.1 v kontexte internetu. Umožňuje výberové dotazovanie dát pomocou presne definovanej cesty k požadovaným elementom. V rámci automatického získavania dát z web stránok umožňuje cielený výber informácií z HTML štruktúry. Čo je obzvlášť užitočné pri extrakcii dát z komplexne štruktúrovaných webových stránok. XPath vyžaduje porozumenie štruktúre dokumentu, aby bolo možné správne definovať cesty k požadovaným dátam [8].

Jednoduchý HTML kód a XPath, ktorý cieli na element:

HTML Kód:

```
<div id="príklad">
    <p>Príklad textu</p>
</div>
```

XPath Výraz:

```
//div[@id='príklad']/p
```

2.5 Aplikačné programovacie rozhranie

API (Application Programming Interfaces) je nástroj umožňujúci komunikáciu medzi rôznymi softvérovými aplikáciami. V súčasnej dobe sú API neoddeliteľnou súčasťou internetu. Poskytujú súbor pravidiel a protokolov pre vývoj a integráciu aplikačného softvéru.

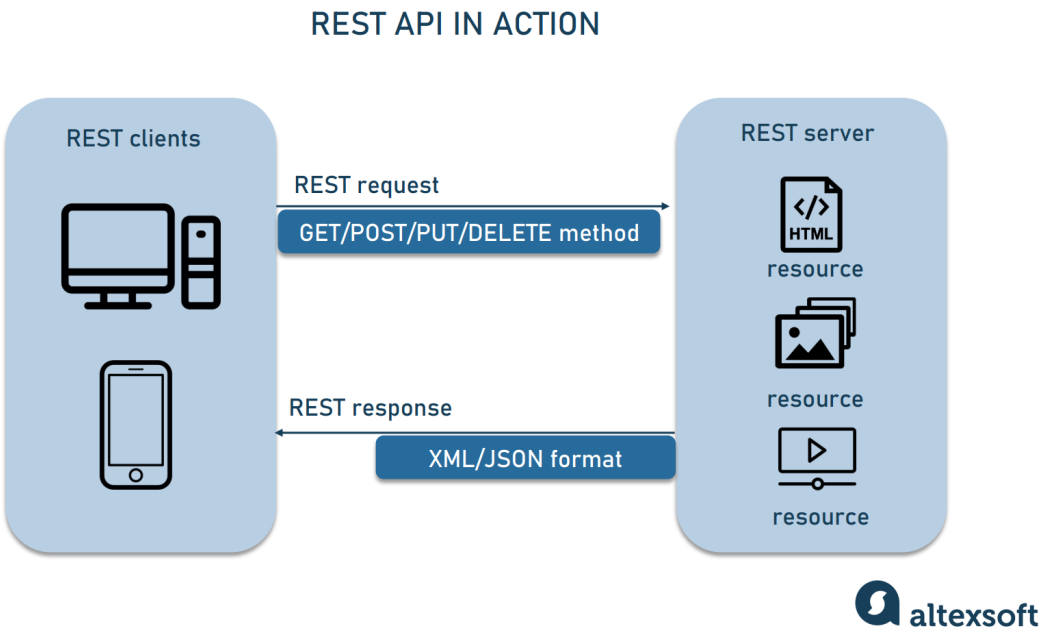
API funguje na báze odosielania požiadaviek medzi rôznymi softvérmi, čo umožňuje výmenu dát a zdieľanie funkcionality. Tento proces je zvyčajne vyvolaný užívateľom v aplikácii, kde potom prevedie API volanie na iný server. Server spracuje požiadavku a odošle

potrebné dáta späť, kde aplikácia vie tieto dáta ďalej spracovať. Využíva sa na rôzne účely, akými sú prístupy k webovým službám, pripojenie k cloudovým službám a integrácie s externými softvérovými platformami. [6]

Existujú rôzne typy API, z ktorých každý je určený pre špecifické použitie:

- **Otvorené API (tiež známe ako verejné API):** sú dostupné vývojárom a ostatným používateľom s minimálnymi obmedzeniami. Umožňujú voľný prístup a poskytujú široký rozsah služieb a dát.
- **Interné API:** používajú sa v rámci spoločnosti na zdieľanie zdrojov a dát medzi interným softvérom a tímami. Zlepšujúc efektivitu a konzistentnosť dát v celej organizácii.
- **Partnerské API:** vyžadujú špecifické práva alebo licencie na prístup. Sú zdieľané externe, ale iba medzi konkrétnymi obchodnými partnermi.
- **Kompozitné API:** kombinujú rôzne dátové a služobné API na vykonanie úlohy alebo poskytnutie služby. Týmto sa umožní vývojárom pristupovať k niekoľkým koncovým bodom v jednom volaní.

Jednou z populárnych architektúr pre návrh sieťových aplikácií je REST (Representational State Transfer). RESTful API využívajú HTTP požiadavky na vykonávanie operácií s dátami modelovanými ako zdroje. Pričom použité sú bežné HTTP metódy, GET, POST, PUT a DELETE. Architektúra REST zdôrazňuje rozšíriteľnosť, bezstavovosť a možnosť ukladania odpovedí do cache. REST API architektúru môžete vidieť na obrázku 2.1.



Obr. 2.1: Obrázok znázorňuje schému práce s REST API. Prevzatý z článku o REST API [2].

2.6 Automatická extrakcia dát z webu

Extrakcia dát z webových stránok je proces získavania a transformácie informácií poskytnutých danou stránkou na štruktúrovanú, čitateľnú alebo logicky zoskupenú formu dát. Proces umožňuje automatické zbieranie dát z rôznych online zdrojov. Sem patria aj sociálne siete, rôzne diskusné fóra, novinárske portály. Dáta získané z procesu sú ďalej analyzované alebo len zobrazované v prívetivejšej forme. Snažíme sa s dátami predviesť užívateľom doposiaľ neznáme informácie nad rámec poskytovaných.

Pri extrakcií dát sa používajú rôzne techniky. Niektoré stránky poskytujú svoje API pripojenia pre priami prístup k im poskytovaným dátam vo formátoch vhodným pre strojové spracovanie. Pokiaľ stránka takýto prístup nepodporuje ďalšou možnosťou je web scraping. Pri scrapingu softvér imituje prehliadanie používateľa a extrahuje z nich údaje. Metóda prináša so sebou aj morálne otázky, rešpektovanie súkromia, ochranu duševného vlastníctva a dodržiavanie pravidiel webových stránok.

Príklady možných aplikácii procesu:

- **Marketing:** Analyzovať online recenzie a sociálne médiá na získanie prehľadu o verejnej mienke ohľadom ich produktov a služieb.
- **Akademický výskum:** Vedci na extrakciu dát pre zhromaždenie veľkých dátových súborov ktoré sú ďalej využívané v štúdiách.

Výzvy a limitácie:

- **Obmedzenia prístupu:** Mnohé webové stránky obmedzujú automatický prístup k svojim dátam pomocou mechanizmov ako CAPTCHA[5]. Vyžadujúc tak pokročilé techniky obchádzania alebo manuálne zásahy do procesu.
- **Zmeny vo formátoch dát:** Stránky pravidelne menia svoju štruktúru a formátovanie. Extrakčné nástroje prestanú správne fungovať bez pravidelných aktualizácií.
- **Právne obmedzenia:** Vrátane autorských práv a ochrany osobných údajov, obmedzujú aké dáta môžu byť extrahované a ako ich používať.

V rámci nášho systému je automatická extrakcia kľúčová. Aby sa jednotlivé témy vedeli vhodne analyzovať, je potrebné množstvo dát ktoré sa nachádzajú na rôznych miestach stránok. Manuálna extrakcia zaberá množstvo času a je neefektívna.

[5] CAPTCHA

2.7 Transformátory

Transformer je prominentný model hlbokého učenia, ktorý bol široko prijatý v rôznych oblastiach, ako je spracovanie prirodzeného jazyka (NLP), počítačové videnie (CV) a spracovanie reči. Transformátor bol pôvodne navrhnutý ako sekvenčný model pre strojový preklad. Neskoršie práce ukazujú že vopred trénované modely založené na transformátoroch môžu dosiahnuť najmodernejšie výkony pri rôznych úlohách. V dôsledku toho sa stal hlavnou architektúrou v NLP. Okrem jazykových aplikácií bol prijatý aj do spracovania zvuku a dokonca aj ďalšie disciplíny, ako je chémia a vedy o živote.

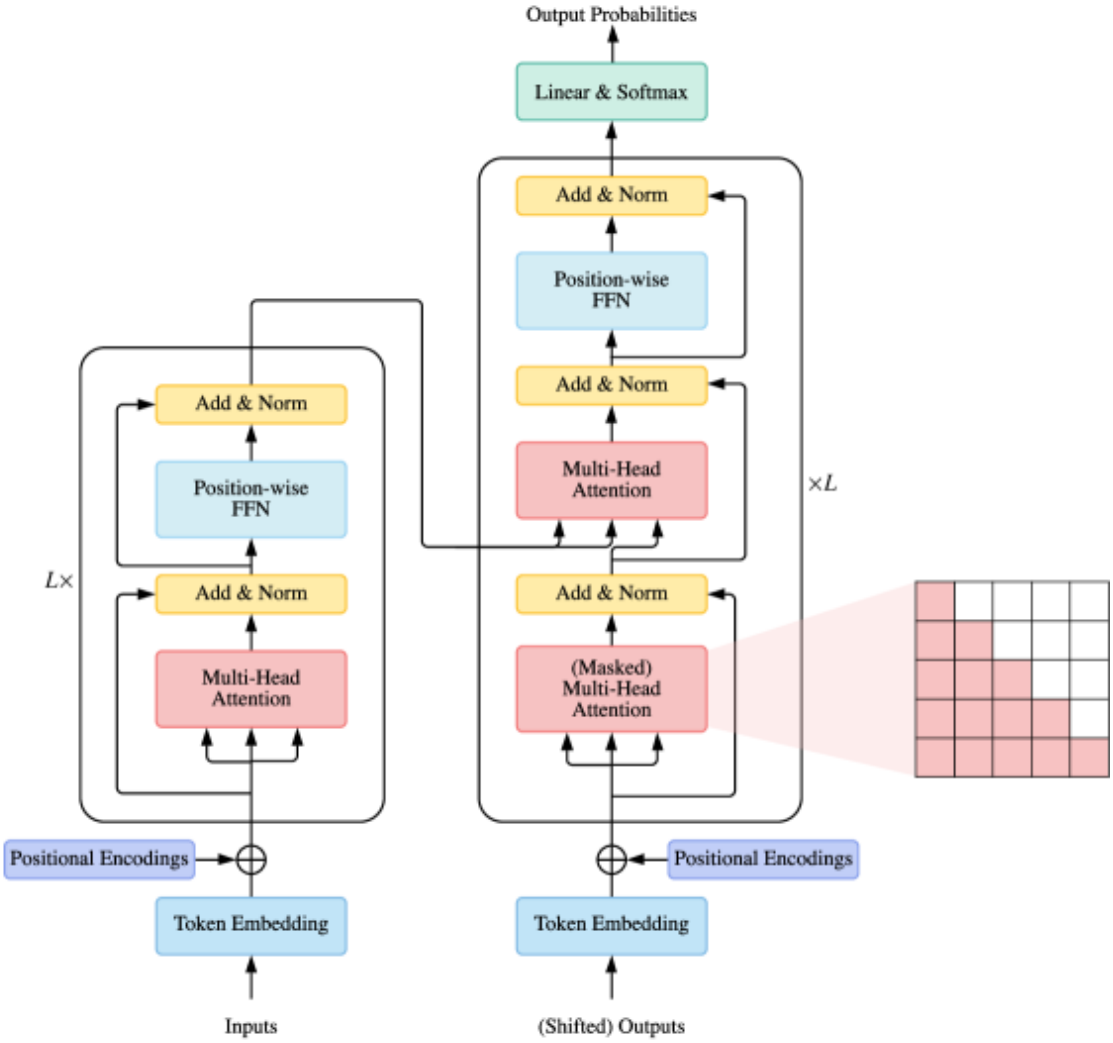
Kvôli úspechu boli v posledných rokoch navrhnuté rôzne varianty transformátorov. Tieto X-formery vylepšujú základný transformátor z rôznych uhlov pohľadu. Informácie prevzaté z [0].

