



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**ASPEKTOVĚ ORIENTOVANÁ ANALÝZA POSTOJŮ
Z UŽIVATELSKÝCH RECENZÍ POČÍTAČOVÝCH HER**

ASPECT-ORIENTED SENTIMENT ANALYSIS FROM USER REVIEWS OF COMPUTER GAMES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FRANTIŠEK SABOL

VEDOUcí PRÁCE

SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2023

Zadání bakalářské práce



140744

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Sabol František**
Program: Informační technologie
Specializace: Informační technologie
Název: **Aspektově orientovaná analýza postojů z uživatelských recenzí počítačových her**
Kategorie: Umělá inteligence
Akademický rok: 2022/23

Zadání:

1. Seznamte s moderními metodami strojového učení pomocí sítí typu transformer a jejich použitím pro analýzu postojů.
2. Vyberte vhodnou platformu, z níž je možné získat uživatelské recenze počítačových her, a shromážděte datovou sadu pro vývoj a průběžné vyhodnocování systému.
3. Navrhněte a implementujte systém, který dokáže v textu recenzí her identifikovat jednotlivé aspekty a analyzovat vyjadřované postoje.
4. Demonstrujte a vyhodnoťte vytvořený systém na shromážděných datech.
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: 1.11.2022
Termín pro odevzdání: 10.5.2023
Datum schválení: 5.5.2023

Abstrakt

Táto bakalárska práca sa zaoberá problematikou aspektovo-orientovanej analýzy sentimentu a implementáciou systému určeného k analýze používateľských recenzií videohier. Pri tvorbe systému nadväzuje na súčasne riešenia a zameriava sa na aplikovanie generatívnych jazykových modelov pre úlohu extrakcie štvoríc (aspekt, kategória, názor, sentiment). Vytvorený systém dokáže automatizovať proces získavania a analýzy recenzií. Pomocou webovej aplikácie poskytuje intuitívne používateľské rozhranie pre vyhľadávanie, filtrovanie a vizualizáciu výsledkov analýzy. Pre účely práce boli taktiež vytvorené dve dátové sady prevažne českých recenzií videohier.

Abstract

This bachelor's thesis delves into aspect-based sentiment analysis (ABSA) and the creation of a system tailored for examining video game user reviews. By building on established solutions, the system emphasizes the use of generative language models for extracting Aspect-Category-Opinion-Sentiment (ACOS) quadruples. The devised system streamlines the collection and evaluation of reviews, providing an intuitive user interface that enables effortless searching, filtering, and visualization of analytical outcomes via a web application. Furthermore, two datasets predominantly consisting of Czech video game reviews have been assembled.

Klíčové slová

aspektovo-orientovaná analýza sentimentu, recenzia, extrakcia dát z webu, generatívny jazykový model, hlboké učenie, extrakcia štvorice aspekt-kategória-názor-sentiment

Keywords

aspect-based sentiment analysis, review, web scraping, generative language model, deep learning, Aspect-Category-Opinion-Sentiment Quadruple Extraction

Citácia

SABOL, František. *Aspektově orientovaná analýza postojů z uživatelských recenzí počítačových her*. Brno, 2023. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. RNDr. Pavel Smrž, Ph.D.

Aspektově orientovaná analýza postojů z uživatelských recenzí počítačových her

Prehlásenie

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

.....
František Sabol
8. mája 2023

Podakovanie

Chcel by som poďakovať pánovi doc. RNDr. Pavlovi Smržovi, Ph.D. za cenné rady pri návrhu systému, odporúčania pri problémoch, a za ochotu a pomoc pri tvorbe bakalárskej práce.

Obsah

1	Úvod	3
2	Teoretický rozbor	4
2.1	Získavanie dát	4
2.2	Analýza sentimentu	7
2.3	Aspektovo-orientovaná analýza sentimentu	8
2.4	Riešenia založené na hlbokom učení	10
2.5	Neurónová sieť typu transformer	11
2.6	Text-to-Text Transfer Transformer	17
2.7	V súčasnosti používané metódy	18
3	Návrh riešenia	19
3.1	Návrh systému	19
3.2	Získavanie dát	19
3.3	Spracovanie a analýza recenzií	22
3.4	Vizualizácia výsledkov	26
4	Implementácia	29
4.1	Použité nástroje	29
4.2	Štruktúra aplikácie	31
4.3	Modul pre získavanie dát	33
4.4	Modul pre analýzu recenzií	35
4.5	Vizualizácia	36
5	Experimenty a vyhodnotenie	41
5.1	Porovnanie generatívnych jazykových modelov	41
5.2	Analýza trendov a porovnanie s realitou	43
5.3	Testovanie používateľského rozhrania	46
6	Záver	48
	Literatúra	49
	A Obsah priloženého pamäťového média	53
	B Relačný model databázy	54
	C Plagát	55

Zoznam obrázkov

2.1	Schéma architektúry transformer	12
2.2	Výpočet pozornosti	13
2.3	Schéma viachlavovej pozornosti	14
2.4	Vizualizácia jednotného formátu vstupu a výstupu pre model T5	17
3.1	Návrh systému	20
3.2	Distribúcia recenzií v dátových sadách na základe počtu znakov.	25
3.3	Mockup webovej aplikácie	26
4.1	Schéma aplikačného zásobníka webového API.	32
4.2	Zjednodušená schéma architektúry aplikácie.	37
4.3	Detailný pohľad na hru	39
4.4	Vizualizácia grafov časového vývoja sentimentu	40
5.1	Porovnanie modelu mT5 a ByT5 pre úlohu extrakcie trojíc	41
5.2	Porovnanie modelu mT5 a ByT5 pre úlohu extrakcie štvorcíc	42
5.3	Analýza trendov	44
5.4	Časový vývoj sentimentu	45

Kapitola 1

Úvod

V posledných rokoch sme svedkami stabilne rastúceho záujmu ľudí o videohry a s tým aj spojeného rastu počtu videohier. Tento trend má za následok zvyšujúcu sa konkurenciu medzi vývojármi, ktorí chcú uspieť a presadiť sa na trhu. Taktiež pre hráčov nemusí byť jednoduché si vybrať medzi množstvom podobných produktov. S riešením týchto problémov môžu do značnej miery pomôcť používateľské recenzie.

Vďaka distribučným platformám, akou je napríklad Steam¹, je možné získať mnoho metrík týkajúcich sa produktov, vrátane recenzií od používateľov. Formát recenzií, ktoré drvivá väčšina distribučných platforiem poskytuje, nám však dovoľuje získať len text recenzie a celkové hodnotenie produktu na určitej škále (napr. 0-5, odporúča-neodporúča), z čoho sa dá odvodiť len celková spokojnosť používateľov s produktom.

Recenzie sú pre hráčov a vývojárov veľmi dôležité. To potvrdzujú aj výsledky predošlých štúdií, ktoré ukazujú, že recenzie majú značný vplyv na komerčný úspech produktu [24]. Pre vývojárov je dôležité poznať názory používateľov, pretože im umožňujú zistiť, čo sa hráčom páči a čo by sa mohlo zlepšiť. To im pomôže vylepšiť budúce verzie hry a prispôbiť ju potrebám a preferenciám hráčov. Hráčom zasa môže prehľad silných a slabých stránok hry uľahčiť rozhodovanie sa o tom, či je vybraná hra pre nich vhodná alebo či zodpovedá sľubom vývojárov. Dáta, ktoré by poskytovali informácie o silných a slabých stránkach hry, však väčšina platforiem neposkytuje, no dajú sa z textu získať aspektovo-orientovanou analýzou sentimentu. Pri nízkom počte recenzií je možné získať prehľad len na základe manuálnej analýzy, no pre získanie prehľadu pri tisíckach recenzií to určite nebude stačiť. Systém, ktorý bude zhromažďovať používateľské recenzie z viacerých zdrojov a poskytovať prehľadnú analýzu sentimentu na úrovni aspektu, je preto veľmi žiadaný. Hráčom poskytne vhodný a rýchly nástroj ako si pred kúpou overiť, či daná hra splňa ich požiadavky a predstavy. Vývojári zase zistia, aké aspekty najviac ovplyvňujú spokojnosť používateľov.

Cieľom práce je teda vytvoriť systém, ktorý bude zhromažďovať používateľské recenzie videohier z distribučných platforiem. Tie následne analyzuje na úrovni aspektu a používateľom poskytne prehľadnú vizualizáciu výsledkov analýzy.

Kapitola 2 rozoberá problém získavania dát, analýzu sentimentu a neskôr sa zameriava na problematiku aspektovo-orientovanej analýzy sentimentu. Záver tejto kapitoly poskytuje prehľad najmodernejších riešení v uvedenej oblasti. Kapitola 3 predstavuje návrh riešenia a kapitola 4 dokumentuje použité technológie a dôležité časti implementácie. Použitelnosť systému popisuje kapitola 5. V závere práce sú zhodnotené dosiahnuté výsledky a načrtnuté možnosti zlepšenia vytvoreného systému.

¹<https://store.steampowered.com/>

Kapitola 2

Teoretický rozbor

V tejto kapitole sú obsiahnuté dôležité znalosti z oblastí, ktoré sú základnými stavebnými prvkami systému. Hlavným cieľom práce je vytvoriť systém určený na aspektovo-orientovanú analýzu videoherných recenzií.

Prvé sekcie sa zaoberajú procesom získavania dát, a to vrátane techník na ich extrakciu z rôznych zdrojov. Nasledujúce sekcie rozoberajú analýzu sentimentu s bližším zameraním na aspektovo-orientovanú analýzu na úrovni vety, ktorá nám umožňuje určiť, aké konkrétne prvky textu vzbudzujú pozitívne alebo negatívne emócie. Záverečné sekcie rozoberajú použité techniky strojového učenia v oblasti aspektovo-orientovanej analýzy.

2.1 Získavanie dát

Podstatnou časťou systému sú dáta, a preto je dôležité aj ich získavanie z internetových zdrojov. Zdrojom video herných recenzií na internete sú ich distribučné platformy, online obchody, video herné fóra alebo web stránky herných žurnálov. Pri získavaní dát z internetu narážame na mnoho technických a taktiež aj právnych prekážok, akými sú ochrana osobných údajov, autorské právo a podmienky používania služby. Preto je dôležité, aby sme pri návrhu a výbere zdrojov dbali aj na to, čo nám umožňujú extrahovať.

Medzi technické prekážky patrí:

- rôznorodosť poskytovaných dát,
- spôsob prezentácie dat (dáta vo forme obrázku napr. hodnotenie hry pri recenzií),
- obmedzený počet prístupov k web stránke pre IP adresu,
- obmedzenie prístupu k obsahu pre automatizované programy (napr. captcha verifikácia).

Z technických prekážok plyní dôležitosť voľby efektívnej techniky na identifikáciu, získavanie a čistenie dát z webových stránok.

Niektoré webové služby poskytujú aplikačné programové rozhranie (angl. Application Programming Interface – API) pre prístup k určitej časti svojich dát, čo do značnej miery uľahčuje ich získavanie.

Získavanie dát pomocou webových API

Aplikačné programové rozhranie (API) definuje súbor pravidiel, protokolov a nástrojov, s cieľom uľahčiť komunikáciu medzi softvérovými aplikáciami.