- **Účinnosť modelu.** Kľúčovou výzvou pri použití transformátora je jeho neefektívnosť pri spracovaní dlhých sekvencií, hlavne kvôli výpočtovej a pamäťovej zložitosti modulu. Metódy zlepšovania zahŕňajú odľahčenú pozornosť a metódy rozdeľuj a panuj.
- **Zovšeobecnenie modelu.** Keďže transformátor je flexibilná architektúra a robí málo predpokladov o štrukturálnej odchýlke vstupných údajov, je ťažké trénovať model na údajoch malého rozsahu. Metódy zlepšovania zahŕňajú zavedenie štruktúrneho skreslenia alebo regularizácie, predbežné učenie na rozsiahlych a neoznačených údajoch.
- **Prispôsobenie modelu.** Cieľom tejto línie práce je prispôsobiť transformátor špecifickým následným úlohám a aplikáciám.

Základný transformátor

Základný transformátor, navrhnutý Vaswanim v roku 2017, je model sekvenčného spracovania založený na enkóderoch a dekóderoch[6], ktoré sa skladajú z viacerých identických blokov. Každý blok enkódera obsahuje modul pozornosti a pozíciu špecifickú pre sieť dopredného prenosu. Dekóderové bloky pridávajú moduly krížovej pozornosti a upravujú moduly samopozornosti, aby zabránili prístupu k následným pozíciám. Architektúra vylepšuje hĺbku modelu pomocou reziduálnych spojení a normalizácie vrstiev, čo zvyšuje efektivitu učenia. Prehľad architektúry transformátora 2.2.

[6] enkóder, dekódér



Obr. 2.2: Obrázok ilustruje štruktúru transformátora s enkódovou a dekódovou časťou.

2.8 Jazykové modely

Jazykové modely sú navrhnuté na porozumenie a generovanie ľudského jazyka a textu. Ich schopnosť spočíva v predpovedaní pravdepodobnosti sekvencie slov alebo tvorbe nového textu na základe zadaných údajov. Medzi najbežnejšie typy jazykových modelov patria n-gramové modely. Odhadujú pravdepodobnosť slov na základe predchádzajúceho kontextu. Jazykové modely však čelia viacerým výzvam, ako sú zriedkavé alebo nevídané slová. Tak isto aj problém pre-učenia a obtiažnosť zachytenia zložitých jazykových javov. Preto sa neustále pracuje na vylepšovaní architektúry a metód tréningu modelov. [5].

Veľké jazykové modely

Veľké jazykové modely (LLM), ako GPT-3[7], sú založené na architektúre s vysokou kapacitou učenia. Významným prvkom týchto modelov je mechanizmus self-attention v transformátore. Umožňuje modelom efektívne spracovávať a interpretovať sekvenčné dáta. Táto schopnosť je užitočná pre modelovanie dlhodobých závislostí v texte. Ďalej umožňuje LLM generovať text ktorý je relevantný a koherentný v kontexte.

Model predpovedá nasledujúce slovo y na základe kontextu X, čo matematicky vyjadrujeme ako:

$$P(y|X) = P(y|x_1, x_2, \ldots, x_{t-1})$$

kde $x_1, x_2, \ldots, x_{t-1}$ sú tokeny v kontextovej sekvencii a t je aktuálna pozícia. Model je trénovaný na maximalizáciu pravdepodobnosti celej sekvencie slov. Dosahuje sa to rozkladom podmienenej pravdepodobnosti na súčin pravdepodobností každej z pozícii:

$$P(y|X) = \prod_{t=1}^{T} P(y_t|x_1, x_2, \ldots, x_{t-1})$$

kde T je dĺžka sekvencie. Týmto spôsobom model predpovedá každé slovo v sekvencii autoregresívnym spôsobom a generuje celý text.

Interakcia s LLM často zahŕňa takzvaný "prompt engineering", kde užívatelia poskytujú špecifické texty úloh. Usmerňujú model k generovaniu želaných odpovedí alebo k vykonávaniu konkrétnych úloh. To umožňuje široké využitie LLM v rôznych aplikáciách, od generovania textu až po dialógové systémy, a ako aj v našom systéme pre analýzu názorov.

Zoznam populárnych a používaných veľkých jazykových modelov:

- GPT-3 - (Generative Pre-trained Transformer 3) od OpenAI
- BERT - (Bidirectional Encoder Representations from Transformers) od Google
- T5 - (Text-to-Text Transfer Transformer) od Google
- GPT-4 - ďalšia generácia modelu GPT od OpenAI
- Gemini - séria multimodálnych generatívnych modelov od Google.

Chat-GPT

ChatGPT skratka pre generatívny transformátor pred trénovaný na konverzáciu. Predstavuje rozsiahly jazykový model pre komunikačné roboty ktorý bol vyvinutý spoločnosťou OpenAI[8] a uvedený na trh 30. novembra 2022. Umožňuje používateľom formovať a usmerňovať konverzáciu smerom k požadovanej dĺžke, formátu, štýlu, úrovni detailov a jazyku. Chat GPT je jazykový model na báze umelej inteligencie, ktorý je navrhnutý na vedenie prirodzených jazykových konverzácií s používateľmi. Chat GPT dokáže poskytovať informácie, pomáhať s úlohami a viesť diskusie na širokú škálu tém. Je trénovaný na diverznom datasete, ale jeho znalosti sú aktuálne len do septembra 2021 [10].

[7] GPT-3
[8] OpenAI

Kapitola 3

Dáta

Veľkou Súčasťou nášho systému sú dáta preto si na začiatku predstavíme ich získavanie. Ďalej prejdeme do samotného procesu vytvárania podskupín dát a potom kategorizácie.

3.1 Pôvod dát

V rámci prípravy časti mojej bakalárskej práce som sa zameral na analýzu verejných názorov a diskusií súvisiacich s rôznymi témami. kľúčové frázy ktoré som používal pri hľadaní názorov na diskusných fórach:

- Aký máte názor na interrupcie ?
- Aký máte názor na elektromobily ?
- Aký máte názor na členstvo v Európskej únii ?

Z dôvodu dynamického charakteru tém a potreby zachytiť aktuálne postoje verejnosti som sa rozhodol využiť dáta získané z online diskusných fór a sociálnych sietí. Akými sú najmä Facebook a Reddit. Tieto platformy predstavujú významný zdroj súčasných názorov a diskusií a ponúkajú široké spektrum perspektív od ľudí z rôznych vekových kategórii a sociálnych vrstiev. Zameranie bolo na českých a slovenských užívateľov. Tieto názory preto odrážajú obyvateľstvo týchto krajín. Názory neboli získané z jedného príspevku ani z jednej stránky z dôvodu kvality. Preto sa nemôžem odkazovať na konkrétne diskusné vlákna alebo príspevky na sociálne otázky. Zdroje z ktorých som čerpal:

- **Facebook**
- **Reddit**

Pri nedostatku aktuálnych názorov som sa obrátil na možnosť vytvorenia vlastnej diskusie k témam. Na platforme Reddit, ktorá je stavaná k vyjadrovaniu názorov, som vytvoril nasledujúce príspevky:

- **Členství v Evropské unii a názor Brna ?**
- **Členstvo v Európskej únii?**
- **Členství v Evropské unii ?**

- **Interrupcie**

Texty príspevku boli napísané v štýle aby povzbudzovali užívateľov k vyjadreniu svojich postojov.

3.2 Formát a štruktúra dát

Získané dáta sú vo forme textových príspevkov z diskusných fór a komentárov na sociálnych sieťach. Sú rozdelené do troch textových súborov podľa tém. V každom súbore sa nachádza 150 unikátnych postojov na danú tému. Postoje neboli syntakticky upravované aby odrážali reálne prostredie. Tak isto nebolo prevedené žiadne predspracovanie textu. Pre každú tému sa zvolilo sedem podkategórii. Kategórie jednotlivých tém sú nasledovné:

- **Interrupcie:** Právo výberu, Právo Plodu, Osobná Skúsenosť, Morálka/Etika, Náboženský Pohľad, Zdravotné Dôvody, Ostatné
- **Elektromobily:** Ekonomika, Logistika, Ekologické, Kultúra, Podpora elektromobilov, Nepodpora elektromobilov, Ostatné
- **Členstvo v EU:** Ekonomický prínos EU, Politický vplyv a suverenita, Byrokracia a legislatíva, Migrácia a voľný pohyb, Obavy a kritika EÚ, Pozitívny postoj k EÚ, Ostatné

Kategórie boli zvolené po prečítaní názorov aby sa vytvorilo pokrytie kde viacero z nich pokrývalo jeden komentár. Jednotlivé pod-témy boli priraďované manuálne aby vystihli podstatu názorov. V súboroch sa nachádzajú texty a k nim priradené jedna alebo viacero pod-tém. Štruktúra súborov je nasledovná:

```
[ID číslo názoru v súbore].[Text názoru]
TAGS:[Zoznam pod-tém oddelených čiarkou];

[ID číslo názoru v súbore]...
```

Príklad reálnych dát:

```
1.Bez EÚ by ekonomická nerovnosť bola o dosť väčšia a migrácia
by tiež bola problém. Toto nie sú problémy, ktoré ytvorila existencia EÚ.
TAGS:Ekonomický prínos EU, Migrácia a voľný pohyb, Pozitívny postoj k EÚ;

10.Dajte ludom najskor platy, aby si mohli elektromobil kupit.
TAGS:Ekonomika;
```

Údaje získané z online diskusných fór a sociálnych sietí boli využité nielen na analýzu verejných názorov, ale aj na populáciu databázy. Aby systéme zobrazoval obsah už od jeho spustenia. Tento prístup umožňuje používateľom okamžite pracovať so stránkou a poskytuje reálne príklady diskusií a názorov. Navyše textové príspevky a získané údaje slúžili ako príklady pre dotaz na GPT API, kde vylepšili ďalšie odpovede nástroja.

Kapitola 4

Návrh

V tejto kapitole sa venujeme návrhu aplikácie, použitým postupom a vývojovým metódam. Patrí sem extrakcia dát z diskusií, mazanie a úprava surových názorov, vytváranie pod-tém jednotlivých sociálnych otázok. Tak isto triedenie názorov do podkategórií a vizualizácia analýzy pre užívateľa. Cieľom bolo vytvoriť systém pre analýzu sociálnych otázok, aby sa užívatelia mohli ľahko zorientovať vo verejnom mienení. Výsledky analýz môžu pomôcť pri rozhodnutiach, ako objasniť nejasnosti v otázkach alebo upresniť v ktorých častiach je potrebné verejnosti poskytnúť viac informácií. Začneme s upresnením požiadaviek pre realizáciu systému. Ďalej prejdeme na návrh a schému systému kde popíšeme jednotlivé moduly a ich účel.

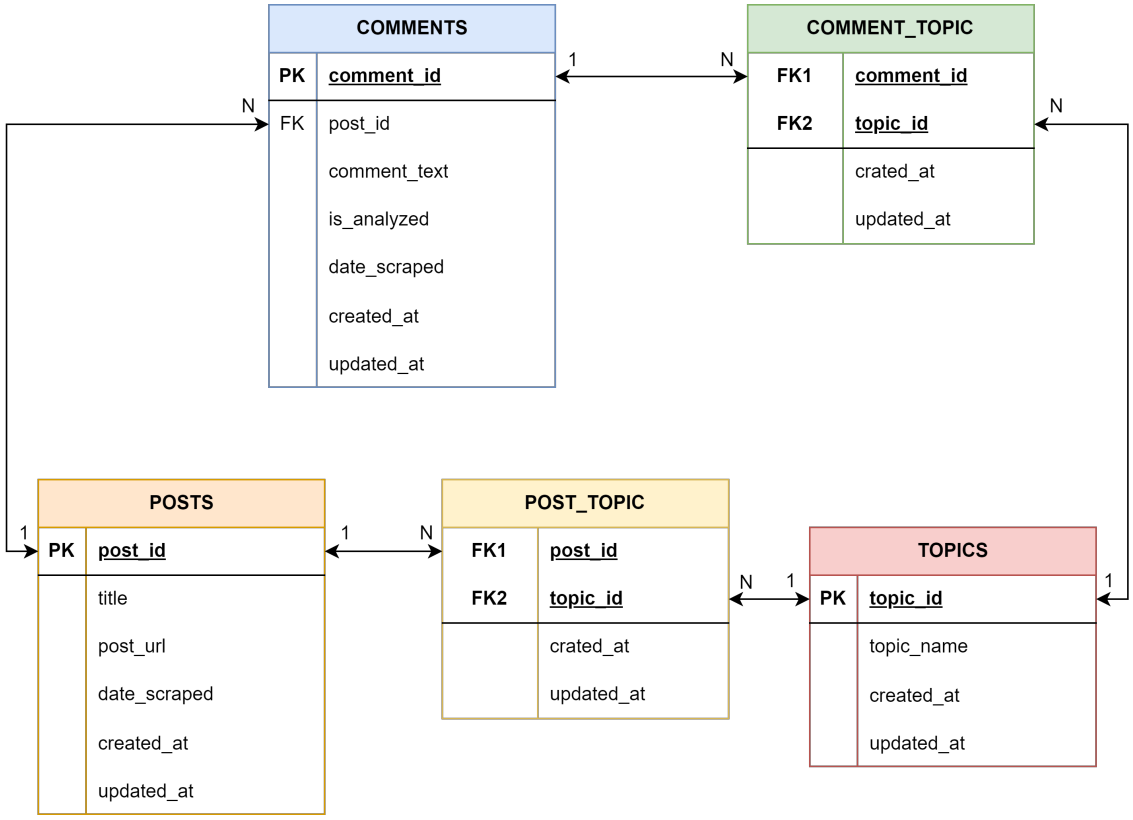
Vzhľadom na charakter a požiadavky tohoto systému sa ukázalo vhodné zvoliť formu webovej aplikácie, ktorú som realizoval s využitím frameworku Laravel. Tento výber umožňuje efektívne spracovanie a vizualizáciu veľkého objemu dát, ako aj jednoduchú a intuitívnu interakciu pre užívateľov prostredníctvom webového rozhrania.