Webové API je špecifickým príkladom, kedy komunikácia medzi softvérovými aplikáciami prebieha cez internet, a to pomocou HTTP protokolu. Získavanie dát je skrz API jednoducho implementovateľné, keďže všetky potrebné informácie k práci s API webovej služby sú podrobne popísané v dokumentácii. Dokumentácia zvyčajne opisuje cesty alebo koncové body (URL adresy), na ktoré smerujú požiadavky s premennými parametrami, buď v ceste URL alebo ako parametre metódy GET HTTP protokolu [15].

Proces získavania dát cez webové API prebieha nasledovne:

- klient odošle požiadavku serveru, najčastejšie pomocou HTTP protokolu, na koncový bod služby, pričom požiadavka musí spĺňať základné pravidlá komunikácie určené dokumentáciou API,
- server odpovie na požiadavku vo formáte špecifikovanom dokumentáciou, najčastejšie sa používa formát eXtensible Markup Language (XML) a JavaScript Object Notation (JSON),
- klient spracuje odpoveď servera, ktorej obsah je štrukturovaný, a uloží potrebné dáta.

Nevýhodou tohto spôsobu získavania dát je to, že služba nemusí poskytovať všetky dáta, ktoré sú v oblasti nášho záujmu. Taktiež môže nastať situácia, kedy budeme musieť dáta súvisiace s jedným konkrétnym objektom získavať z viacerých koncových bodov, a tak treba zvážiť, či nám tento spôsob zjednoduší prácu alebo naopak pridá nadbytočnú prácu oproti metóde web scraping.

Web scraping

Web scraping je proces získavania štrukturovaných dát z webových stránok. Tento proces spočíva v extrakcii dát dostupných na webových stránkach a ich následného spracovania do štrukturovanej podoby. Web scraping môže byť realizovaný manuálne alebo automaticky, pomocou programu.

Programy určené k web scraping-u zdieľajú tieto základné podúlohy [7]:

- **Prístup na stránku:** program komunikuje s cieľovou webovou stránkou prostredníctvom HTTP protokolu. Pri komunikácii sa takmer výlučne používa metóda *GET*, ktorá je určená k žiadostiam o zdroje. Metóda *POST* sa využíva za účelom autentifikácie, v prípadoch, kedy je prístup k obsahu povolený len autorizovaným klientom. Dôležitou hlavičkou žiadosti je *User-Agent*, pretože na základe nej server zisťuje, aký druh programu pristupuje k jeho obsahu (prehliadač versus program). Okrem toho sa od webových zberačov dát, očakáva, že sa budú riadiť podmienkami použitia služby, ako je uvedené v súbore *robots.txt* (súbor uložený na serveri, ktorý uvádza, ktoré zdroje by nemali byť prístupné automatizovanými programom).
- **Spracovanie HTML a extrakcia obsahu:** po stiahnutí HTML dokumentu scraper dát môže extrahovať obsah, ktorý ho zaujíma. Na tento účel sa často používa zhoda regulárnych výrazov, samostatne alebo v kombinácii s ďalšími metódami. Ako alternatívy existujú knižnice na spracovanie HTML (pracujúce nad štruktúrou Document Object Model webových stránok) a jazyky založené na výbere konkrétneho

elementu na základe cesty, akými sú XPath¹ a syntax CSS selektorov. Odporúča sa používať čo najvšeobecnejšie výrazy, ktoré označujú požadovaný obsah, aby boli programy menej náchylné na zmeny v HTML dokumente. Web scraper je taktiež odolnejší, ak stránka implementuje sémantické značkovanie, ako napríklad mikroformáty² alebo mikrodáta³.

- **Spracovanie výstupu:** hlavným cieľom je premeniť extrahovaný obsah na štruktúrované dáta, ktoré sú vhodné na ďalšiu analýzu alebo uloženie. Niektoré nástroje poskytujú dátové štruktúry a textové riešenia, akými sú napríklad textové reťazce alebo súbory (typicky XML alebo CSV súbory).

Existuje niekoľko druhov web scraping systémov, ktoré sa líšia svojou robustnosťou, výkonom, spôsobom prehľadávania webu a dizajnom, ktorý zodpovedá konkrétnej doméne použitia. Ďalej sa bude táto sekcia venovať najbežnejším z nich.

Jednoduché scrapery založené na princípe požiadavka-odpoveď

Najjednoduchší spôsob web scrapingu spočíva v odosielaní HTTP požiadaviek na cieľovú webovú stránku, spracovaní HTML odpovede a extrakcii požadovaných dát. Tento prístup je najvhodnejší pre statické webové stránky s jednoduchou štruktúrou. Scrapery založené na tomto princípe sú ľahko implementovateľné a udržiavateľné, ale môžu mať problémy s komplexnejšími alebo dynamickými webovými stránkami.

Scrapery s automatizáciou prehliadača

Niektoré webové stránky sa spoliehajú na JavaScript a dynamické načítanie alebo zobrazenie obsahu, čo komplikuje prístup jednoduchých scraperov k informáciám. V takýchto prípadoch môže byť použitý scraper s automatizáciou prehliadača, ktorý napodobňuje interakciu používateľa s webstránkou skrz webový prehliadač. Tento prístup je náročnejší na zdroje, ale dokáže lepšie zvládnuť dynamický obsah.

Webové pavúky

Pre rozsiahle web scraping projekty, ktoré zahŕňajú prechádzanie viacerých stránok alebo celých webových stránok, je vhodnejší prístup pomocou webových pavúkov (angl. web crawler alebo spider). Tieto systémy prehľadávajú webové stránky tým, že nasledujú každý nájdený odkaz a získavajú údaje z každej navštívenej stránky. Počiatočná stránka, sa nazýva seed page, z ktorej obsahu sa získajú prvotné webové adresy na prehľadávanie. Pri tomto dizajne je nutné implementovať stratégie na výber vhodných URL odkazov, ukládanie zatiaľ nespracovaných URL odkazov a ich prehľadávanie. Webové pavúky sú zložitejšie, ale dokážu získavať veľké množstvo dát.

Distribuované web scraping systémy

Distribuované web scraping systémy využívajú viacero strojov alebo procesov na súčasné získavanie údajov. Tento dizajn je užitočný pre rozsiahle projekty, ktoré vyžadujú vysokú

¹<http://www.w3.org/TR/xpath20>

²<http://microformats.org>

³<http://www.whatwg.org>

priepustnosť, alebo keď jediný stroj nedokáže zvládnuť zaťaženie. Distribuované systémy taktiež umožňujú obísť opatrenia súvisiace s obmedzovaním rýchlosti a znížiť pravdepodobnosť blokovania zo strany cieľových webových stránok.

Inteligentné a adaptívne scrapery

Niektoré web scraping systémy využívajú strojové učenie alebo iné pokročilé techniky na prispôsobenie sa meniacim sa štruktúram webových stránok alebo na spracovanie obsahu webových stránok, ktoré používajú opatrenia s cieľom zamedziť prístup automatizovaným systémom. Tieto inteligentné systémy dokážu autonómne rozpoznávať a extrahovať relevantné údaje z webových stránok, aj keď sa zmení základná štruktúra, alebo keď čelia výzvam, ako sú CAPTCHA alebo sledovanie pomocou súborov cookies⁴.

Vyber vhodného dizajnu systému závisí od konkrétnych požiadaviek a výziev projektu. Mali by sa zohľadniť faktory ako komplexnosť a veľkosť potrebného obsahu webovej stránky, prítomnosť dynamického obsahu a potreba škálovateľnosti a adaptívnosti.

2.2 Analýza sentimentu

Analýza sentimentu je oblasť výskumu zameriavajúca sa na identifikáciu a extrakciu polaritu názorov alebo emócií z textu. Jej cieľom je zistiť, či je text pozitívny, negatívny alebo neutrálny vo vzťahu k danej téme, aspektu alebo entite [11].

K analýze sa najčastejšie používajú metódy založené na pravidlách alebo strojvom učení [14].

Metódy založené na pravidlách zahŕňajú použitie slovníkov slov s predom definovanými hodnotami sentimentu a pravidlami na ich kombinovanie. Ich nevýhodou je vysoká náročnosť na manuálnu prácu a nižšia úspešnosť pri aplikovaní na určité domény.

Strojové učenie zahŕňa tréning modelov, akými sú napríklad Naivný Bayesov klasifikátor, Support Vector Machines (SVM) alebo transformer modely, na špecifických dátových sadách [34].

Úrovně analýzy sentimentu

Analýze sentimentu sa môžeme venovať na troch úrovniach: úrovni dokumentu, vety a úrovni entity a aspektu [11].

Úroveň dokumentu — úlohou na tejto úrovni je klasifikovať sentiment vyjadrený celým dokumentom. Napríklad pri hodnotení hry systém určí, či hodnotenie vyjadruje celkovo pozitívny alebo negatívny názor na danú hru. Táto úroveň analýzy predpokladá, že každý dokument vyjadruje názory na jednu entitu (napr. jednu hru, produkt...), preto nie je jej aplikácia vhodná na dokumenty, ktoré hodnotia alebo porovnávajú viaceré entity.

Úroveň vety — analýza na tejto úrovni spočíva v určení, či veta v dokumente vyjadruje pozitívny, negatívny alebo neutrálny názor. Neutrálny zvyčajne znamená žiadny názor. Táto úroveň analýzy súvisí aj s určovaním subjektivity, ktorá rozlišuje vety vyjadrujúce faktické informácie od viet vyjadrujúcej subjektívne názory a postoje. Treba však poznamenať, že aj objektívne vety môžu naznačovať určitý sentiment [27].

⁴<https://www.dataversity.net/developing-ai-based-solution-for-web-scraping-lessons-learned>

Úroveň entity a aspektu — analýza na úrovni aspektu je jemnejšia. Namiesto skúmania jazykových konštruktov (dokumentov, odsekov, viet, klauzúl alebo fráz) sa úroveň aspektu priamo zameriava na názor na cieľovú entitu alebo aspekt samotný.

2.3 Aspektovo-orientovaná analýza sentimentu

Klasifikácia sentimentu na úrovni dokumentu alebo vety analyzuje všeobecný sentiment, ktorý je vyjadrený v danom texte, ale nedokáže zachytiť rozličné názory na špecifické aspekty entity, čo je napríklad pre analýzu recenzií nepostačujúce, keďže recenzia môže vyjadrovať protikladný sentiment voči rôznym aspektom alebo môže hovoriť o objektívnej vlastnosti. Analýza sentimentu na úrovni aspektov rieši tento nedostatok tým, že klasifikuje sentiment vo vzťahu ku konkrétnym aspektom entít.

Pojmom **aspekt** označujeme konkrétnu vlastnosť, atribút alebo časť entity (napríklad produktu alebo služby), na ktorú sa vzťahuje názor alebo hodnotenie. Napríklad vo vete „*Hra je super, no dĺžka kampane je príliš krátka, aspoň cena je dobrá.*“ je entitou konkrétna hra a jej aspektmi sú *dĺžka kampane* a *cena*. Sentiment pre *dĺžku kampane* je negatívny, zatiaľ čo sentiment pre *cenu* je pozitívny.

Tento prístup umožňuje komplexnejšie a presnejšie pochopenie názorov a je obzvlášť užitočný pre aplikácie, ktoré sa snažia poskytnúť bližší obraz o hodnotení entity [14].

V oblasti aspektovo-orientovanej analýzy bolo definovaných mnoho podúloh. Ich cieľom je extrahovať alebo klasifikovať určitú časť zo vzťahu medzi aspektom, entitou a názorom, vyjadrenými, či už explicitne v texte alebo implicitne vyplývajúcich z kontextu. Je vhodné si definovať ďalšie používané termíny.

Aspektový výraz (angl. aspect term) je slovo alebo fráza, ktorá označuje konkrétny atribút alebo vlastnosť entity (napríklad produktu alebo služby), spomínanej v texte. Napríklad vo vete „*Životnosť batérie telefónu je úžasná*“ je aspektovým výrazom „*životnosť batérie*“. Taktiež umožňuje identifikovať, na ktorú časť alebo vlastnosť sa vyjadrený názor vzťahuje.

Kategória aspektu (angl. aspect category) je širšia preddefinovaná skupina, ktorá zahŕňa rôzne aspekty, vlastnosti alebo časti entity. Narozdiel od aspektových výrazov sa kategória v texte nemusí priamo spomínať, ale môže vyplývať z kontextu (byť implicitne vyjadrená). Napríklad pri hodnotení hry môžu byť kategórie aspektov takéto: hrateľnosť (ovládanie, obsah, dĺžka kampane), kvalita (systémové požiadavky, chyby, stabilita), prostredie (atmosféra, dizajn sveta), cena (hodnota za peniaze, cenová úroveň).

Názorový výraz (angl. opinion term) je slovo alebo fráza, ktorá vyjadruje pozitívny, negatívny alebo neutrálny postoj autora voči konkrétnemu aspektu, entite alebo téme. Názorové výrazy sú dôležité pri analýze sentimentu, pretože pomáhajú identifikovať a zhodnotiť emocionálny tón (názor) vyjadrený v texte. Príkladmi môžu byť slová ako „vynikajúci“, „úžasný“ (pozitívne), „zlyhanie“ (negatívne) alebo „priemerný“, „neutrálny“ (neutrálne).

Polarita sentimentu (angl. sentiment polarity) predstavuje emocionálny náboj alebo hodnotenie, ktoré je pripísané aspektu. Polarita sentimentu je zvyčajne rozdelená do troch základných kategórií: pozitívna, negatívna a neutrálna.

Extrakcia aspektu

Extrakcia aspektu sa zvyčajne rozdeľuje na extrakciu aspektového výrazu a detekciu kategórie aspektu.

Extrakcia aspektového výrazu

Extrakcia aspektového výrazu (angl. Aspect Term Extraction (ATE)), taktiež nazývaná extrakcia názorového cieľa [20], má za cieľ z textu extrahovať slová alebo frázy vyjadrujúce aspekty určitej entity [19].

Nech $d = (s_1, s_2, \dots, s_n)$ je daný dokument s n vetami. Nech veta $s_i = (w_1, w_2, \dots, w_m)$, je i -tá veta dokumentu s m slovami. Potom cieľom je nájsť množinu aspektových výrazov $A_i = \{a_1, a_2, \dots, a_o\}$, kde každé a_k je podmnožinou slov vo vete s_i , ktoré reprezentujú určitý aspektový výraz.

Detekcia kategórie aspektu

Detekcia kategórie aspektu (angl. Aspect Category Detection (ACD)) zaraďuje aspekt, na ktorý bol v texte vyjadrený názor, do širšej skupiny, ktorá má význam pre konkrétnu úlohu alebo doménu. Množina kategórií aspektov je, narozdiel od aspektových výrazov, vopred definovaná. Kategóriami na doméne videohier môžu byť napríklad hrateľnosť, kvalita, prostredie, cena atď. [30].

Nech $d = (s_1, s_2, \dots, s_n)$ je daný dokument s n vetami a nech C je množina preddefinovaných kategórií aspektov. Úlohou detekcie kategórie aspektov je pre každú vetu s v dokumente d , určiť množinu kategórií aspektov $K \subseteq C$, spomenutých vo vete s .

Extrakcia názoru

Extrakcia názoru je úloha, ktorej cieľom je identifikovať slová alebo frázy v texte, ktoré vyjadrujú subjektívne hodnotenie alebo stanovisko vo vzťahu k aspektom (entitám a ich vlastnostiam) v danom texte.

Formálne môžeme definovať úlohu extrakcie názorov nasledovne:

Nech $s = (w_1, \dots, w_n)$ je veta s n slovami. Úlohou extrakcie názorov je nájsť množinu názorov $O = \{o_1, \dots, o_m\}$, kde každý názor o_i je podmnožinou slov vo vete s , ktorá vyjadruje subjektívne hodnotenie alebo stanovisko vo vzťahu k aspektom (aspektovým výrazom a kategórií) v s .

Extrahované názory je ešte potrebné priradiť k prislúchajúcim aspektom, na ktoré sú názory cielené.

Klasifikácia polarity sentimentu

Cieľom klasifikácie polarity sentimentu (angl. Aspect Sentiment Classification (ASC)) je predpovedať polaritu sentimentu vybraného aspektu.

Nech sekvencia slov kontextu $w^c = (w_1, w_2, \dots, w_n)$. Nech aspektová sekvencia $w^a = (w_i, \dots, w_j)$, pričom $1 \leq i \leq j \leq n$. Aspektová sekvencia w^a je podsekvenciou kontextovej sekvencie w^c . Potom cieľom úlohy ASC je predpovedať polaritu sentimentu $y \in \{\textit{positive}, \textit{negative}, \textit{neutral}\}$ pre aspekt určený sekvenciou w^a v kontexte w^c .

Pri ASC sa stretávame s niekoľkými problémami. Jedným z nich je kontextová závislosť orientácie polarity sentimentu. Tento problém do značnej miery ovplyvňuje aj moderné prístupy založené na hlbokých neurónových sieťach a mechanizme pozornosti a okliesňuje

doménu použitia týchto riešení ([28][16][13]). Na doméne recenzií videohier sa tento problém vyskytuje napríklad pri recenziách hororových hier. Hráči vyjadrujú svoju spokojnosť s hororovou hrou tým, ako veľmi ich daná hra dokáže vystrašiť, no slová, ktoré evokujú strach, majú pri väčšine hier negatívny sentiment.

Spoločná extrakcia

V kontexte aspektovo orientovanej analýzy sentimentu sa termín spoločná extrakcia (angl. joint extraction) používa na označenie prístupu, pri ktorom sa súčasne extrahujú alebo identifikujú viaceré prvky alebo atribúty vstupného textu. Tento prístup je založený na domnienke, že spoločná extrakcia viacerých zložiek z textu môže zlepšiť výsledky vzhľadom na to, že rôzne zložky sú často prepojené a závislé od seba.

Formálne môžeme spoločnú extrakciu definovať ako funkciu $\phi(\mathbf{R})$, ktorá zobrazuje vstupnú sekvenciu slov $\mathbf{R} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)$ na množinu extrahovaných zložiek, pričom každá zložka C_i obsahuje prvky z rôznych domén D_1, D_2, \dots, D_k

$$\phi(R) = \{C_1, C_2, \dots, C_m\},$$

kde

$$C_i = \{d_{i1} \in D_1, d_{i2} \in D_2, \dots, d_{ik} \in D_k\}.$$

V prípade úlohy Aspect-Category-Opinion-Sentiment Quadruple Extraction (ACOS) domény D_1, D_2, D_3, D_4 zodpovedajú aspektom, kategóriám, názorom a hodnotám sentimentu. Spoločná extrakcia potom spočíva v identifikácii štvoric $(\mathbf{a}, \mathbf{c}, \mathbf{o}, \mathbf{s})$, kde $\mathbf{a} \in D_1$ je aspekt, $\mathbf{c} \in D_2$ je kategória, $\mathbf{o} \in D_3$ je názor a $\mathbf{s} \in D_4$ je sentiment.

2.4 Riešenia založené na hlbokom učení

Hlboké učenie je významnou oblasťou strojového učenia, ktorá vyniká v modelovaní zložitých vzťahov a abstrakcií obsiahnutých v prirodzenom texte. Hlavným kľúčom k úspechu hlbokého učenia je použitie umelých neurónových sietí s mnohými vrstvami, ktoré modelom umožňujú naučiť sa mnoho úrovni abstrakcie na danej dátovej sade. Najpoužívanejšími prístupmi v hlbokom učení sú konvolučné neurónové siete (CNN), rekurentné neurónové siete (RNN) a siete založené na architektúre typu transformer.

Dosiahnutie uspokojivých výsledkov pomocou hlbokých neurónových sietí si vyžaduje veľké množstvo dát a taktiež je náročné na výpočetné zdroje pri procese tréningu. Z tohto dôvodu je veľmi využívanou a podstatnou technikou v tejto oblasti metóda **učenia s prenosom** (angl. transfer learning). Je to technika založená na princípe prenosu znalosti z jednej úlohy a aplikovanie týchto znalostí na druhú príbuznú úlohu. Za účelom modelovania určitej úlohy sa ako počiatočný bod zvolí model s už natrénovanými váhami príznakov na inej príbuznej úlohe, čo vedie k značnému urýchleniu tréningu a zlepšeniu vlastností zovšeobecňovania ([8][23]). K tomu, aby bola táto metóda použiteľná na prenos znalosti z jednej úlohy na inú, musia mať naučené príznaky modelu všeobecný charakter vzhľadom k východze a cieľovej úlohe, čo znamená, že nemôžu byť špecifické len pre prvotnú úlohu [33].

V kontexte spracovania prirodzeného jazyka hlboké učenie prináša revolučné zlepšenia v rôznych úlohách, ako sú strojový preklad, rozpoznávanie mien, syntaktická analýza, generovanie textu a analýza sentimentu.

V oblasti aspektovo orientovanej analýzy sentimentu, hlboké učenie prináša niekoľko metód a s nimi spojených výhod. Jedným z prístupov je použitie rekurentných neurónových sietí, ako sú LSTM (Long Short-Term Memory) alebo GRU (Gated Recurrent Units), ktoré sú schopné modelovať sekvenčnú povahu textu a zachytiť závislosti medzi slovami. Tieto modely sa dajú použiť na extrakciu aspektov a názorov, ako aj na klasifikáciu sentimentu ([26][17]).

Ďalším prístupom je použitie konvolučných neurónových sietí, ktoré sú schopné identifikovať lokálne štruktúry v texte. Tieto modely môžu byť užitočné pri identifikácii aspektov a názorov, pretože umožňujú zohľadniť kontext slov a identifikovať dôležité n-gramy (sekvencie slov alebo znakov). Výhodou použitia konvolučných neurónových sietí oproti LSTM je najmä možnosť paralelizácie výpočtu pre množinu vstupov, čo pri LSTM nie je možné [31]. Na druhej strane sú ich schopnosti v oblasti zachytenia vzdialených závislostí pre dlhé vstupné sekvencie limitované oproti LSTM a to kvôli veľkosti vnemového poľa jednotlivých vrstiev konvulčnej siete. Tomuto problému sa však dá do určitej miery pomôcť napríklad zväčšením hĺbky siete [1].

Najnovším a v súčasnosti najpoužívanejším prístupom je použitie siete typu transformer, ktoré sú založené na mechanizme vlastnej pozornosti (self-attention). Tieto siete sú schopné efektívne zachytiť závislosti aj medzi vzdialenými časťami v texte a poskytovať bohatú reprezentáciu slov v prislúchajúcom kontexte. Modely, založené na architektúre transformer, akými sú BERT [5] alebo T5 [21], predtrénované na veľkých slovníkoch, sa dajú ďalej doladiť na špecifické úlohy, kde využívajú bohaté znalosti prirodzeného jazyka a tak zlepšujú výsledky na týchto úlohách.