4.1 Analýza požiadavkou systému

Tvoríme systém schopný v reálnom čase analyzovať rozličné sociálne témy, extrahovať relevantné názory z platforiem sociálnych médií a tieto efektívne ukladať. Systém by mal poskytovať intuitívne rozhranie, ktoré umožní užívateľom prácu so získanými dátami. Bude podporovať spracovanie tematicky rozmanitých diskusií skrze dynamicky definované podtémy, do ktorých sa bude automaticky kategorizovať obsah diskusií. Pre zvýšenie prehľadnosti a uľahčenie interpretácie výsledkov analýzy budú informácie vizualizované pomocou grafických elementov. Zabezpečujúc tak užívateľom okamžitý prehľad o vývoji a štruktúre verejnej diskusie na zvolené témy.

4.2 Návrh databázy

Databázový model zohľadňuje vzťahy medzi tromi tabuľkami: POSTS (Sociálne otázky), COMMENTS (Komentáre/názory) a TOPICS (Pod-témy). Ďalšie dve tabuľky POST_TOPIC a COMMENT_TOPIC ktoré riešia vzťah many-to-many. Model poskytuje prehľad o štruktúre databázy a vzťahoch medzi jednotlivými entitami ako je zobrazené na schéme 4.1.



Obr. 4.1: Obrázok znázorňuje schému databázy použitú v systéme.

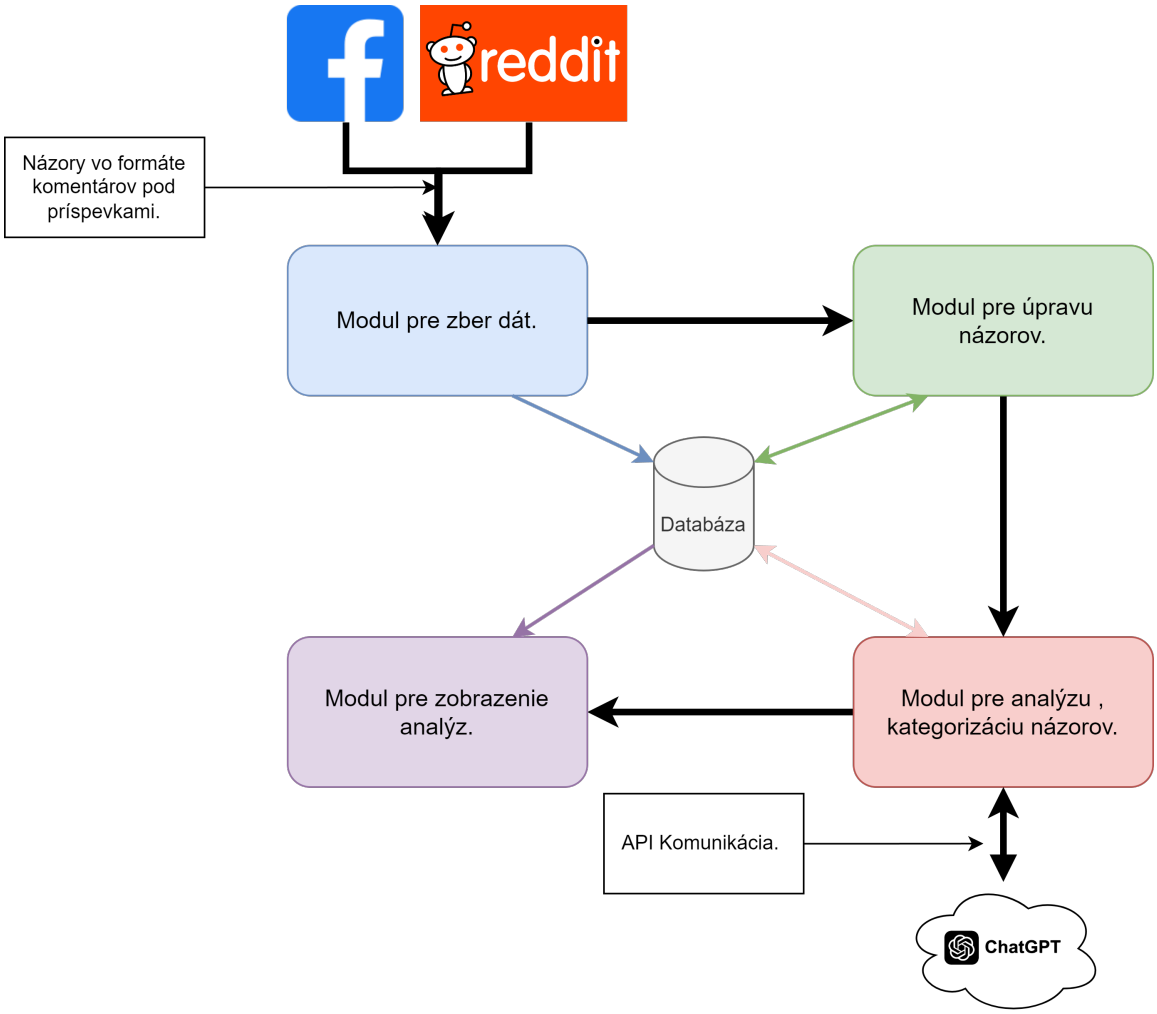
- **Primárne kľúče (PK):** Každá tabuľka má primárny kľúč (post_id, comment_id, topic_id), ktorý jednoznačne identifikuje záznamy v tabuľke. Primárny kľúč je základom pre indexáciu a optimalizáciu dopytov.
- **Cudzie kľúče (FK):** Sú použité na vytvorenie referencií medzi tabuľkami, čo umožňuje databáze zachovať integritu dát a pre naviazanie vzťahov.
- **Texty a metaúdaje:** Stĺpce ako title, comment_text, a topic_name uchovávajú primárne informácie o obsahu záznamu. Metaúdaje ako date_scraped zaznamenávajú časové údaje relevantné pre procesy, ako je sťahovanie dát zo sociálnych sieti.
- **Boolovské stĺpce:** is_analyzed poskytuje rýchly prehľad o stave spracovania komentára, či už bol analyzovaný alebo nie.
- **Timestampy:** created_at a updated_at sú štandardné stĺpce v rámci Laravel Eloquent ORM a sledujú vytvorenie a poslednú úpravu záznamu.

4.3 Návrh systému

Systém je navrhnutý ako súbor vzájomne prepojených modulov, ktoré spoločne komunikujú a zabezpečujú komplexné spracovanie a analýzu sociálnych dát. Celý systém sa delí na štyri hlavné moduly, ktoré reflektujú rôzne aspekty procesu od zberu dát až po ich prezentáciu koncovému užívateľovi. Každý modul je špecializovaný na určitú činnosť a spoločne

tvoria integrovaný systém. Flexibilný, rozšíriteľný a ľahko použiteľný. Využitie MVC architektúry zaručuje, že logika aplikácie (Modely), užívateľské rozhranie (View) a aplikačná logika (Controller) sú dobre oddelené. Toto uľahčuje údržbu a rozširovanie kódu.

Nižšie uvedená schéma poskytuje ilustráciu architektúry a vzájomných vzťahov medzi jednotlivými modulmi systému 4.2:



Obr. 4.2: Schéma znázorňuje architektúru systému.

4.4 Modul pre zber dát

Modul pre zber dát predstavuje základnú komponentu systému. Je zodpovedný za extrakciu užívateľských názorov na špecifické sociálne otázky zo sociálnych sietí akými sú Facebook a Reddit. Užívateľská interakcia je začatá prostredníctvom webového rozhrania, kde užívateľ formuluje dotaz a vyberá preferovanú sociálnu sieť. Následne sú Python skripty, optimalizované pre príslušný typ sociálnej siete. Ďalej aktivované cez Laravel controller. Pre modul boli navrhnuté tieto funkcionality a metódy:

- **Scraping obsahu z Redditu (scrapeReddit)**

  - Metóda umožňuje automatické získavanie obsahu z platformy Reddit podľa užívateľom špecifikovaných parametrov. Samotné sťahovanie dát prevádza python skript ktorí je umiestnený na servery. Spúšťanie prebieha pomocou vytvorenia nového procesu. Návrh samotného skriptu sa nachádza 4.4.

- **Scraping obsahu z Facebooku (scrapeFacebook)**
  - Metóda umožňuje automatické získavanie obsahu z platformy Facebooku podľa užívateľom špecifikovaných parametrov. Samotné sťahovanie dát prevádza python skript ktorí je umiestnený na servery. Návrh samotného skriptu sa nachádza 4.4.

- **Ukladanie príspevkov (storePost)**
  - Funkcia storePost je navrhnutá pre správu a ukladanie užívateľských príspevkov do databázy. Po extrakcii dát zo sociálnej siete sa overia podľa pravidiel databázy. Vytvoria sa korešpondujúce záznamy akými sú nová téma a k nej napojené postoje.

- **Overenie existencie príspevku (checkPostExists)**
  - Metóda na overenie existencie príspevku umožňuje rýchle zistenie, či už bol príspevok s daným názvom v minulosti analyzovaný. Navrhnutý pre vyhľadávanie a minimalizáciu duplikácií v dátach.

Zber dát z Facebooku

Vzhľadom na obmedzenia Facebook API nie je možné priamo vyžiadať komentáre k príspevkom. Python poskytuje knižnice, ktoré sú schopné túto funkcionalitu obchádzať. Využívam knižnicu 'Selenium'[1], ktorá imituje správanie užívateľa a pracuje s webovou stránkou. Skript prijme zadanú sociálnu otázku spolu s prihlasovacím menom a heslom na platformu. Vykoná sa prihlásenie pomocou funkcie *login(email, pas)* a *slow_type(element, text, min_delay, max_delay)*. Pomalé písanie chráni účet pred označením za podozrivý. Pretože rýchle interakcie môžu vzbudiť podozrenie prítomnosti automatizovaného skriptu a nie reálneho užívateľa. Následne sa spustí vyhľadávanie sociálnej témy prostredníctvom vyhľadávača na platforme. Filtre ako napríklad nedávne príspevky sú tiež nastavované pre kvalitnejšie výsledky. Nasleduje cyklus, ktorý prechádza cez určený počet príspevkov a extrahuje z nich komentáre. Počas tohto procesu sa môžu vyskytnúť rôzne problémy, vrátane:

- Problém s hľadanou frázou – nenašli sa žiadne príspevky pre hľadanú frázu alebo ich bolo nájdené len málo.
- Možná prítomnosť reklamy – v príspevkoch sa môže vyskytnúť reklama alebo platený obsah, ktorý s témou nesúvisí.
- Nedostatok komentárov – v príspevku sa nenachádzajú žiadne komentáre alebo obsahuje len obrázky, ako napríklad GIFy, bez textového obsahu na analýzu.
- Nerelevantné komentáre – niektoré komentáre sa môžu týkať úplne inej témy, čo komplikuje analýzu zameranú na špecifickú sociálnu otázku.