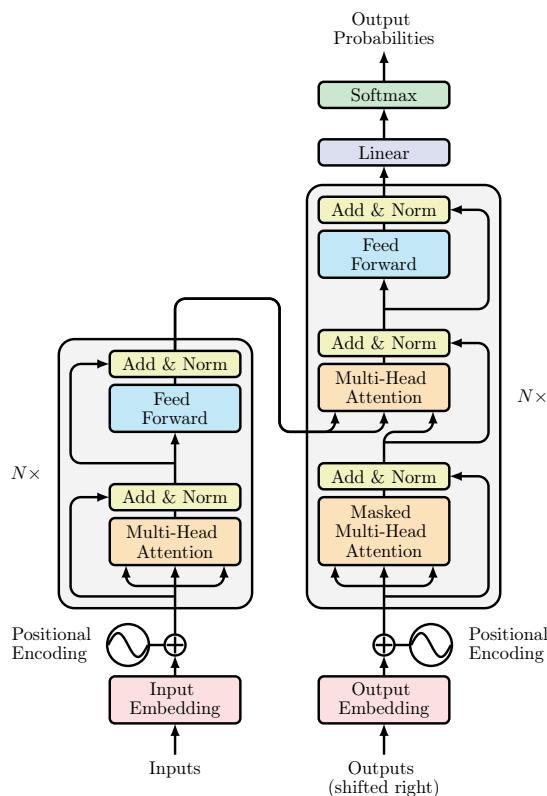
2.5 Neurónová sieť typu transformer

Neurónová sieť typu transformer bola prvýkrát predstavená autormi *Vaswani a spol.* v článku *Attention Is All You Need* [25]. Je založená na mechanizme nazývanom self-attention (vlastná pozornosť), kódovacích a dekódovacích blokoch a eliminuje potrebu tradičných rekurentných a konvulčných vrstiev, čím taktiež umožňuje efektívne spracovávať dlhé sekvenencie a paralelizovať výpočet.

Spracovanie vstupu do vektorovej reprezentácie

Na začiatku transformácie sa každý vstupný text rozdelí pomocou tokenizátora na sekvenciu slov alebo podslovných jednotiek, označovaných tokeny. V tomto kroku sú tokeny reprezentované celočíselnou hodnotou, ktorá je identifikátorom daného tokena v slovníku použitého tokenizátora. Každý vstupný token je ďalej embedding vrstvou prevedený na svoju vektorovú reprezentáciu (word embedding), s ktorou pracujú ostatné vrstvy modelu.

Word embedding je metóda, pomocou ktorej je možné ku každému slovu z určitého konečného slovníka (množina unikátnych tokenov, s ktorými sa model pri predtrénovaní na určitej dátovej sade stretne) priradiť jeho vektorovú reprezentáciu, ktorá zachytáva sémantické a syntaktické vlastnosti daného slova. Výhodou tejto reprezentácie slov je to, že vo vektorovom priestore generovanom týmito vektormi sú si blízke práve tie vektory, ktoré reprezentujú slová s podobným významom alebo použitím v jazyku. Embedding vrstva je potom tá časť, ktorá vykonáva prevod vstupov (tokenov) na vektorovú reprezentáciu vhodnú pre spracovanie ďalšími vrstvami. Táto vrstva preto zohráva kľúčovú úlohu v modeloch strojového učenia zaoberajúcich sa spracovaním prirodzeného jazyka, akými sú aj transformery. Pri spätnom prevode z vektorovej reprezentácie na výstupné tokeny je tak-



Obr. 2.1: Schéma architektúry transformer predstavenej v práci *Attention Is All You Need* [25].

tiež potrebná embedding vrstva a preto sa často využíva len jedna váhová matica pre obe vrstvy.

Polohové kódovania

Vo vektorových reprezentáciách slov chýba informácia o polohe slova, ktoré reprezentujú. Keďže transformery pracujú s paralelným spracovaním sekvencií, potrebujú nejaký spôsob, ako získať informácie o relatívnych alebo absolútnych pozíciách slov vo vstupných dátach.

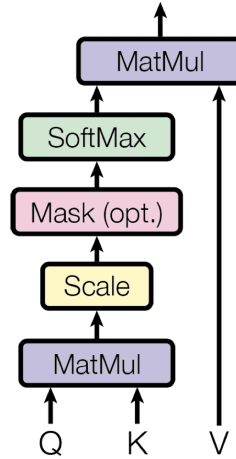
Kódovania pozícií sa vytvárajú ako vektory, ktoré reprezentujú pozíciu každého slova v sekvencii. Tieto vektory sú potom sčítané s vektorovými reprezentáciami slov. Výsledné kombinované vektory obsahujú informácie o slovách aj o ich pozíciách.

Existuje niekoľko spôsobov, ako kódovať polohu slova vo vstupnej sekvencii. V pôvodnej architektúre transformer *Vaswanih a spol.* navrhujú polohové kódovania vytvoriť pomocou sinusoidálnych funkcií s rôznymi frekvenciami, ktoré umožňujú modelu interpolovať polohovú informáciu pre sekvencie rôznych dĺžok, narozdiel od naučených polohových kódovaní aj takých, s ktorými sa model pri tréňovaní nestretol.

Pozornosť

Mechanizmus pozornosti je mechanizmus, ktorý bol navrhnutý pre neurónové siete na zlepšenie ich schopnosti modelovať závislosti medzi vstupmi a výstupmi sekvencií. Mechanizmus pozornosti sa zameriava na váženie dôležitosti rôznych častí vstupnej sekvencie pri gene-

rovaní výstupu. Tento prístup umožňuje neurónovým sieťam získať relevantné informácie z rôznych častí vstupnej sekvencie a efektívne sa učiť vzdialené závislosti medzi jednotlivými časťami vstupu [3].



Obr. 2.2: Výpočet pozornosti pomocou skalárneho súčinu matíc (scaled dot-product attention) reprezentujúcich dotazy (Q), kľúče (K) a hodnoty (V). Pri výpočte je možné použiť masku, čo je hlavným rozdielom medzi pozornosťou kódovacieho bloku, ktorý masku nepoužíva a dekódovacieho bloku, kde sa maska používa (prevzaté z [25]).

Vlastná pozornosť

Vlastná pozornosť (angl. self-attention) je špeciálny prípad mechanizmu pozornosti, kde vstupná sekvencia a výstupná sekvencia sú rovnaké. V kontexte transformera, vlastná pozornosť umožňuje modelu získať informácie o vzťahoch medzi rôznymi slovami v rámci jednej sekvencie. To znamená, že model sa učí ako priradiť rôzne váhy pozornosti jednotlivým slovám v sekvencii, aby bol schopný lepšie pochopiť kontext a závislosť medzi slovami.

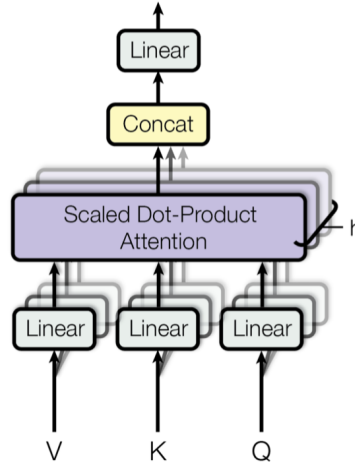
Výpočet pozornosti prebieha nasledovne. Pre sekvenciu vstupných vektorov, ktorú reprezentuje matica vstupu $X \in \mathbb{R}^{N \times d_{\text{model}}}$, sa vytvoria matice dotazov Q , kľúčov K a hodnôt V s rozmerom $N \times d_k$. Tieto matice sú získané násobením matice vstupu X a váhových matíc lineárnych vrstiev W^Q , W^K , W^V . Váhové matice majú rozmer $d_{\text{model}} \times d_k$. Symbol N predstavuje dĺžku vstupnej sekvencie, d_{model} je veľkosť vektorovej reprezentácie slova a d_k je hyperparameter modelu a jeho hodnota je zvyčajne d_{model}/h , kde h je počet hláv pri viachlavovej pozornosti. Skóre pozornosti sa vypočíta ako skalárny súčin medzi maticami dotazov a kľúčov, pričom sa normalizuje delením súčinu odmocninou z d_k . Skalárny súčin poskytuje informáciu o tom, ako veľmi sú jednotlivé slová v sekvencii relevantné vo vzťahu k ostatným slovám.

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V, \quad (2.1)$$

kde Q , K a V sú matice dotazov, kľúčov a hodnôt.

Po vypočítaní skóre pozornosti sa aplikuje aktivačná funkcia softmax, ktorá zabezpečuje, že všetky hodnoty sú nezáporné a ležia na intervale (0,1). Nakoniec sa normalizovaná matica

skóre pozornosti vynásobí s maticou hodnôt, čím sa získajú váhované hodnoty pre každé slovo v sekvencii.



Obr. 2.3: Schéma reprezentujúca mechanizmus viachlavovej pozornosti (prevzaté z [25]).

Viachlavová pozornosť

Viachlavová pozornosť (angl. multi-head attention) je rozšírením mechanizmu self-attention, ktorý umožňuje modelu súčasne spracovať viaceré aspekty kontextu a vzťahov medzi slovami vstupnej sekvencie. Pre lepšie pochopenie si môžeme multi-head attention predstaviť ako skupinu paralelných pozorností pracujúcich na rôznych aspektoch vstupných dát.

Pre výpočet potrebujeme h lineárnych vrstiev pre dotazy, kľúče a hodnoty, kde h je počet hláv. Tieto matice majú rozmery $d_{model} \times d_k$, kde d_{model} je veľkosť vstupného vektora a d_k je hyperparameter modelu. Pre každú z h hláv sa vypočíta self-attention nezávisle, podľa vzorca uvedeného vyššie 2.1.

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O, \quad (2.2)$$

kde

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.3)$$

pre $i = 1, \dots, h$, a W_i^Q, W_i^K, W_i^V sú váhové matice pre dotazy, kľúče a hodnoty pre i -tú hlavu. $W^O \in \mathbb{R}^{hd_k \times d_{model}}$ je ďalšia váhová matica, ktorá kombinuje výstupy jednotlivých hláv. Výstupy z jednotlivých hláv sa zrefázia (Concat) a potom sa transformujú pomocou ďalšej lineárnej vrstvy (W^O), čím sa získajú výsledné váhované hodnoty pre každé slovo v sekvencii.

Vrstvová normalizácia

Vrstvová normalizácia je dôležitým nástrojom v hlbokom učení, ktorý zlepšuje trénovanie neurónových sietí [2]. Hlavným cieľom vrstvovej normalizácie, podobne ako aj iných typov normalizácie [9], je zlepšenie konvergence a stabilizácia učenia. Pri trénovaní hlbokých neurónových sietí môžu nastať problémy spojené s explodujúcimi alebo miznúcimi gradientami. Tieto problémy môžu spôsobiť, že sieť sa bude učiť pomaly. Vrstvová normalizácia rieši tieto problémy tým, že normalizuje aktivačné hodnoty, čo zabraňuje veľkým

hodnotám aktivačných funkcií ovplyvniť tréning. Normalizácia úplne nezabráni týmto problémom, no znižuje pravdepodobnosť ich výskytu. Týmto spôsobom sa zvyšuje stabilita učenia a zlepšuje konvergencia modelu.