[1] Selenium

Zber dát z Redditu

Pre uľahčenie je využitá staršia verzia stránky ktorá má jednoduchšiu HTML štruktúru. Na zber dát z platformy Reddit používam knižnicu 'Scrapy[2]', ktorá je špecializovaná na extrakciu dát z webových stránok. Skript prijme zadanú sociálnu otázku a na základe nej vyhľadáva relevantné príspevky na subredditoch:

- `r/Slovakia`
- `r/Bratislava`
- `r/czech`

Špecifických pre jazykovú lokalitu (SK alebo CZ). Po inicializácii a nastavení jazykovej verzie skript generuje začiatočné požiadavky na URL adresy pre vyhľadávanie na stránkach. Následne sa vykonáva extrakcia názvov, príspevkov a URL odkazov na tieto témy. Skript potom prechádza na stránku každého príspevku, kde extrahuje komentáre. Komentáre sa získavajú pomocou XPath výrazov, ktoré identifikujú textový obsah v rámci HTML elementov 2.4. Tento typ prehľadávania umožní ľahké pridanie alebo zmenu lokalít kde témy hľadať. Taktiež by sa dalo využiť pre vytváranie názorovej mapy regiónov.

Pri návrhu tohoto modulu sa našlo viacero problémov a predstavoval najväčšiu prekážku vo vývoji. Jedným z hlavných problémov je nenájdené témy, keď diskusné vlákna alebo príspevky nemajú jasne definovaný alebo uvedený základný kontext. Sťažujúc identifikáciu relevantných obsahov. Ďalším častým problémom sú irelevantné komentáre, kde užívatelia míňajú tému a diskutujú o nepríbuzných oblastiach, čo zaplavuje dátový set nepotrebnými alebo zavádzajúcimi informáciami. Reklamy a promočné príspevky tiež predstavujú výzvu. Sú často zamieňané za organický obsah, čím dochádza k znečisteniu zberaných dát.

Návrh obrazovky obsluhy pre tento modul 4.3:

[2] Scrapy

Obr. 4.3: Návrh ktorý slúži ako podklad pre vývoj modulu na získavanie názorov.

4.5 Modul pre úpravu názorov

Modul je určený na správu užívateľských názorov, konkrétne pre zobrazenie, úpravu a mazanie komentárov pridružených k príspevkom. Modul je implementovaný s využitím PHP frameworku Laravel a obsahuje funkcionality na serverovej strane (Controller) aj na klientskej strane (View s JavaScriptom). Tento modul je v celej architektúre postrádateľný. K jeho pridaniu ma donútila nespoľahlivosť kvality dát ktoré sa nachádzajú pod príspevkami. Bežne prítomné vulgarizmy bez kontextu neprispievajú k pochopeniu témy takže je na používateľovi či si ich tam chce ponechať alebo nie.

Serverová časť modulu poskytuje nasledujúce funkcionality:

- **Zobrazenie príspevkov (index)**
  - Načíta všetky príspevky a ich príslušné témy pomocou metódy Eager Loading pre optimalizáciu dopytov do databázy.
  - Vráti názory do pohľadu, kde sú zobrazené v rozbaľovacom menu.
- **Získanie komentárov (getComments)**
  - Preberá postId z URL ako parameter.
  - Vypíše ID príspevku v logu a načíta komentáre pre daný príspevok.
  - Vracia samotné komentáre ako JSON odpoveď.
- **Mazanie komentárov (deleteComment)**

  - Získa commentId z URL ako parameter.
  - Skúsi nájsť a vymazať komentár z databázy.

- **Úprava komentárov (updateComment)**
  - Získa commentId ako parameter.
  - Získa nový text komentára z tela požiadavky.
  - Skúsi nájsť komentár v databáze pomocou commentId. Ak komentár existuje, aktualizuje jeho text a uloží zmeny.

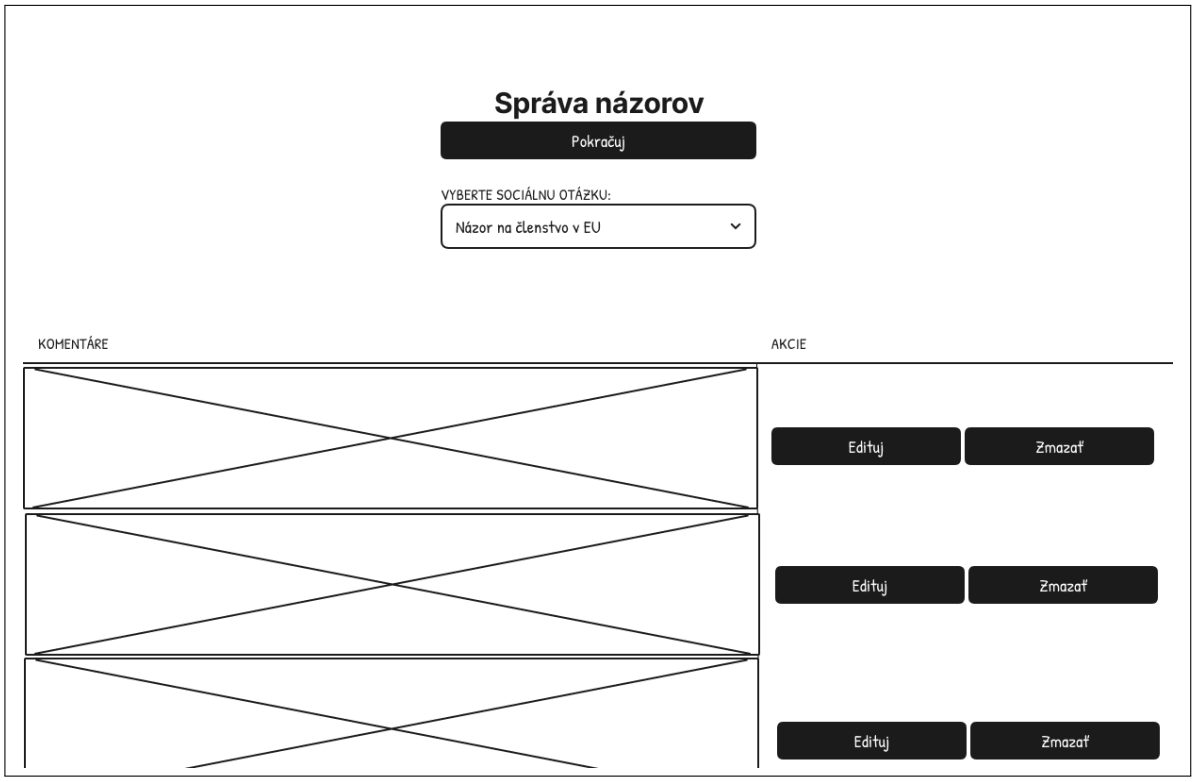
Klientska časť má nasledujúcu štruktúru:

- **HTML a CSS**
  - Štruktúra stránky obsahuje rozbaľovacie menu pre výber príspevku a tabuľku pre zobrazenie komentárov. V tabuľke sa nachádzajú dva stĺpce jeden pre texty komentárov a druhý pre akcie nad týmito dátami.
  - CSS pravidlá určujú štýly pre rôzne časti stránky vrátane minimálnej šírky pre stĺpce tabuľky.

- **JavaScript**
  - Obsluha udalosti zmeny v rozbaľovacom menu pre výber príspevku, načítanie komentárov pomocou AJAX požiadavky na základe vybraného príspevku.
  - Funkcia loadComments dynamicky vkladá komentáre do tabuľky a zobrazuje alebo skrýva kontajner podľa dostupnosti dát.
  - Priradenie event listenerov pre tlačidlá "Zmazať"po načítaní komentárov.

Tento modul spravuje interakcie s užívateľskými názormi na serverovej aj klientovej strane. Využíva Laravel pre backend operácie a AJAX volania pre plynulú interakciu bez potreby obnovenia stránky. Umožňuje užívateľom spravovať obsah príspevkov v reálnom čase a podporuje užívateľskú skúsenosť pomocou dynamického načítavania a interakcií v prehliadači.

Návrh obrazovky obsluhy pre tento modul 4.4:



Obr. 4.4: Návrh ktorý slúži ako podklad pre vývoj modulu na správu názorov.

4.6 Modul pre analýzu a kategorizáciu názorov

Modul analýzy a kategorizácie je hlavnou časťou systému. Jeho úlohou je efektívne spravovať pod-témy jednotlivých sociálnych otázok a následne komentáre týchto otázok roztriediť do jednotlivých podkategórii. Samotná analýza neprebieha na servery. Využíva sa nástroja CHAT GPT ktorý dostane navrhnutú úlohu a vráti odpoveď. Tieto úlohy sú zasielané pomocou CHAT GPT API.

Každá sociálna otázka by mala byť rozdelená do presne siedmich podkategórií. Toto rozdelenie pomáha zachovať organizovanosť dát a zjednodušuje následnú analýzu. Je odporúčané, aby jedna z týchto kategórií bola vždy označená ako Östatné". Táto kategória slúži ako zberné miesto pre všetky komentáre alebo aspekty témy, ktoré sa jednoznačne nezmestia do ostatných špecifikovaných podkategórií.

Funkcionálne požiadavky modulu:

- **Zobrazenie príspevkov:**
  - Modul by mal umožniť užívateľovi zobraziť zoznam všetkých dostupných príspevkov spolu s priradenými témami a počtom neanalyzovaných komentárov. To pomôže užívateľovi vybrať príspevok na analýzu.
- **Manuálne vytváranie pod-tém:**
  - Užívatelia by mali mať možnosť pridávať nové témy k príspevkom podľa svojho uváženia tieto pod-témy uložiť ku konkrétnej sociálnej otázke ktorú chceme analyzovať.

- **Automatické vytváranie pod-tém:)**
  - Poskytovanie automatizovaných návrhov pod-tém založených na obsahu témy. Táto funkcionalita vyžaduje integráciu s externou AI službou, ktorá analyzuje text a navrhuje relevantné pod-témy.
- **Filtrovanie a kategorizácia komentárov**
  - Na základe vybraných tém by modul mal filtrovať a kategorizovať komentáre do pod-tém, čo umožní hlbšiu analýzu názorov spojených s konkrétnymi sociálnymi otázkami.

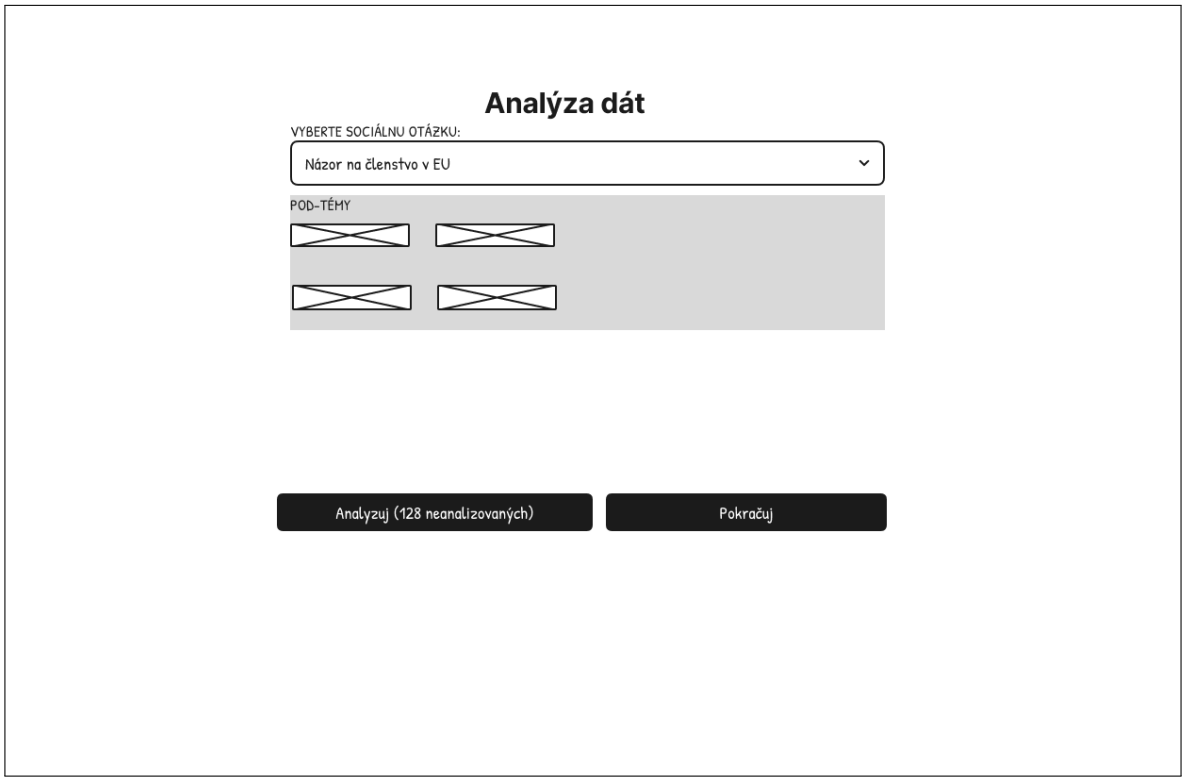
Servis pre komunikáciu s API OpenAI

Trieda OpenAIService je navrhnutá ako servisná komponenta v rámci Laravel aplikácie, ktorá sa zameriava na integráciu s API OpenAI. Služba poskytuje metódy na generovanie pod-tém a priradenie kategórií k jednotlivým komentárom v rámci sociálnych otázok. Kľúčovou časťou je použitie API kľúča z prostredia (env), ktorý zabezpečuje autorizáciu požiadaviek.