Pre každú vzorku x_i (v kontexte transformer architektúry je to vektor o veľkosti d_{model}) z dávky B o veľkosti m , $B = x_1, x_2, \dots, x_m$, vrstvomá normalizácia prebieha nasledovne:

Vypočítame priemer μ_i a variáciu σ_i^2 pre každú vzorku x_i :

$$\mu_i = \frac{1}{K} \sum_{k=1}^K x_{i,k}, \quad (2.4)$$

$$\sigma_i^2 = \frac{1}{K} \sum_{k=1}^K (x_{i,k} - \mu_i)^2, \quad (2.5)$$

kde K je počet prvkov vo vzorke x_i .

Každá vzorka sa normalizuje tak, aby jej prvky mali nulový priemer a jednotkovú variáciu, s použitím stabilizačného člena ϵ , ktorý je potrebný hlavne pre hodnoty blízke nule:

$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \quad \hat{x}_{i,k} \in \mathbb{R} \quad (2.6)$$

Škálovanie a posun normalizovaných prvkov pomocou naučiteľných parametrov γ a β :

$$y_i = \gamma \hat{x}_i + \beta \equiv \text{LN}_{\gamma, \beta}(x_i), \quad x_i \in \mathbb{R}^k \quad (2.7)$$

kde γ a β sú naučiteľné parametre a $\text{LN}_{\gamma, \beta}(x_i)$ označuje výsledok vrstvomá normalizácie pre vzorku x_i .

Kódovací blok

Kódovací blok tvorí celkovo 6 identických vrstiev, kde každá vrstva pozostáva z dvoch podvrstiev: viachlavovej vlastnej pozornosti a doprednej plne prepojenej siete (position-wise feed-forward network).

Viachlavová vlastná pozornosť

Ako bolo vysvetlené v predošlej sekcii o pozornosti, používa sa na získanie informácií o vzťahoch medzi rôznymi slovami v rámci jednej sekvencie. Vstupná sekvencia sa prenáša do viachlavovej vlastnej pozornosti, kde sa vypočíta výstup pre každé slovo na základe kontextu poskytnutého ostatnými slovami v sekvencii.

Dopredná plne prepojená sieť

Táto sieť je navrhnutá tak, aby sa aplikovala na každý vektor výstupu z viachlavovej vlastnej pozornosti. V podstate sa jedná o dve plne prepojené vrstvy, ktoré sa aplikujú na každý vstupný vektor nezávisle na ostatných vstupoch. Prvá vrstva je aktivačná funkcia v podobe retifikovanej lineárnej jednotky (ReLU) a druhá vrstva je lineárna transformácia, ktorá zmení rozmer vstupného vektoru na požadovanú veľkosť výstupu.

V rámci kódovacieho bloku sa tieto dve časti spájajú do jedného celku. Výstup z viachlavovej vlastnej pozornosti sa sčíta s pôvodnými vstupnými vektormi (tzv. reziduálne prepojenie)

a následne sa normalizujú vrstvovou normalizáciou. Potom sa uplatní plne prepojená dopredná sieť na normalizovaný výstup. Výstup z plne prepojenej doprednej siete sa opäť sčíta s výstupom z viachlavovej vlastnej pozornosti a normalizuje vrstvovou normalizáciou. Tieto výstupy potom vstupujú do ďalšieho kódovacieho bloku alebo sa použijú ako vstupy do dekódovacej časti transformera.

Kódovací blok sa opakuje niekoľkokrát (zvyčajne 6 alebo 12-krát, v závislosti od konkrétnej implementácie transformera) na vytvorenie konečného výstupu. Opakovanie kódovacích blokov umožňuje modelu postupne zlepšovať reprezentáciu vstupnej sekvencie a extrahovať zložitejšie vzťahy medzi jednotlivými slovami.

Dekódovací blok

Dekódovací blok (angl. Decoder block) je ďalší základný stavebný prvok transformera. Štruktúra tohto bloku je podobná ako štruktúra kódovacieho bloku. Tvorí ho taktiež 6 identických vrstiev, ktoré sa skladajú z rovnakých podvrstiev ako vrstvy kódovacieho bloku, no obsahujú jednu dodatočnú podvrstvu - viachlavovú pozornosť medzi kódovanou a dekódovanou sekvenciou. Každá vrstva kódovacieho bloku sa teda skladá z: maskovanej viachlavovej vlastnej pozornosti, viachlavovej pozornosti medzi kódovanou a dekódovanou sekvenciou a doprednej plne prepojené siete.

Maskovaná viachlavová vlastná pozornosť

Rovnako ako v kódovacom bloku sa používa na získanie informácií o vzťahoch medzi rôznymi slovami v rámci jednej sekvencie, no od pozornosti kódovacieho bloku sa líši tým, že pozornosť sa kladie iba na predchádzajúce slova v sekvencii. Táto limitácia sa dosiahne tým, že sa aplikuje maska nad hodnotami získanými škálovaným skalárnym súčinom matíc Q a K ako je možné vidieť na obrázku 2.3.

Viachlavová pozornosť medzi kódovanou a dekódovanou sekvenciou

Táto podvrstva slúži na získanie informácií o vzťahu medzi dekódovanou sekvenciou a kódovanou sekvenciou. Pomocou tejto časti model dokáže získať informácie o tom, ako sa dekódovaná sekvencia viaže na pôvodnú vstupnú sekvenciu. Vstupom tejto vrstvy sú dotazy z predošlej vrstvy pozornosti dekódovacieho bloku a kľúče a hodnoty z výstupu kódovacieho bloku.

Dopredné plne prepojené siete

Rovnako ako v kódovacom bloku, sa aplikujú na každý vektor výstupu z druhej viachlavovej pozornosti. Prvá vrstva je aktivačná funkcia ReLU a druhá vrstva je lineárna transformácia, ktorá zmení rozmer vstupného vektoru na požadovanú veľkosť výstupu.

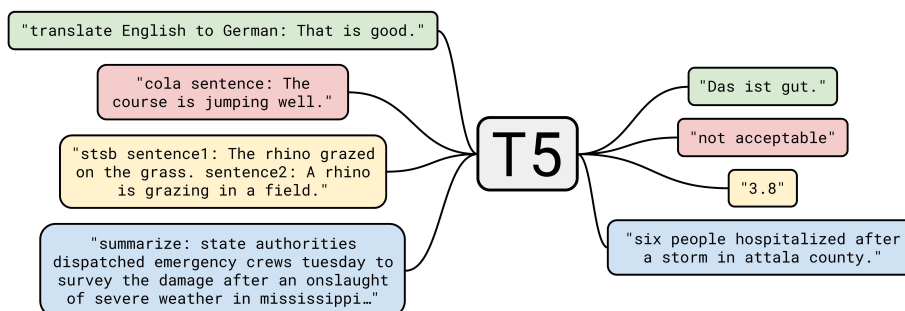
Dekódovací blok spája tieto tri časti do jedného výstupu. Výstupy z prvej viachlavovej vlastnej pozornosti sa sčítajú s pôvodnými vstupnými vektormi (reziduálne prepojenie) a následne sa normalizujú vrstvovou normalizáciou. Potom sa uplatní druhá viachlavová pozornosť, ktorá zohľadňuje kódovanú sekvenciu (výstup kódovacieho bloku). Výsledky tejto pozornosti sa opäť sčítajú s výstupmi z prvej viachlavovej pozornosti a normalizujú vrstvovou normalizáciou. Nakoniec sa uplatní dopredná plne prepojená sieť na tieto normalizované výstupy. Výsledky z plne prepojených sietí sa opäť sčítajú s výstupmi z druhej viachlavovej pozornosti a normalizujú vrstvovou normalizáciou. Tieto výstupy potom vstupujú do

ďalšieho dekodovacieho bloku alebo sa použijú ako konečný výstup dekodovacej časti transformera.

Podobne ako v prípade kódovacej časti, dekodovacie bloky sa opakujú niekoľkokrát (zvyčajne 6 alebo 12-krát, v závislosti od konkrétnej implementácie transformera) na vytvorenie konečného dekodovaného výstupu. Opakovanie dekodovacích blokov umožňuje modelu postupne zlepšovať reprezentáciu dekodovanej sekvencie a extrahovať zložitejšie vzťahy medzi jednotlivými slovami a pôvodnou vstupnou sekvenciou.

2.6 Text-to-Text Transfer Transformer

Text-to-Text Transfer Transformer (T5) je jazykový model, vyvinutý výskumnou skupinou Google Research [21], ktorý stavia na úspechoch predošlých modelov založených na architektúre transformer, tým, že zavádza jednotný text-to-text rámec pre viacúlohové učenie. Tento rámec umožňuje modelu T5 vykonávať rôzne úlohy z oblasti spracovania prirodzeného jazyka, zahrňujúce preklad, sumarizáciu textu, analýzu sentimentu, konvertovaním vstupného textu na výstupný text. Hlavnou myšlienkou je priniesť zmenu v prístupe trénovaní na jednotnej forme dát pre všetky úlohy a neprispôbovať architektúru konkrétnym úlohám s cieľom lepšie využiť učenie s prenosom znalostí. Dostupné predtrénované T5 modely boli trénované na veľkom množstve úloh, čo podľa autorov umocňuje ich schopnosť učenia a zovšeobecňovania naprieč rôznymi doménami.



Obr. 2.4: Schéma vstupov a výstupov modelu T5 pre viaceré úlohy v jednotnom formáte. Vstupy sa od seba líšia úvodným slovným popisom úlohy (prevzaté z [21]).

V kontexte aspektovo orientovanej analýzy má tento model veľký potenciál v identifikovaní aspektov na úrovni jednotlivých zložiek (výraz, názor, aspekt, sentiment). To vychádza z jeho schopnosti porozumieť komplexným jazykovým nuansám a inherentnej schopnosti riešiť viaceré podúlohy súčasne.

Kľúčovými vlastnosťami modelu sú transformer architektúra, možnosť viacúlohového učenia (angl. multitask learning) a jednotný formát z textu do textu pre celú škálu úloh prirodzeného jazyka. Tak ako aj ostatné transformer modely aj tento model využíva princíp mechanizmu vlastnej pozornosti, čo umožňuje zachytiť vzdialené závislosti a zložité vzťahy založené na kontexte vo vstupnom texte. Viacúlohový jednotný rámec uľahčuje proces spracovania dát a umožňuje viacúlohové učenie transformovaním jednotlivých úloh do jednotného formátu vstupných a výstupných sekvencií. Za účelom prevodu vstupného textu do celočíselnej reprezentácie využíva SentencePiece tokenizáciu, čo je prístup založený na delení textu na podslovné zložky, ktorý umožňuje spracovávať text viacerých jazykov. Je o niečo flexibilnejší ako bežne používané metódy tokenizácie vstupného textu. K vetám sa

pristupuje ako ku sekvenciám unicode znakov a neaplikuje sa žiadna jazykovo závislá logika, čo robí túto tokenizáciu jazykovo nezávislou.