Hlavné funkcie triedy zahŕňajú:

- **Generovanie pod-tém (generateSubtopics):** Táto metóda je zodpovedná za komunikáciu s OpenAI API. Používateľ poskytne tému, a systém vygeneruje súvisiace pod-témy. Požiadavka obsahuje prednastavený dialóg, ktorý inštruuje AI model (v tomto prípade gpt-3.5-turbo), aby identifikoval a zoznamoval sedem pod-tém pokrývajúcich rôzne aspekty danej témy.
- **Priraďovanie pod-tém k komentárom (assigneSubtopics):** Funkcia prijíma tri parametre - hlavnú tému, zoznam pod-tém a komentáre. Tieto údaje sú formátované a odoslané do OpenAI služby, kde každý komentár je analyzovaný a kategorizovaní do preddefinovaných pod-tém. Komunikácia s API opäť zahŕňa zabezpečenie požiadavky cez hlavičku s autorizačným tokenom. Celý proces je dokumentovaný v záznamoch.

Návrh obrazovky obsluhy pre tento modul 4.5:



Obr. 4.5: Návrh ktorý slúži ako podklad pre vývoj modulu na analýzu názorov.

4.7 Modul pre vizualizáciu analýz

Modul pre vizualizáciu analýz je navrhnutý tak, aby poskytoval interaktívne grafické zobrazenie dát získaných z analýz sociálnych otázok. Tento modul je integrovaný do Laravel aplikácie a využíva knižnicu 'Chart.js' na tvorbu grafov.

Popis Funkcionalít:

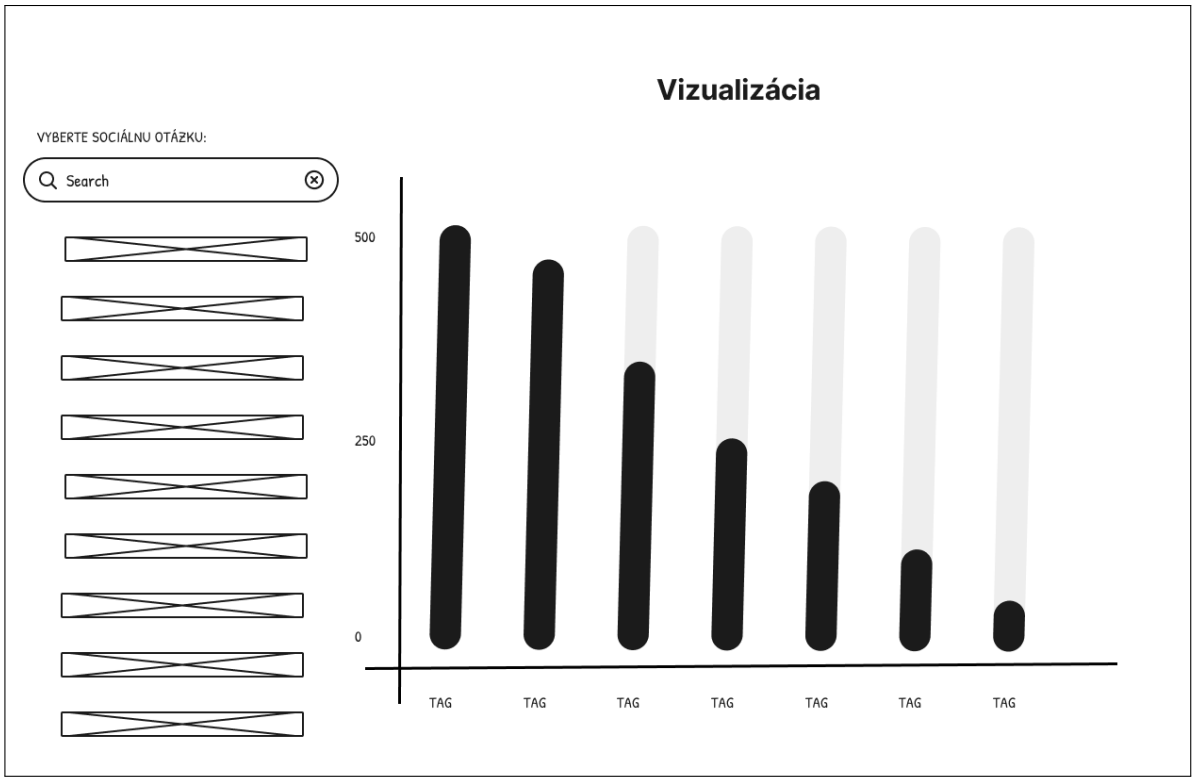
- **Zobrazenie Zoznamu Príspevkov:**
  - Modul poskytuje panel pre zvolenie sociálnej otázky. V tomto paneli sa nachádza aj filtrovacia možnosť. Užívatelia môžu vyhľadávať príspevky podľa názvu pomocou vyhľadávacieho poľa, čo uľahčuje navigáciu.
- **Interaktivita:**
  - Po kliknutí na príspevok v zozname sa na hlavnom obsahovom paneli zobrazí graf, ktorý ilustruje distribúciu názorov podľa zvolenej témy. Funkcionalita zahŕňa dynamické zobrazenie informácií podľa výberu užívateľa.
- **Backend Implementácia:)**
  - Server spravuje dáta potrebné pre frontend. Metóda index zabezpečuje načítanie príspevkov, ktoré sú poskytnuté obrazovke.
  - Metóda show získava a spracováva dáta o komentároch pre konkrétny príspevok. Tieto dáta sú potom poskytnuté ako odpoveď vo formáte JSON, ktorá obsahuje názvy tém (labels) a počty komentárov (data) pre každú tému.

- Zabezpečuje sa aktualizácia grafu bez nutnosti obnovovať stránku.

- **Filtrovanie a kategorizácia komentárov**
  - Na základe vybraných tém by modul mal filtrovať a kategorizovať komentáre do pod-tém, čo umožní hlbšiu analýzu názorov spojených s konkrétnymi sociálnymi otázkami.

Vizualizačný modul je kľúčový pre efektívne pochopenie rozsiahlych dát získaných z analýzy sociálnych médií. Interaktivita a grafické zobrazenie umožňujú užívateľom ľahko identifikovať a analyzovať trendy a vzory v diskusiách.

Návrh obrazovky obsluhy pre tento modul 4.6:



Obr. 4.6: Návrh ktorý slúži ako podklad pre vývoj modulu na zobrazenie názorov.

4.8 Základné systémové funkcionality poskytované Laravel frameworkom

- **Autentifikácia:** Laravel poskytuje autentifikačný systém, ktorý je možné prispôsobiť. Tento systém zahŕňa registráciu nových užívateľov, prihlásenie, odhlásenie a funkciu na obnovu zabudnutých hesiel.
- **Autorizácia a Ochrana Prístupu:** Na kontrolu prístupu k rôznym častiam aplikácie sa využívajú gates a policies.
- **Migrácie a Seeds:** systém migrácií pre definovanie štruktúry databáz a seeds na vyplnenie databázy preddefinovanými dátami. Užitočné pre rýchly vývoj a testovanie aplikácie.

- **ORM (Object-Relational Mapping):** ORM systém nazvaný Eloquent, ktorý je navrhnutý pre efektívnu prácu s databázami. Eloquent umožňuje pracovať s databázovými záznamami ako s objektami, čo zjednodušuje CRUD operácie a znižuje potrebu písania surového SQL kódu.
- **Testovací Server:**PHP server pre lokálne testovanie a vývoj. Spustením príkazu ‘php artisan serve‘ sa vytvorí lokálny testovací server.

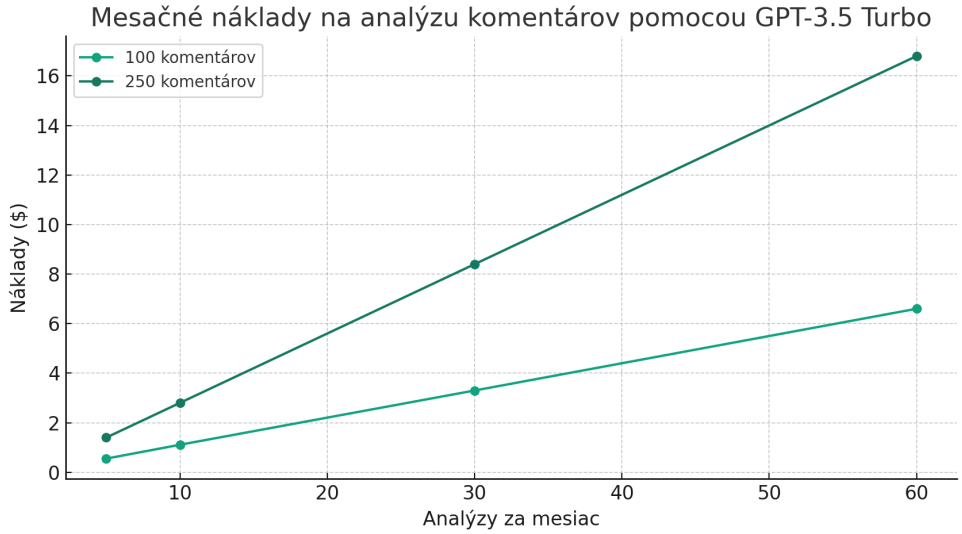
4.9 Cenové požiadavky využívania systému

Aplikácia používa CHAT GPT API, konkrétne model CHAT GPT-3.5 Turbo. V procese analýzy sa komunikácia s API využíva na dvoch miestach. Pre získanie pod-tém a pre priradenie pod-tém ku komentárom, ktoré sú rozdelené do skupín po 15 až kým nie sú analyzované všetky.

Cenník pre tento model je:

Model	Input	Output
gpt-3.5-turbo-0125	$0.50 / 1M tokenov	$1.50 / 1M tokenov

To znamená, že pri analýze témy pri pod-témach je to približne 950 tokenov a pri priradení pod-tém k názorom záleží hlavne na dĺžke názoru, pohybuje sa to od 1600 do 2500. Pri 100 komentároch to je približne 200 000 tokenov, čo pri cenníku je 0,11$ alebo pri aktuálnej konverzii tiež 0,11 eur. V tomto je započítaná aj cena odpovedí modulu. Pre lepšiu predstavu pozrite graf 4.7 možných mesačných nákladov.



Obr. 4.7: Tento graf znázorňuje mesačné náklady na analýzu komentárov pri dvoch rôznych objemoch dát: 100 a 250 komentárov.

Kapitola 5

Implementácia

Kapitola rozoberá praktickú aplikáciu teoretickej a návrhovej časti. Cieľom je preniesť návrhy do konkrétnej formy a demonštrovať, ako boli realizované jednotlivé komponenty systému pre zber, ukladanie a analýzu dát získaných zo sociálnych sieti. Pre jednotlivé moduly predstavíme ich časti ako je controller a view čo predstavuje základné časti modulu. Rozoberá sa integrácia modulov a taktiež akým spôsob bola riešená databáza. Poukázané je aj na problémy na ktoré sa narazilo pri implementácií systému.

5.1 Štruktúra systému

Aplikácia je riešená pomocou Laravel frameworku čím sa uľahčila práca s viacerými komponentami systému. Prostredie prišlo prednastavené s viacerými adresármi do ktorých sa už len implementovali konkrétne časti. Niktoré hlavné predstavím pre lepšiu orientáciu kde sa čo nachádza.