2.7 V súčasnosti používané metódy

Jedným z najpoužívanejších prístupov pre extrakciu štvoric (aspekt, kategória, názor, sentiment) je sekvenčný prístup, ktorý postupne identifikuje rôzne zložky a ich vzťahy za použitia viacerých modelov. Napríklad, sa najprv identifikujú aspekty a ich kategórie, potom sa identifikujú názory a napokon sa identifikuje sentiment pre dané názory a aspekty. Z výsledkov štúdie, vykonanej *Hongjie Cai-m a spol.* [4], vychádza, že vhodným prístupom by mohla byť kombinovaná spoločná extrakcia aspektu a názoru a následná spoločná klasifikácia kategórie aspektu a sentimentu. Tento postup viedol v porovnaní s ostatnými štandardnými metódami k lepším výsledkom, no výhody spoločnej extrakcie pre úlohu extrakcie štvoric využíva len čiastočne a trpí akumuláciou chýb z jednotlivých podúloh.

Výhody spoločnej extrakcie spočívajú v tom, že model môže efektívne využiť spoločnú reprezentáciu jednotlivých zložiek, a tak lepšie zohľadniť vzájomné závislosti medzi nimi, čo môže viesť k presnejšiemu a robustnejšiemu modelu. Na druhej strane však rastie zložitosť modelu, a tým aj náročnosť tréningu a ladenia.

Nové riešenia v oblasti extrakcie štvoric ukazujú, že prístupy založené na generatívnych modeloch, umožňujú spoločnú extrakciu a prinášajú v súčasnosti najlepšie výsledky [18]. *Liu a spol.* [12] sa domnievajú, že týmto prístupom sa efektívnejšie využije schopnosť sumarizácie na sémantickej úrovni predtrénovaného modelu a to tým, že sa nepridávajú nové parametre určené na klasifikáciu, ktoré by bolo potrebné natrénovať. Z ich experimentálnych výsledkov tiež vyplýva, že tento prístup dosahuje lepšie výsledky na štandardných dátových sadách a má lepšie vlastnosti v oblasti učenia s niekoľkými príkladmi (few-shot) a bez príkladov (zero-shot). Tieto vlastnosti sú pri tejto úlohe veľmi dôležité, keďže je takmer nemožné zahrnúť v dátovej sade všetky možné spôsoby vyjadrenia názoru v prirodzenom jazyku. Nevýhodou je veľká závislosť na predtrénovaných modeloch, kde obzvlášť záleží na akých úlohách boli tréované.

Ďalšie štúdie v tejto oblasti ([22][6]) sa snažia vytvoriť jednotný rámec pre prístup k aspektovo orientovanej analýze s cieľom zlepšiť transferové vlastnosti výsledných generatívnych modelov medzi rôznymi doménami a podúlohami. Za týmto účelom definujú jednotný formát vstupu pre úlohu (angl. task prompt), zložený z viacerých menších častí (angl. prompt elements), z ktorých sa dá vyskladať formát vstupu pre rôzne podúlohy. Toto riešenie taktiež umožňuje využiť pri tréningu dátové sady, ktoré sú vytvorené len pre určité podúlohy.

Kapitola 3

Návrh riešenia

Táto kapitola dokumentuje proces návrhu a kroky pri tvorbe systému pre zber, ukladanie, analýzu a vizualizáciu výsledkov analýzy recenzií video hier a súvisiacich informácií. Cieľom tejto práce je navrhnúť a implementovať systém pre aspektovo-orientovanú analýzu sentimentu používateľských recenzií videohier. Systém by mal automatizovať proces získavania, spracovania a analýzy recenzií s dôrazom na identifikáciu sentimentu pre konkrétne kategórie aspektov hry, akými sú hrateľnosť, audiovizuálne prevedenie, komunita, optimalizácia, chyby a cena. Hlavným cieľom je poskytnúť hráčom, vývojárom a vydavateľom bohatší a podrobnejší prehľad o názoroch hráčov na rôzne aspekty videohier pomocou grafickej vizualizácie výsledkov analýzy a používateľského rozhrania webovej aplikácie.

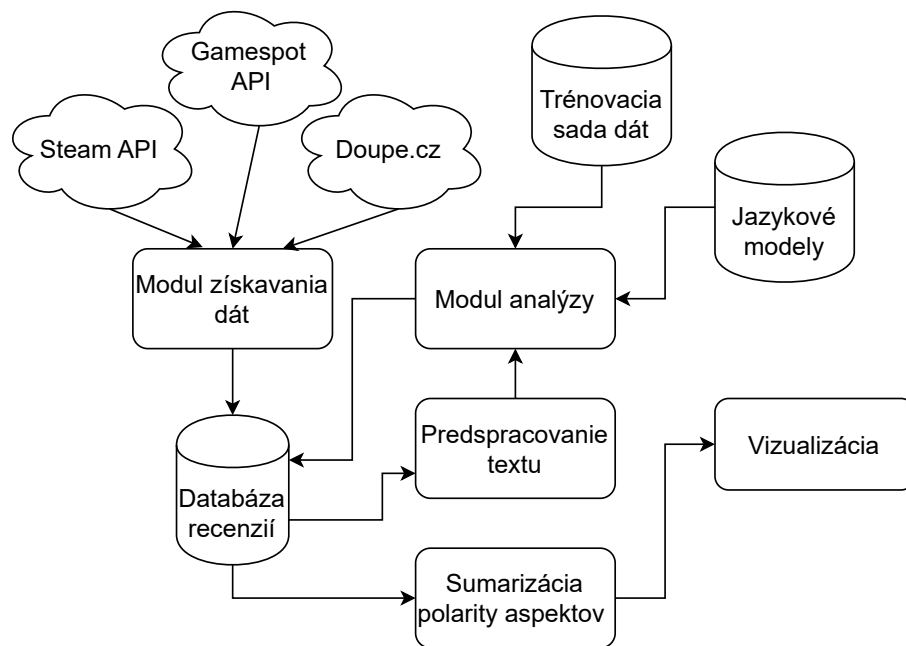
3.1 Návrh systému

Na základe cieľov práce bol navrhnutý systém, ktorý efektívne integruje rôzne moduly a komponenty potrebné na dosiahnutie stanovených cieľov. Architektúra je založená na samostatných moduloch, ktoré zabezpečujú funkcionality týchto častí systému: získavanie recenzií, analýza sentimentu, spracovanie dát a vizualizácia výsledkov. Tento prístup zaisťuje modularitu, rozšíriteľnosť a udržateľnosť systému.

3.2 Získavanie dát

Vzhľadom na to, že systém by nemal byť obmedzený na získavanie dát iba z jedného zdroja, je navrhnutý tak, aby získaval recenzie z rôznych zdrojov, pričom môžeme očakávať rôznorodosť extrahovaných údajov. Táto rôznorodosť údajov je zohľadnená v databázovom modeli, ktorý obsahuje viacero stĺpcov pre zaznamenávanie rôznych informácií, ktoré môžu byť získané z rôznych zdrojov recenzií. Medzi hlavné údaje, ktoré sú kľúčové pre fungovanie systému, patrí:

- **jedinečný identifikátor recenzie** – tento údaj umožňuje sledovať, ktoré recenzie sú už získané, čo je potrebné, aby sme zbytočne nezbierali duplikované dáta, a taktiež umožňuje znova nájsť danú recenziu v systéme zdroja. Pre recenzie získané z API zdroja je týmto údajom identifikátor v systéme zdroja a pre recenzie kritikov, ktoré boli získané metódou web scrapingu, je týmto údajom URL adresa recenzie, keďže každá recenzia má svoju jedinečnú web stránku.



Obr. 3.1: Návrh systému

- **text** – samotný text recenzie, ktorý je podstatný pre analýzu sentimentu založenú na aspektoch. Text recenzie je základom pre identifikáciu dôležitých aspektov a následné hodnotenie sentimentu pre jednotlivé aspekty.
- **jazyk recenzie** – tento údaj je dôležitý pre správne spracovanie textu a analýzu sentimentu. Je získaný buď priamo zo zdroja, alebo je vyvodенý z textu recenzie pri spracovaní recenzie.
- **dátum a čas vytvorenia recenzie** – táto informácia je dôležitá hlavne kvôli zvolenej vizualizácii, ktorá umožňuje používateľovi sledovať vývoj sentimentu pre jednotlivé kategórie aspektov a iných dát v čase.

Okrem hlavných údajov, som sa na základe existencie webových stránok žurnálov a recenzií kritikov, ktoré obsahujú viac štrukturovaných dát recenzie, rozhodol získavať aj skóre a vymenovanie dobrých a zlých aspektov hry. Zdrojmi dát pre tieto recenzie je web Doupe.cz¹ a Gamespot². Skóre je číselné hodnotenie videohry na základe recenzie na škále od 1 do 5, ktoré často predstavuje kvantitatívne vyjadrenie sentimentu kritika voči videohre. Skóre môže byť užitočné pri stanovení základného hodnotenia hry. Vymenovanie dobrých a zlých aspektov sa dá využiť pre vytvorenie sady dát pre sumarizáciu rozsiahlejších recenzií, čo umožní efektívnejšie spracovávať takýto typ recenzií.

Za hlavný zdroj dát som zvolil platformu Steam, ktorá poskytuje API rozhranie pre získavanie recenzií vytvorených používateľmi. Dôvodom voľby tejto platformy je fakt, že poskytuje najväčší celkový počet používateľmi vytvorených recenzií pre veľkú vzorku hier, čo zabezpečuje dostatok dát pre sumarizáciu výsledkov aspektovo-orientovanej analýzy [10].

¹<https://doupe.cz>

²<https://gamespot.com>

Modul pre získavanie dát

Tento modul je zodpovedný za získavanie recenzií videohier z rôznych zdrojov, ako sú webové stránky a API. Modul podporuje pravidelné a automatické aktualizácie datasetu recenzií, aby boli výsledky analýzy aktuálne a relevantné. Výstupom tohto modulu sú štrukturované dáta obsahujúce informácie o recenziách. Keďže je cieľom poskytnúť sumarizáciu výsledkov na úrovni jednotlivých videohier, tak spolu s recenziami získava informácie aj o jednotlivých hrách (napr. názov, vydavateľ, vývojár) a recenzentoch, čo taktiež umožňuje ďalšie rozšírenia úrovne analýzy a sumarizácie výsledkov. Získavanie dát má byť efektívne, a preto bol modul navrhnutý tak, aby bolo možné viacprocesové získavanie dát pre jednotlivé hry. Rýchlosť získavania dát pre hru zabezpečujú dávkové asynchrónne požiadavky a celková asynchrónna implementácia jednotlivých funkcií modulu. Pri návrhu modulu bol taktiež braný ohľad aj na všeobecnosť získavaných dát, a preto je možné modul rozšíriť o zdroje recenzií mimo domény videohier, a to implementáciou logiky získavania dát pre daný zdroj do definovaného rozhrania modulu. Rozhranie modulu obsahuje nasledujúce hlavné komponenty a metódy:

- inicializácia modulu – táto časť rozhrania zahŕňa parametre a nastavenia potrebné pre konfiguráciu modulu, ako sú URL adresy zdrojov pre jednotlivé dáta, formát očakávanej odpovede, autentifikačné údaje, nastavenie relácie atď. Inicializácia modulu bola navrhnutá tak, aby umožňovala jednoduché prispôbenie rôznym zdrojom dát a konkrétnym požiadavkám projektu.
- generátor recenzií – metóda, ktorá umožňuje postupné prechádzanie recenzií zo zdroja dát na úrovni stránok. Generátor je schopný zvládnuť rôzne veľkosti stránok a prispôbiť sa rôznym nastaveniam, aby bol efektívny a prispôsobivý pri získavaní recenzií z rôznych zdrojov. Pre správne fungovanie táto metóda očakáva, aby mal modul inicializovaný daný zdroj s URL adresou pre získavanie stránok recenzií a správne nastavenú metódu získavania (API alebo web scraping).
- získavanie detailov recenzie – metóda, ktorá získava podrobné informácie o konkrétnej recenzii, ako sú napríklad text recenzie, hodnotenie hry, identifikátor hry, identifikátor recenzenta atď.
- generácia stránok a získavanie detailov o hrách a recenzentoch – rozhranie obsahuje metódy pre získavanie detailných informácií o hrách a recenzentoch, akými sú napríklad názov hry, žáner, platforma, hodnotenie, identifikátor recenzenta a taktiež metódy, ktoré umožňujú postupné prechádzanie zoznamu hier alebo recenzentov zo zdroja dát. Tieto metódy umožňujú prepojenie medzi rôznymi typmi dát a analýzou sentimentu a sú špecifické pre implementáciu systému na doméne videohier, kde zabezpečujú lepšiu výslednú vizualizáciu a sumarizáciu.
- získavanie recenzií pre hru – metóda, ktorá umožňuje získať dáta pre vybranú hru a z vybraného zdroja.
- získavanie recenzií pre všetky hry, ktorých recenzie neboli aktualizované za určité časové obdobie – metóda, ktorá umožňuje spustenie pravidelného získavania recenzií.