- **App/Models:**
  - **Účel:** Modely v Laraveli sú základné komponenty, ktoré reprezentujú tabuľky v databáze a zabezpečujú interakciu s databázou. Tento adresár obsahuje všetky modely Eloquent, ktoré umožňujú vykonávať databázové operácie. Medzi ktoré patria CRUD (vytváranie, čítanie, aktualizácia, mazanie) operácie, bez priameho písania SQL dotazov.
  - **Obsah:** Modelové súbory definujúce vlastnosti a vzťahy medzi dátovými modelmi, ako napríklad vzťahy typu „jeden k mnohým“ alebo „mnoho k mnohým“.
- **App/Http:**
  - **Controllers:** Tento pod-adresár obsahuje ovládače, ktoré sú súčasťou MVC architektúry. Ovládače sú zodpovedné za spracovanie vstupov z HTTP požiadaviek, vykonávanie logiky aplikácie a odoslanie odpovede späť klientovi.
- **Database/Migrations:**
  - **Účel:** Migrácie sú ako verzovací systém pre databázové schémy. Umožňujú definovať a zdieľať databázovú štruktúru aplikácie. S migráciami sa dá vytvárať meniť a mať kontrolu nad zmenami štruktúry databázových tabuliek.

  - **Obsah:** Súbory migrácií, ktoré predstavujú jednotlivé kroky v evolúcii databázovej schémy, každý so značkou kedy bol vytvorený.

- **Database/Seeds:**
  - **Účel:** Seeder súbory sú využívané na naplnenie databázy počiatočnými alebo testovacími dátami. Používané pre vývoj a testovanie, keď treba overiť funkcionalitu častí systému. **Obsah:** Skripty PHP ktoré definujú ako sa má naplniť databáza testovacími údajmi.

- **Public/index.php:**
  - **Účel:** Vstupný bod pre všetky požiadavky odoslané na Laravel aplikáciu. To znamená, že všetky HTTP požiadavky smerujú cez tento súbor a sú potom presmerované do aplikácie.
  - **Obsah:** Spúšťač Laravel aplikácie, ktorý inicializuje a spúšťa framework.

- **Public/CSS, JS, obrázky:**
  - **Účel:** Adresár public ukladá statické zdroje. Sú tu umiestnené preto, aby boli priamo prístupné klientovi.
  - **Obsah:** Štýlovacie súbory (CSS), JavaScriptové súbory (JS), obrázky (PNG, JPG, SVG atď.).

- **resources/views:**
  - **Účel:** Vo views sa nachádzajú Blade šablóny, ktoré Laravel používa na vykreslenie HTML stránok. Blade je výkonný šablónovací nástroj Laravelu písaný v PHP jazyku. Umožňuje dátové spojenia, dedičnosť šablón a vkladanie komponentov.
  - **Obsah:** Šablóny Blade s príponou .blade.php, ktoré môžu obsahovať HTML značky spolu s Blade direktívami. Tieto šablóny podporujú separáciu logiky aplikácie od prezentácie, čím uľahčujú údržbu kódu a zvyšujú jeho čitateľnosť podľa štandardov MVC.

- **routes/web.php:**
  - **Účel:** Súbor web.php obsahuje definície trás, ktoré sú určené pre webový rozhranie aplikácie. Tieto trasy môžu mať funkcie ako autentifikácia, CSRF[1] ochrana.
  - **Obsah:** PHP súbory s definíciami trás, kde každá trasa má priradenú URL (cestu), ovládač a metódu.

- **.env:**
  - **Účel:** Súbor .env je určený na uchovávanie citlivých a špecifických nastavení pre dané prostredie. Tieto sú databázové pripojenia, API kľúče, heslá a iné dôležité konfiguračné premenné.
  - **Obsah:** Obsahuje databázové pripojenie a poverenia. Nastavenia mailového servera, API kľúče pre rôzne služby. Nastavenia aplikácie ako APP_ENV, ktorý určuje prostredie v ktorom aplikácia beží.

[1] CSRF

5.2 Implementácia databázy a modelov

Venujeme sa tu krokom potrebným pre implementáciu databázy v našej aplikácii, pričom hlavný dôraz kladieme na proces tvorby tabuliek, definovanie modelov a seedovanie databázy. Cieľom je poskytnúť jasné porozumenie toho, ako sme prešli od konceptuálnych návrhov databázy k jej skutočnej realizácii v rámci vývojového prostredia.

Tvorba tabuliek

Proces tvorby tabuliek začína definovaním migrácií v Laraveli, čo sú základné skripty ktoré popisujú štruktúru databázových tabuliek a vzťahy medzi nimi. Migrácie poskytujú metódu up(), ktorá obsahuje logiku na vytvorenie tabuliek. Metódu down(), ktorá definuje ako tieto zmeny vrátiť späť. Týmto spôsobom Laravel vieme spravovať históriu databázovej schémy a poskytuje systém pre správu verzií databázy. Vytvorenie takej tabuľky a práca s ňou prebiehala pomocou týchto príkazov:

- **php artisan migrate**
  Spustí všetky nové migrácie, ktoré ešte neboli aplikované na databázu. Príkaz je základným nástrojom na aktualizáciu štruktúry databázy podľa najnovšieho stavu migračných súborov.
- **php artisan migrate:rollback**
  Vráti späť poslednú skupinu migračných operácií. To umožňuje jednoducho vrátiť databázu do predchádzajúceho stavu, ak najnovšie zmeny spôsobia problémy.
- **php artisan migrate:refresh**
  Obnoví databázu spustením príkazu pre všetky migrácie a následne ich znovu aplikuje. Toto je užitočné pre kompletné obnovenie databázy do pôvodného stavu podľa migračných súborov.
- **php artisan migrate:status**
  Zobrazí zoznam všetkých migrácií spolu s informáciou o tom, či boli alebo neboli aplikované. Užitočný pre kontrolu stavu migračných operácií.
- **php artisan db:seed**
  Spustí databázové seedery, ktoré naplnia databázu počiatočnými alebo testovacími dátami.
- **php artisan make:migration**
  Vytvorí nový migračný súbor. Tento príkaz je základom pre definovanie nových zmien v štruktúre databázy, ako je pridávanie tabuliek alebo stĺpcov.

Implementácia modelov

Po vytvorení tabuliek sa presúvame k definícii modelov, ktoré v Laraveli využívajú Eloquent ORM (Object-Relational Mapping) pre prácu s dátami v objektovo-orientovanom štýle. Modely v Laraveli sú zásadné pre interakciu s databázou, keďže každý model zodpovedá tabuľke v databáze a jeho inštancie predstavujú záznamy v tejto tabuľke. Modely reprezentujú dáta, definujú vzťahy (napríklad jedno k mnohým, mnoho k mnohým). Taktiež určujú pravidlá validácie.

Nastavenia vykonávané pri tvorbe modelu comments:

- $fillable: Atribút určuje ktoré stĺpce databázy môže Eloquent masovo priraďovať. V našom prípade sú to post_id, topic_id, comment_text, date_scraped. Toto zabezpečuje, že pri vytváraní alebo aktualizácii záznamov cez model môžu byť tieto atribúty bezpečne nastavené.
- $primaryKey: Atribút $primaryKey určuje, ktorý stĺpec v tabuľke sa má použiť ako primárny kľúč. V tomto prípade je to comment_id.
- $table: Atribút $table explicitne udáva, ktorá tabuľka databázy je spojená s modelom. Pre tento model je to tabuľka comments.

Seedovanie databázy

Seedovanie databázy je proces ktorým sa do prázdnej databázy vložia počiatočné dáta. Laravel poskytuje mechanizmus seeders, ktorý umožňuje automatizované naplnenie databázy testovacími údajmi. Toto je užitočné nielen pre testovanie a vývoj, ale aj pre inicializáciu aplikácie s predvolenými dátami po jej nasadení. Počiatočné dáta pre náš systém boli prevedené z dátovej sady získanej na začiatku práce. Pre ich štýl uloženia sa vytvoril program ktorí ich spracoval a vytvoril jednotlivé záznamové údaje. Tieto údaje boli skopírované do tohoto súboru. Programy ktoré sa vytvorili sú podobné manuálnemu nahratiu dát v module pre zber dát preto rozšírenie o túto funkcionalitu by nezabralo veľa času.

5.3 Implementácia modulu pre zber dát

Cieľom tohto modulu je automatizovaný zber informácií z externých zdrojov, ako sú Reddit a Facebook, pričom každý zdroj vyžaduje špecifický prístup a spracovanie dát.

Jednou zo základných častí je spúšťanie zberného procesu. Keďže sú skripty písané v Pythone a uložené na serverovej časti, je treba ich spustiť pomocou nového procesu. Knižnica 'Symfony'[2] toto priamo umožňuje. Je potrebné nastaviť environmentálne parametre ako SYSTEMROOT a PATH. Tieto premenné zabezpečujú že proces má správne nastavenia na prístup k systémovým zdrojom a iným programom potrebným pre jeho beh. Taktiež je potrebné špecifikovať miesto uloženia skriptu a predávať parametre z aplikácie. Tieto premenné sú dynamické a slúžia ako argumenty pre Python skript, umožňujúce mu pracovať s rôznymi otázkami.

Uvedené sú jednotlivé časti modulu a ich implementácia.

Controller

- **Metóda scrapeReddit** Metóda spracováva požiadavky na získavanie dát z platformy Reddit. Využíva externý Python skript (scrape-reddit.py) s parametrami získanými z HTTP požiadavky. Metóda zaznamenáva akcie do záznamov, spracováva výsledky skriptu a v prípade úspechu vracia dáta v JSON formáte. Pri neúspechu metóda spracuje chyby a vráti adekvátnu odpoveď.
- **Metóda scrapeFacebook:** Podobne ako pri Reddite táto metóda zabezpečuje zber dát z Facebooku. Vyžaduje overenie vstupných dát, ako sú téma diskusie, email a heslo. Ďalej kontrolované cez Laravel validátor. Ak sú údaje správne spustí sa Python

[2] Symfony

skript s príslušnými argumentmi a získané výsledky sú opäť vrátené v JSON formáte alebo ako text v prípade chyby.

- **Metóda storePost:** Ukladanie príspevkov: Metóda storePost spravuje ukladanie príspevkov a pridružených komentárov do databázy. Využíva validáciu dát na overenie, či sú vstupné polia ako názov príspevku a komentáre správne. Po validácii sa príspevky a komentáre uložia a vráti sa JSON odpoveď.

View

V šablóne sa nachádza formulár, ktorý umožňuje užívateľovi zadávať sociálne otázky a vyberať jazyk, pre ktorý chce vykonať vyhľadávanie. Funkcionalita pre tento proces je nastavená ale nevyužíva sa aby sa názory zbierali pre obe národnosti. Toto je realizované pomocou input polí a select boxu:

- Input pre sociálnu otázku: Umožňuje užívateľovi zadávať otázku, ktorá bude použitá na zber dát.
- Select box pre výber jazyka: Užívateľ vie vybrať jazyk v ktorom chce vykonávať zber dát, čím sa zabezpečuje, že dáta budú relevantné pre špecifikovaný jazykový kontext. Ako bolo spomenuté funkcionalita je pripravená ale nie spustená.
- Dve input polia pre zadanie mena a hesla do platformy Facebook. Využité pri emulácii užívateľa. Skript tieto údaje automaticky vyplní na platforme.

5.4 Implementácia modulu pre úpravu názorov

Modul sa zameriava na manipuláciu s názormi užívateľom zhromaždenými z online platforiem, prostredníctvom rozhrania aplikácie. Hlavným cieľom je umožniť správu akou je zobrazovanie, úpravu a mazanie komentárov pridružených k príspevkom.

Z teoretického hľadiska je tento modul zanedbateľný z dôvodu integrity analýzy. Úprava názorov a ich mazanie môže ovplyvniť výslednú analýzu témy. Tento modul ale rieši jeden z najväčších problémov, ktorým sú irelevantné dáta pod relevantnými príspevkami. Často sa stane, že užívatelia nevyjadria svoj názor k téme, ale vyjadrujú sa k ľuďom alebo k autorovi. Taktiež sa môžeme vyskytnúť so scenárom, kedy je časť komentáru relevantná a chceme odstrániť nadbytočné vety, ktoré sa témy netýkajú.

V module sa pracuje hlavne s Eloquent modelmi, ktoré uľahčujú spravovacie operácie nad dátami. Ďalej sú uvedené jednotlivé časti modulu a ich implementácia.