Modul komunikuje s databázou pomocou vlastného konektora, ktorý sa stará o funkcionálnu a integritu pri vkladaní nových dát do databázy.

3.3 Spracovanie a analýza recenzií

Analýza recenzií je ďalšou dôležitou súčasťou systému, ktorej výsledky sú hlavnou podstatou systému. Po definovaní toho, čo bude potrebné z textu recenzií extrahovať a klasifikovať, aby výsledky poskytli podrobnejší pohľad na názory hráčov vyjadrené v texte recenzie, bola pre túto analýzu zvolená úloha extrakcie aspektových štvoríc. Prvkami jednotlivých štvoríc sú: aspekt (aspektový výraz), kategória aspektu, názor, a sentiment. Takto analyzované recenzie poskytujú podrobný pohľad na hru a umožňujú lepšie sumarizovať výsledky analýzy do vopred definovaných kategórií. Kategórie aspektov boli navrhnuté tak, aby pokrývali hlavné oblasti záujmu hráčov a vývojárov. Týmito kategóriami sú:

- **hrateľnosť** – táto kategória sa zameriava na mechaniky hry, ovládanie, obsah a celkovú zábavu, ktorú hra poskytuje. Príkladom môžu byť tieto úryvky z recenzií: *„tejto hre chýba akcia“*, *„ovládanie auta je super responzívne“*, *„parádne mapy“*.
- **audiovizuálna stránka** – zahŕňa grafiku, dizajn, animácie, zvuky a hudbu. Vizualne a zvukové prvky sú dôležitým aspektom videohier, pretože ovplyvňujú atmosféru a pocity, ktoré hra evokuje. Príkladmi takýchto aspektov sú: *„pohodová atmosféra levelov“*, *„úžasný soundtrack“*, *„textúry sú trochu zaostalé“*.
- **optimalizácia a chyby** – táto kategória sa týka technických problémov a chýb, aktualizácií a celkovej optimalizácie hry. Príkladom môže byť tento úryvok z recenzie: *„nová aktualizácia opravila mnoho chýb“*.
- **komunita** – komunita zastrešuje aspekty týkajúce sa interakcie hráčov, podpory zo strany vývojárov a komunikácie medzi hráčmi a tvorcami hry. Ako príklady aspektov z tejto kategórie je možné uviesť: *„vývojári sú nápomocní a aktívni na fóre“*, *„komunita tejto hry je veľmi toxická“*.
- **cena** – do tejto kategórie patria aspekty, ktorými hráči vyjadrujú spokojnosť s cenou alebo samotnú cenu hry, ako napríklad: *„super nízka cena“*, *„trochu drahá, ale stojí za to“*.
- **celková spokojnosť** – táto kategória zahŕňa celkový zážitok a spokojnosť hráčov s hrou. Príkladom, kde sa vyskytuje aspekt z tejto kategórie je napríklad recenzia *„Super hra 9/10 .“*

Úlohu extrakcie aspektových štvoríc som sa na základe dobrých výsledkov dosiahnutých a predstavených v článkoch [18] a [6], rozhodol riešiť pomocou generatívneho modelu T5 s využitím inštrukcií.

Predspracovanie textu recenzií

Pri predspracovaní textu recenzií je potrebné vykonať niekoľko krokov, ktoré pomáhajú pripraviť textové dáta pre analýzu vybranými modelmi. Tieto kroky zahŕňajú odstraňovanie nežiaduceho formátovania, konverziu textu do jednotného formátu, tokenizáciu a čistenie textu od šumu (napríklad zbytočných medzier, interpunkcie a špeciálnych znakov). Pri predspracovaní textu je taktiež často využívaná lemmatizácia a stematizácia.

Lemmatizácia a stematizácia sú techniky predspracovania textu, ktoré sa tradične používajú na zjednodušenie textu, pričom tak zredukujú jeho dimenzionalitu. Lemmatizácia prevedie slová na ich základný tvar (lemmu), zatiaľ čo stematizácia odstraňuje koncovky

slov s cieľom získať kmeň slova. Tieto techniky môžu byť užitočné pri práci s jednoduchšími modelmi strojového učenia, ktoré nemajú dostatočnú kapacitu na učenie sa zložitých vzťahov medzi slovami. Avšak pre modely typu transformer, akým je aj T5, nie je nutné vykonávať lemmatizáciu ani stematizáciu v procese spracovania. Tieto modely sú navrhnuté tak, aby si samy dokázali poradiť s prirodzeným jazykom vo svojej pôvodnej forme, bez potreby redukcie slov na ich základný tvar. Tieto modely dokážu pochopiť a zachovať jazykovú kontextualitu a vyjadrenie, čo by mohlo byť stratené pri použití lemmatizácie a stematizácie.

Dôležitou časťou spracovania textu pri učení s inštrukciami je konverzia vstupnej sekvencie do jednotného formátu inštrukcie. Použitý formát inštrukcií obsahuje okrem samotnej inštrukcie pre konkrétnu úlohu aj príklady vstupov a im zodpovedajúcich výstupov a bol inšpirovaný prácou [22].

```
[task instruction]
example 1-
input: [example sequence (review)]
output: [task output]
[end of examples task label]
input: [input sequence]
output:
```

Výpis 3.1: Formát inštrukcie používaný pri ladení

Navrhnutý formát výstupu pre extrakciu štvoric sa skladá zo štvoric (aspekt, kategória, názor, sentiment), ktoré sú medzi sebou oddelené znakom „|“. Jednotlivé prvky štvorice sú oddelené znakom „:“. Model na základe vstupnej sekvencie slov vygeneruje výstupnú sekvenciu, ktorá musí zodpovedať navrhnutému formátu výstupu. Ak vygenerovaný výstup alebo niektorá jeho zložka nespĺňa tento formát, tak potom je výstup považovaný za zlý. Prázdny výstup má nasledujúci tvar: NULL:NULL:NULL:NULL.

Dáta

Užívateľské recenzie videohier sú veľmi špecifickou doménou, ktorá sa týka hier a ich vlastností. Vytvorením datasetu založeného na týchto recenziách bolo zabezpečené, že model bude schopný správne rozumieť a identifikovať aspekty v kontexte hier. Jedným z dôvodov, prečo som sa rozhodol vytvoriť špecifický dataset, bolo aj to, že na základe prieskumu existujúcich datasetov som nenašiel žiadny dataset zameraný na české recenzie videohier a ich aspektovo-orientovanú analýzu, čo vytváralo potrebu vytvorenia takéhoto datasetu, keďže som potreboval dáta, ktoré by boli relevantné a špecifické pre danú jazykovú a kontextovú oblasť.

Na začiatku procesu bolo potrebné získať dostatočné množstvo dát, ktoré by som mohol použiť pre tréning môjho modelu. Ako zdroj dát pre tento proces slúžili používateľské recenzie na platforme Steam³. Po získaní dostatočného množstva recenzií som prešiel k spracovaniu dát. Tento krok zahŕňal čistenie textu od prebytočných znakov a rozdelenie dlhých recenzií na menšie recenzie. Bolo dôležité, aby recenzie obsahovali rôzne aspekty z rôznych predom definovaných kategórií a s rôznou polaritou sentimentu, aby som mohol vytvoriť čo najreprezentatívnejší dataset. Táto časť sa pri tvorbe datasetu ukázala byť veľmi zložitou, keďže ako vyplynulo z manuálnej analýzy prvých 100 vybraných recenzií, početnosť aspektov jednotlivých kategórií bola veľmi nevyvážená, čo je možné vidieť v tabuľke 3.1.

³<https://store.steampowered.com>

Tabuľka 3.1: Výsledok prvých 100 analyzovaných recenzií, kde bolo identifikovaných 194 aspektov. Je možné si všimnúť, že v recenziách sú najviac spomínané aspekty hrateľnosti a ukazuje sa nedostatok neutrálnych aspektov.

Kategória	Negatívne	Neutrálne	Pozitívne	Celkom
Audiovizuál	10,34%	3,45%	75,86%	15,46%
Komunita	50%	0%	50%	5,15%
Hrateľnosť	37,17%	9,73%	49,56%	57,22%
Celková spokojnosť	26,67%	6,67%	66,67%	7,73%
Optimalizácia a chyby	59,26%	11,11%	25,93%	13,40%
Cena	62,50%	0%	37,50%	4,12%

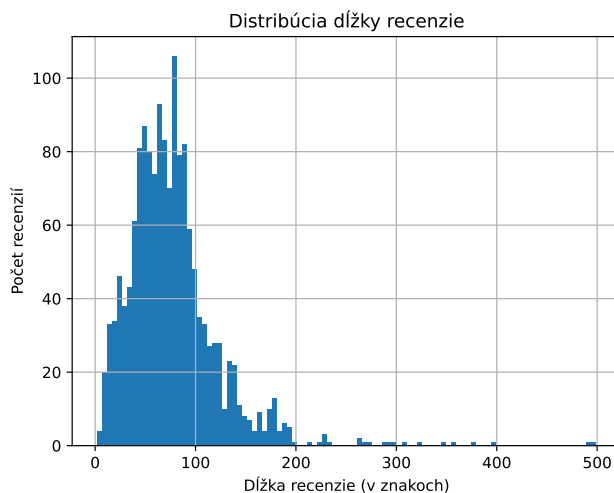
Tabuľka 3.2: Štatistika datasetu pre extrakciu štvorice (ACOS)

Kategória	Negatívne	Neutrálne	Pozitívne	Celkom
Audiovizuál	49 (22,07%)	12 (5,41%)	161 (72,52%)	222 (10,96%)
Komunita	28 (59,57%)	4 (8,51%)	15 (31,91%)	47 (2,32%)
Hrateľnosť	500 (48,36%)	59 (5,71%)	475 (45,94%)	1034 (51,04%)
Celková spokojnosť	138 (52,47%)	22 (8,37%)	103 (39,16%)	263 (12,98%)
Optimalizácia a chyby	231 (57,89%)	15 (3,76%)	153 (38,35%)	399 (19,69%)
Cena	36 (59,02%)	2 (3,28%)	23 (37,70%)	61 (3,01%)
Celkom	982 (48,47%)	114 (5,63%)	930 (45,90%)	2026