Controller

- getComments: Načítava komentáre pre špecifikovaný príspevok podľa ID získaného z URL parametra. Vracia komentáre ako JSON odpoveď.
- deleteComment: Maže komentár podľa jeho ID, ktoré je získané z URL parametra. Zaznamenáva pokus o mazanie a jeho výsledok do záznamu. Vracia JSON odpoveď so stavom mazania.
- updateComment: Aktualizuje text komentára na základe jeho ID a nového textu prijatého cez požiadavku. Zaznamenáva aktualizáciu a jej výsledok. Vracia JSON odpoveď o úspechu alebo neúspechu operácie.

View

- Formulár pre výber príspevkov: Rozbaľovacie menu pre výber príspevkov, ktoré zobrazuje názvy príspevkov a umožňuje používateľovi ho vybrať pre zobrazenie komentárov.
- Zobrazenie tém príspevkov: Dynamické zobrazenie tém spojených s vybraným príspevkom. Aktualizuje sa podľa výberu príspevku v rozbaľovacom menu.
- Tabuľka komentárov: Dynamicky generovaná tabuľka pre zobrazenie komentárov vybraného príspevku. Obsahuje stĺpce pre text komentára a možnosti akcií (editácia, mazanie).

5.5 Implementácia modulu pre analýzu názorov

Cieľom tohto modulu je automatizovaná analýza názorov z komentárov na sociálne otázky, s možnosťou automatického vytvorenia pod-tém. Modul umožňuje dynamické pridávanie nových tém a štítkov ku komentárom, čo umožňuje prispôsobenie analýzy aktuálnym požiadavkám. Tak isto upresňujem inštrukcie pre modely ktoré boli zostavené podľa inštrukcii [9].

Controller

- **Metóda saveTopics:** Umožňuje uloženie nových pod-tém priradených k príspevkom. Pod-témy môžu byť vytvorené používateľmi na základe obsahu príspevku, čo umožňuje flexibilné rozširovanie databázy pod-tém podľa potrieb analýzy.
- **Metóda SugestSubtopics:** Využíva externú službu AI na generovanie návrhov pod-tém z textu príspevkov. Táto funkcia pomáha automaticky identifikovať relevantné štítky pre kategorizáciu komentárov.
- **Metóda getCommentsFilteredByTopics:** Získava komentáre ktoré sú zatiaľ neanalyzované, a priraďuje ich k vybraným témam na základe analýzy obsahu.

View

- **Výber príspevku:** Dropdown menu umožňuje výber príspevku, na ktorý sa má analýza vykonať. Každý príspevok je spárovaný s informáciou o počte neanalyzovaných komentárov, čo poskytuje prehľad o tom, koľko dát ešte nebolo spracovaných.
- **Priradenie štítkov:** Umožňuje pridávanie nových štítkov k príspevkom, ako aj automatické návrhy štítkov. Pokiaľ sú štítky už priradené k téme tak sa len zobrazia.
- **Tlačidlo pre analýzu:** Po výbere príspevku a priradení štítkov môže používateľ spustiť analýzu komentárov. Výsledky analýzy môžu byť potom zobrazené v aplikácii.

Analýza pomocou CHAT GPT API

Komunikácia je implementovaná ako servis pre aplikáciu. Nachádzajú sa tu 2 funkcie ‘generateSubtopics‘ a ‘assigneSubtopics‘ ktoré zabezpečujú správnu komunikáciu medzi aplikáciou a CHAT GPT.

Funkcia **generateSubtopics($topic)** automatizuje proces generovania pod-tém pre danú hlavnú tému pomocou OpenAI. Používa sa Laravel HTTP klient na odoslanie požiadavky na OpenAI API. Požiadavka obsahuje nasledujúce hlavičky:

- **Authorization:** Bearer token pre autorizáciu pri prístupe k OpenAI API.
- **Content-Type:** Nastavený na 'application/json' pre správne načítanie obsahu požiadavky.

Telo HTTP požiadavky obsahuje určenie modelu GPT-3.5-turbo. Správy definujú kontext konverzácie, kde systémová správa definuje úlohu a používateľská správa obsahuje aktuálnu tému. Obsahujú tiež pokyny pre AI na vygenerovanie sedmičky pod-tém. Štruktúra pod-tém zahŕňa šesť špecifických oblastí a jednu všeobecnú kategóriu „Ostatné“, čo pomáha udržať organizované a konzistentné výstupy. Text úlohy je špeciálne navrhnutý a delí sa niekoľko častí ktoré dokopy tvoria celé inštrukcie pre generovanie pod-tém:

Vstupná Téma a Úloha

Vstupná téma: Dostanete tému súvisiacu so sociálnymi otázkami. Témy sú široké oblasti ktoré vyžadujú nanucované rozčlenenie pre ich plné preskúmanie.

Úloha: Vašou úlohou je identifikovať a zoznamovať sedem pod-tém, ktoré pokrývajú rôzne dimenzie danej témy. Tieto pod-témy by mali poskytnúť komplexný prehľad o hlavných problémoch, perspektívach a debatách ktoré obklopujú tému.

Štruktúra Pod-tém

Pod-témy musia zahŕňať šesť špecifických oblastí relevantných pre tému a jednu všeobecnú kategóriu označenú ako 'Ostatné'. Táto je navrhnutá tak aby zahŕňala akékoľvek ďalšie aspekty alebo perspektívy, ktoré sa nezmestia do ostatných šiestich kategórií.

Jazyk a Formát Výstupu

Jazyk: Témy a pod-témy budú poskytnuté v češtine alebo slovenčine. Uistite sa, že pod-témy sú presne reprezentované v danom jazyku a sú výstižné a nie dlhšie ako 3 slová!

Formát výstupu: Vaša odpoveď by mala striktne dodržiavať tento formát: *Podtémy: [Podtéma 1], [Podtéma 2], [Podtéma 3], [Podtéma 4], [Podtéma 5], [Podtéma 6], Ostatné.* Nahraďte [Podtéma 1] až [Podtéma 6] konkrétnymi podtémami, ktoré identifikujete pre tému. Striktne ich musí byť dokopy 6 + Ostatné.

Príklady

- Pre tému ako 'Názor na členstvo v Európskej únii (EÚ)', váš výstup by mohol vyzerať takto: *Podtémy: Ekonomické výhody, Politický vplyv a suverenita, Byrokracia a legislatíva, Migrácia a voľný pohyb, Obavy a kritika EÚ, Pozitívny postoj k EÚ, Ostatné.*
- Pre tému ako 'Elektrické autá' príkladný výstup by mohol byť: *Podtémy: Ekonomika, Logistika, Ekologický dopad, Kultúrne zmeny, Podpora elektromobilov, Opozícia voči elektromobilom, Ostatné.*

Funkcia **assigneSubtopics($post,$topics,$comments)** je navrhnutá na analyzovanie a kategorizáciu komentárov do definovaných pod-tém pre danú sociálnu tému. Volanie tejto funkcie prebieha opakovane až kým nie sú analyzované všetky komentáre. Hlavička a telo funkcie je podobne nakonfigurované ako v prvej časti. Líši sa jej definícia úlohy ktorá je špeciálne navrhnutá a delí sa niekoľko častí. Dokopy tvoria celé inštrukcie pre automatické priraďovanie pod-tém:

Inštrukcia a odpoveď

> **Inštrukcie:** Pre danú sociálnu tému budú komentáre analyzované a kategorizované do vopred definovaných podtém. Tento proces zahŕňa identifikáciu a priradenie relevantných značiek ku každému komentáru, ktoré odzrkadľujú jeho obsah vo vzťahu k hlavnej téme a špecifikovaným podtémam.
>
> **Hlavná téma:** '$post' Podtémy: '$topicsString' Formát odpovede: Pre každý komentár by odpoveď mala obsahovať číslo komentára presne vo formáte v akom je pred komentárom nasledované 'TAGY:' a zoznamom priradených značiek oddelených čiarkou toto je veľmi dôležité používaj iba toto slovo. Nepíš nič iné ani komentár dodržuj [číslo komentára]. TAGY:.

Poznámka

> Poznámka: Je dôležité brať do úvahy nuance v komentároch pre správne priradenie značiek. Ak komentár spadá do viacerých podtém, priraďte všetky relevantné značky. Každý komentár začína číslom a končí prázdnym riadkom, čo uľahčuje jeho identifikáciu a spracovanie.

5.6 Implementácia modulu pre vizualizáciu názorov

Modul poskytuje grafické zobrazenie analyzovaných údajov získaných z príspevkov a ich komentárov na sociálnych platformách. Umožňuje vizualizáciu distribúcie komentárov podľa rôznych pod-tém príspevkov v reálnom čase. Vizualizácia umožňuje užívateľom vidieť vzory, trendy a výnimky vo veľkých dátových súboroch rýchlejšie než by to bolo možné len prostredníctvom textových oznámení alebo surových tabuliek. Stĺpcové grafy sú ideálne na porovnávanie množstva alebo počtu naprieč rôznymi kategóriami. Každý stĺpec poskytuje vizuálnu reprezentáciu veľkosti príslušného atribútu, čo umožňuje ľahko porovnať rôzne skupiny.

Controller

- **show**: Metoda spracováva dopyty na zobrazenie špecifických príspevkov a ich podtém. Na základe identifikátora príspevku získaného z HTTP požiadavky sa vykonáva načítanie a jeho pod-tém s počtami komentárov. Informácie o témach a počte sú agregované do polí, ktoré sa následne používajú pre vizualizáciu.

View

- **Vyhľadávanie**: Umožňujú užívateľom vyhľadávať príspevky podľa kľúčových slov.

- **Graf**: Canvas používaný knižnicou ‘Chart.js‘[3] pre zobrazenie počtu komentárov v závislosti od témy príspevku. Interakcia s prvkami na obrazovke dynamicky aktualizuje graf.

[3] Chart.js

Kapitola 6

Experimenty

V tejto časti pracujeme na experimentoch ktoré priblížia ako dobre sa so systémom pracuje a aké výsledky dosahuje. V experimentoch kde boli potrebný respondenti som požiadal troch ľudí o vypracovanie otázok.

6.1 predstavenie respondentov

- **Jakub Majer:**
  - Vek: 21
  - Práca: Študent informačných technológii
- **Jana Čierna:**
  - Vek: 43
  - Práca: Obchodný zástupca v logistickej firme
- **Ivan Mahút:**
  - Vek: 22
  - Práca: Študent informačných technológii

6.2 Experiment užívateľskej skúsenosti na stránke

Cieľom experimentu je zistiť, ako intuitívne môžu užívatelia vykonávať základné úlohy na stránke. Obsahuje tri hlavné úlohy ktoré účastníci musia vykonať a na konci vyplnia tri otvorené otázky. Výsledky sú vyhodnotené a je vyvodený záver. Nasleduje popis užívateľských úloh ktoré boli požadované od účastníkov tohoto experimentu.

Úloha 1

- **Inštrukcia:** Vykonanie stiahnutia dát na ľubovoľnú sociálnu otázku a uložiť získané dáta.
- **Hodnotenie:**
  - **Čas dokončenia:** Ako dlho trvalo respondentovi vykonať úlohu.
  - **Počet chýb:** Počet zlých akcii prevedených účastníkom.

Úloha 2:

- **Inštrukcia:** Previesť analýzu zadanej sociálnej otázky vrátane nastavenia pod-tém.
- **Hodnotenie:**
  - **Čas dokončenia:** Ako dlho trvalo respondentovi vykonať úlohu.
  - **Počet chýb:** Počet zlých akcii prevedených účastníkom.

Úloha 3:

- **Inštrukcia:** Prejdite do menu a potom pozrite ako vyzerá vizualizácia vašej analýzy.
- **Hodnotenie:**
  - **Čas dokončenia:** Ako dlho trvalo respondentovi vykonať úlohu.
  - **Počet chýb:** Počet zlých akcii prevedených účastníkom.

Úloha 4: Otvorené otázky

- Ako by ste ohodnotili jednoduchosť používania aplikácie na škále od 1 do 10?
- Aké aspekty navigácie na webovej stránke by ste zlepšili?
- Ako by ste aplikáciu využívali v budúcnosti?

Vyhodnotenie predošlého dotazníku:

- **Priemerný čas dokončenia:**
  - **Úloha 1:** 1:11
  - **Úloha 2:** 0:55
  - **Úloha 3:** 0:35
- **Priemerný počet chýb:**
  - **Úloha 1:** 0
  - **Úloha 2:** 1
  - **Úloha 3:** 0
- **Otvorená otázka 1:**
  - **Respondent 1:** 9
  - **Respondent 2:** 7
  - **Respondent 2:** 8
- **Otvorená otázka 2:**
  - **Respondent 1:** Pri výbere štítkov na analýzu konkrétnej témy by som na tlačidlo uložiť dal ako povinné alebo ho zautomatizoval.
  - **Respondent 2:** Vyhľadávanie z Facebooku sa nespustilo hneď.

  - **Respondent 3:** Názory nepatrili k hľadanej otázke.

- **Otvorená otázka 3:**
  - **Respondent 1:** Aplikáciu by som využíval na získanie prehľadu názorov ľudí na zmeny zákonov ohlásené vládou.
  - **Respondent 2:** Pre kuriozitu aké sú názory keďže môžem analyzovať hocičo.
  - **Respondent 3:** Keby som písal prácu kde je potrebné získať údaje ako sú názory, využil by som to kvôli jeho jednoduchosti.

V súvislosti s používaním aplikácie sa našli určité chyby ako sú navigačné problémy pri analýze a neočakávané dáta vrátené zo sociálnej sieti. Ovplyvniť čo vracajú vyhľadávače na sociálnych sieťach sa nedá. Každopádne dalo by sa navrhnúť rozšírenie kde podstatné časti príspevku sa pošlú na overenie do GPT-3.5 a podľa výsledku by sa stiahli komentáre. Tak isto pre zlepšenie by sa v budúcnosti mala upraviť navigácia a práca s ukladaním štítkov pre užívateľov. Silnými stránkami je jednoduchosť používania a rýchlosť analýz. Respondenti si pochvaľovali celkovú flexibilitu systému v podobe analýzy ľubovoľnej témy.

6.3 Ohodnotenie analýzy prevedenej systémom

Hlavnou časťou programu je správne prevedenie analýzy. Preto prevediem test kde sa ohodnotí každý komentár a k nemu priradené pod-témy. Sociálna otázka je vybraná z rady predom pripravených dát je to názor na elektrické vozidlá. K nej sú priradené už spomínané pod-témy. Analýza bude tak isto časovaná pre prehľad koľko času systému zaberie spracovať 150 názorov. Hodnotenie je v niektorých prípadoch subjektívne preto sa pri sporných prípadoch radím s respondentmi.

Metriky hodnotenia

Predstavujem metriky na základe ktorých sa hodnotenie vykonalo:

Presnosť: Meria podiel správne priradených štítkov z celkového počtu štítkov priradených modelom. Výpočet je daný vzorcom:

$$\text{Presnosť} = \frac{\text{Počet správne priradených pod-tém}}{\text{Počet priradených pod-tém}}$$

Pokrytie: Pokrytie hodnotí aký veľký podiel z relevantných pod-tém bol správne identifikovaný systémom. Výpočet vyzerá nasledovne:

$$\text{Pokrytie} = \frac{\text{Počet správne identifikovaných relevantných štítkov}}{\text{Počet relevantných štítkov}}$$

F1 Skóre: F1 skóre je harmonický priemer presnosti a pokrytia, čo znamená, že zohľadňuje oba aspekty - presnosť aj pokrytie. Metrika je užitočná pri vyhodnotení celkového výkonu analýzy:

$$F1 = 2 \cdot \frac{\text{Presnosť} \times \text{Pokrytie}}{\text{Presnosť} + \text{Pokrytie}}$$

Priemerné F1 Skóre: Pre celkové zhodnotenie presnosti na dátach vypočítame priemerné F1 skóre pre všetky príklady, čo nám poskytne agregovaný pohľad na výkonnosť:

$$\text{Priemerné F1} = \frac{\sum_{i=1}^{N} F1_i}{N}$$

Hodnotenie

Nasledujú hodnoty získané z experimentu predstavené v metrikách a ďalšie

- **Rýchlosť analýzy a chyby**
  - **Čas:** Analýza trvala 59,3 sekundy čo je z ohľadom na počet dát prijateľné.
  - **Pokrytie:** Ostatné(11), Ekonomika(61), Logistika(35), Kultúra(8), Podpora(10), Nepodpora(64)
  - **Chyby:** Podarilo sa zanalyzovať 131/150 komentárov. 19 neanalyzovaných je pravdepodobne chybou zlého formátu odpovede GPT API.
- **Metriky**
  - **Priemerné F1 skóre:** 76,30% - Skóre naznačuje že systém má relatívne vysokú schopnosť balansovať medzi presnosťou a pokrytím. Pričom minimalizuje počet nesprávnych alebo štítkov.
  - **Priemerné pokrytie:** 79,10% - Pokrytie ukazuje schopnosť identifikovať veľký podiel relevantných štítkov z tých, ktoré by mali byť priradené.
  - **Priemerná presnosť:** 76,54% - Vysoká priemerná presnosť naznačuje, že štítky priradené systémom sú správne a relevantné vo väčšine prípadov.

Výsledok

Výsledky experimentu poukazujú že proces analýzy dosiahol veľmi dobrú rovnováhu medzi presnosťou a pokrytím. S priemerným F1 skóre 76,30%, priemerným pokrytím 79,10% a priemernou presnosťou 76,54% môžeme byť spokojní s výkonom systému, najmä pokiaľ ide o správne identifikovanie a priraďovanie relevantných pod-tém. Tieto výsledky sú obzvlášť povzbudzujúce keď zohľadníme že analýza trvala menej ako minútu a bola schopná spracovať veľkú časť komentárov.

Napriek týmto pozitívnym výsledkom existuje priestor pre vylepšenie. Konkrétne, potenciálne zlepšenia zahŕňajú prepracovanie inštrukcií pre GPT-3.5 alebo presunutie na lepší model ako GPT-4. Tento presun by pomohol aj k vylepšeniu samotného F1 skóre.

Kapitola 7

Záver

Cieľom aplikácie je efektívne zbierať, ukladať, analyzovať a vizualizovať dáta z externých zdrojov, ako sú sociálne siete Reddit a Facebook. Hlavným zmyslom systému je poskytnúť rozhranie pre analýzu sociálnych otázok. Poskytnúť užívateľom možnosť sa v nich zorientovať a ďalej využívať poznatky z analýz. Aplikácia má za úlohu umožniť užívateľom interaktívne a efektívne pracovať s týmito dátami. Pomocou používateľského rozhrania uľahčiť prácu s funkcionalitami pre všetky potrebné etapy procesu.

Dosiahnutým výsledkom práce je systém, ktorý používa externý model CHAT GPT 3 turbo od OPEN AI. Pre navrhovanie pod-tém k sociálnym otázkam ako aj priraďovanie jednotlivých pod-tém ku týmto názorom. Produktom práce je aj zhromaždená dátová sada. Dáta sú získané pre 3 sociálne otázky ako aj manuálne kategorizované. Aplikácia ponúka grafovú vizualizáciu rozloženia jednotlivých pod-tém. Systém je všeobecne stavaný, aby poskytol dynamickú analýzu na širokú škálu tém, čo je umožnené využívaním silného externého modelu. Analýza preto nemusí byť striktne ohraničená na sociálne otázky, ale práve s nimi pracuje najlepšie. Celková rýchlosť analýz je prijateľná a bolo zistené že pri 150 komentároch trvala maximálne 1 minútu. Dosiahnuté priemerné F1 skóre systému je 76,30%. Systém má vysokú schopnosť efektívne identifikovať a priraďovať relevantné pod-témy k poskytnutým názorom.

Zhodnotenie dosiahnutej práce nie je perfektné. Viaceré časti projektu by potrebovali vylepšenie. Čo zahŕňa korektnejšie overenie získaných dát, ako aj lepšia prezentácia výsledkov. Celkovo systém spĺňa požiadavky, no v niektorých situáciách sa nespráva korektne. Toto bolo spomenuté aj vo výsledkoch experimentov.

Literatúra

[1] *Laravel Documentation*. Navštívené dňa 2023-04-03. Dostupné z: https://laravel.com/docs.

[2] AUTOR, N. *REST API Design: Best Practices for RESTful API Design*. 2023. Navštívené dňa 2023-04-04. Dostupné z: https://www.altexsoft.com/blog/rest-api-design/.

[3] CASTRO, E. a HYSLOP, B. *HTML5 and CSS3*. Berkeley, CA: Peachpit Press, 2020.

[4] CENTER, P. R. *Americans' Social Media Use*. 2024. Dostupné z: https://www.pewresearch.org/internet/2024/01/31/americans-social-media-use/.

[5] CHANG, Y., WANG, X., WANG, J., WU, Y., YANG, L. et al. A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.* New York, NY, USA: Association for Computing Machinery. mar 2024, zv. 15, č. 3. DOI: 10.1145/3641289. ISSN 2157-6904. Dostupné z: https://doi.org/10.1145/3641289.

[6] DOCS, M. W. *Introduction to web APIs* [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction]. 2023. Navštívené dňa 2023-04-04.

[7] FREECODECAMP. *The HTML Handbook – Learn HTML for Beginners*. 2019. Dostupné z: https://www.freecodecamp.org/news/the-html-handbook/.

[8] GUNAWAN, R., RAHMATULLOH, A., DARMAWAN, I. a FIRDAUS, F. Comparison of Web Scraping Techniques : Regular Expression, HTML DOM and Xpath. In: *Proceedings of the 2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)*. Atlantis Press, 2019/03, s. 283–287. DOI: 10.2991/icoiese-18.2019.50. ISBN 978-94-6252-689-1. Dostupné z: https://doi.org/10.2991/icoiese-18.2019.50.

[9] OPENAI. *Prompt Engineering Guide*. 2024. Accessed: 2024-04-21. Dostupné z: https://platform.openai.com/docs/guides/prompt-engineering.

[10] SHEIKH, R. A., PAWAR, S. G., AISHWARYA, S. L. a GAURI, R. K. Chat GPT, Curse or Blessings. *International Research Journal of Innovations in Engineering and Technology*. September 2023, zv. 7, č. 9, s. 150–152. Copyright - © 2023. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms available at https://irjiet.com/about$_{o}pen_{a}ccess$; $Lastupdated - 2023 - 11 - 04$. $Dostupnz$: .

TAY, Y., DEHGHANI, M., BAHRI, D. a METZLER. A Survey of Transformers. *ArXiv preprint arXiv:2106.04554*. 2020. Navštívené dňa 2023-04-03. Dostupné z: https://ar5iv.labs.arxiv.org/html/2106.04554.

Príloha A

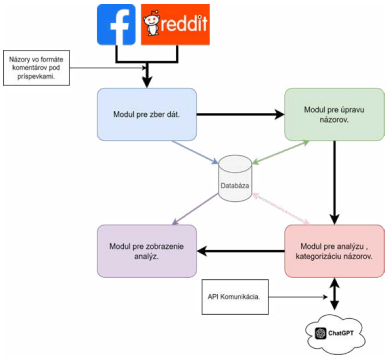
Plagát

Analýza postojů českých a slovenských uživatelů na základě dat ze sociálních sítí a webových diskusí

Dušan Slúka
Vedúci práce: doc. RNDr. Pavel Smrž, Ph.D.

Ciele práce

Cieľom tejto práce bolo navrhnúť a implementovať systém, ktorý využíva metódy umelej inteligencie pre analýzu textových dát získaných zo sociálnych sietí. Sociálne diskusie sa odohrávajú na viacerých miestach v sociálnej sieti. Systém by preto mal byť schopný automaticky vytvárať podkategórie tém a roztrieďovať jednotlivé postoje do týchto kategórií. Dôležité je, aby vedel spracovať širokú škálu sociálnych tém a vizuálne reprezentovať zistenia v užitočnej forme.



Prínosy

Práca prináša inovatívne využitie pokročilého modelu CHAT GPT 3 turbo od OPEN AI na analýzu a kategorizáciu sociálnych otázok. Umožňuje hlbšie pochopenie verejných názorov v digitálnom priestore.

Výsledky

Podarilo sa vytvoriť webovú aplikáciu, ktorá dokáže sťahovať, spracovávať a analyzovať názory na sociálne témy. Aplikácia bola podrobená experimentom, ktoré testovali ako dobre vie tieto dáta zanalizovať. Systém úspešne spracoval 131 z celkového počtu 150 názorov za 59,3 sekundy, pričom sa vyskytlo 19 chýb z dôvodu nekompatibility formátov dát. Pri analýze boli dosiahnuté hodnoty kľúčových metrík:

Metriky	Presnosť	Pokrytie	F1-Skóre
Priemerné hodnoty	76,54%	79,10%	76,30%

Webová aplikácia je prívetivá a intuitívna, poskytuje funkcionalitu pre zjednodušenie procesu.

Obr. A.1: Plagát práce