Tabuľka 3.3: Štatistika datasetu pre extrakciu trojice (ACS)

Kategória	Negatívne	Neutrálne	Pozitívne	Celkom
Audiovizuál	24 (20,34%)	5 (4,24%)	89 (75,42%)	118 (11,11%)
Komunita	44 (53,66%)	4 (4,88%)	34 (41,46%)	82 (7,72%)
Hrateľnosť	197 (36,82%)	69 (12,90%)	269 (50,28%)	535 (50,38%)
Celková spokojnosť	29 (17,79%)	14 (8,59%)	120 (73,62%)	163 (15,34%)
Optimalizácia a chyby	79 (60,31%)	9 (6,87%)	43 (32,82%)	131 (12,34%)
Cena	18 (54,55%)	0 (0,00%)	15 (45,45%)	33 (3,11%)
Celkom	391 (36,82%)	101 (9,51%)	570 (53,67%)	1062

V tvorbe dátovej sady som pokračoval anotáciou dát, čo zahŕňalo označenie trojice: aspektový výraz, kategória aspektu a sentiment (pozitívny, negatívny a neutrálny) v jednotlivých recenziách, čo som neskôr rozšíril aj o extrakciu názoru. Tento proces bol vykonaný ručne. Počas anotácie som sa stretol s niekoľkými problémami, napríklad nejednoznačnosťou aspektov alebo nejasnými formuláciami v recenziách. V takýchto prípadoch som sa snažil dosiahnuť konsenzus medzi viacerými ľuďmi, ktorým boli takéto vzorky ukázané. V prípadoch, kedy nebolo možné dôjsť k jednoznačnej dohode, som recenziu vyradil.



Obr. 3.2: Distribúcia recenzií v dátových sadách na základe počtu znakov.

Keďže som sa v neskoršej fáze projektu rozhodol túto úlohu rozšíriť o extrakciu názoru, tak som preto vytvoril novú sadu dát. Výsledkom mojej práce sú 2 datasety, ktorých bližšie štatistiky je možné vidieť v tabuľkách 3.2 a 3.3. Nedostatky týchto datasetov spočívajú v nevyváženosti jednotlivých kategórií a nedostatku neutrálnych aspektov a recenzií, ktorých je však málo aj v reálnych dátach.

Modul pre analýzu recenzií

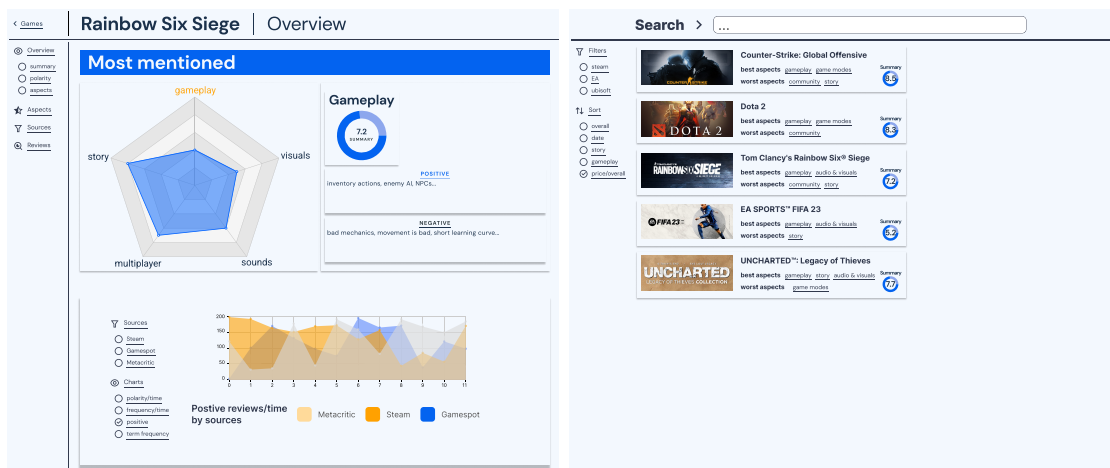
Modul je možné rozšíriť o ďalšie typy analýz, no primárne bol navrhnutý pre aspektovo-orientovanú analýzu, a konkrétne úlohu extrakcie aspektových trojíc alebo štvoríc. Vstupom modulu sú recenzie a jeho výstupom sú analyzované recenzie, kde je uvedený model, typ úlohy a predikcia. Ďalším výstupom modulu sú štrukturované dáta o extrahovaných aspektoch, ktoré sa menia podľa typu úlohy. Pri extrakcii štvoríc je to aspekt, kategória, názor, sentiment. Analýza recenzií je realizovaná pomocou predtrénovaných modelov. Tie sú inicializované a načítané do pamäte. Preto je ďalším vstupom modulu konfigurácia, v ktorej je potrebné špecifikovať miesto adresára, kde sú uložené váhy a iné konfiguračné súbory modelu. Tento modul komunikuje s databázou pomocou konektora, ktorého rozhranie sa využíva k získaniu dát z databázy, vkladaniu informácií o analýze a jej výstupoch.

Za účelom efektívnej analýzy bola navrhnutá metóda dávkového analyzovania recenzií s možnosťou ukladania jednotlivých spracovaných dávok do súborov, ktoré môžu byť následne postupne paralelne spracované a vložené do databázy. Tento prístup prináša stabilnejšie zaťaženie výpočtovej jednotky a tým aj zefektívňuje samotný proces analýzy.

3.4 Vizualizácia výsledkov

Jedným z cieľov práce bolo poskytnúť používateľom efektívny a používateľsky príjemný spôsob zobrazovania výsledkov analýzy. V systéme opísanom v predchádzajúcich sekciách bol návrh vizualizačnej časti vytvorený tak, aby dáta boli zrozumiteľne a pútavo prezentované koncovému používateľovi prostredníctvom webovej aplikácie zloženej z backendu a frontendu.

Backend systému je schopný efektívne a spoľahlivo spracovávať dáta a poskytovať ich frontendu vo forme čistých a štruktúrovaných dátových objektov skrz webové API rozhranie. Tieto dátové objekty obsahujú všetky potrebné informácie pre vizualizácie, ktoré sú následne generované na strane frontendu. API navrhutej aplikácie sa riadi REST architektonickým štýlom, ktorý sa používa na vytvorenie moderných webových aplikácií. Tento štýl využíva štandardné protokoly (HTTP, HTTPS) a metódy (GET, POST, PUT a DELETE) na komunikáciu medzi klientom a serverom. REST API poskytuje rozhranie, ktoré umožňuje výmenu dát medzi aplikáciou a databázou alebo medzi rôznymi komponentmi aplikácie. Základom tohto rozhrania sú tzv. zdroje, ktoré sú jednoznačne identifikované prostredníctvom URL adries. V rámci navrhutej aplikácie sa dáta prenášajú vo formáte JSON.



Obr. 3.3: Mockup webovej aplikácie

Frontend zohráva kľúčovú úlohu pri prezentácii dát používateľovi. Bol navrhnutý tak, aby bol intuitívny, responzívny a vizuálne príťažlivý. Pri návrhu vizualizačnej časti bola venovaná pozornosť aj prispôsobivosti a rozšíriteľnosti systému. Výsledkom je vizualizačná časť systému, ktorá používateľom poskytuje ľahko pochopiteľné a interaktívne prostredie pre analýzu a pochopenie dát. Týmto spôsobom sa dosahuje efektívna komunikácia informácií medzi systémom a používateľom, čo umožňuje rýchlejšie a informovanejšie rozhodovanie. Pri návrhu vizualizácie bol braný ohľad hlavne na kľúčové aspekty dát, ktorými sú pozitívne a negatívne názory hráčov videohier na aspekty konkrétnej hry a ich vývoj v čase. Ďalej poskytujú prehľad o analyzovaných recenziách na úrovni zdroja, čo umožňuje používateľom porovnávať rozdiely medzi názormi recenzentov medzi jednotlivými zdrojmi. To im napríklad umožňuje zistiť určitý zámer platformy pri procese zverejňovania používateľských recenzií alebo detegovať fenomén review bombing, kedy určitá skupina ľudí zahlť platformu negatívnymi recenziami v snahe poškodiť predaj alebo popularitu produktu.

Okrem vizuálneho spracovania dát sa do návrhu vizualizačnej časti systému zahrnuli aj ďalšie komponenty, ako sú napríklad nástroje na vyhľadávanie a zoradenie analyzovaných videohier.

Sumarizácia výsledkov

Sumarizácia výsledkov je podstatnou úlohou pri vizualizácii dát, keďže výsledky sumarizácie sú práve to, čo je používateľovi prezentované. Sumarizácia výsledkov analýzy na úrovni kategórie aspektu má za úlohu premietnuť používateľom sumarizovanú polaritu sentimentu pre kategóriu aspektu na číselné hodnotenie na škále od 0 do 10, na základe čoho získa používateľ prehľad o silných a slabých stránkach hry. Pri premietaní pozitívnych a negatívnych recenzií je potrebné brať do úvahy rozloženie polarít sentimentu v recenziách. Preto bolo zvolené váhové škálovanie, kde na základe empirických pokusov, boli jednotlivým polaritám sentimentu pridelené rozdielne váhy. Konkrétne je pre sumarizáciu použitý tento výpočet:

$$P_{pos} = \frac{N_{pos}}{N_{total}}, \quad P_{neg} = \frac{N_{neg}}{N_{total}}, \quad (3.1)$$

$$w = \alpha P_{pos} - \beta P_{neg} \quad (3.2)$$

$$f_{score} = \max(0, \min(10, 5 + w)), \quad (3.3)$$

kde P_{pos} (resp. P_{neg}) vyjadruje percento pozitívnych (resp. negatívnych) aspektov pre kategóriu, N_{pos} (resp. N_{neg}) je počet pozitívnych (resp. negatívnych) aspektov pre kategóriu, N_{total} je celkový počet aspektov pre kategóriu, w predstavuje váhu, kde koeficient α (resp. β) odráža relatívnu dôležitosť pozitívnych (resp. negatívnych) aspektov, f_{score} je funkcia pre výpočet číselného hodnotenia kategórie.

Backend

Backend zabezpečuje spracovanie a zhrnutie analyzovaných recenzií a identifikovaných aspektov, ktoré sú potom poskytnuté frontendu na vizualizáciu a interakciu s používateľom. Návrh backendu začal analýzou a identifikáciou kľúčových funkcií, ktoré by mal systém zvládať. To zahŕňa získavanie textových recenzií (využitím rozhrania modulu pre získavanie dát), aplikovanie metód na analýzu sentimentu, extrakciu a klasifikáciu dôležitých aspektov z recenzií (pomocou rozhrania modulu pre analýzu) a generovanie zhrnutí pre jednotlivé aspekty. Hlavnými koncovými bodmi webového API rozhrania sú koncové body, určené k:

- získavaniu zoznamu analyzovaných videohier,
- získavaniu prehľadnej sumarizácie analyzovaných recenzií na úrovni zdrojov pre špecifikovanú hru,
- získavaniu sumarizácie na úrovni kategórie aspektov pre špecifikovanú hru,
- získavaniu slovného oblaku najviac spomínaných aspektov pre špecifikovanú hru.

Navrhnuté API rozhranie taktiež poskytuje základné metódy pre vytvorenie, čítanie, aktualizáciu a vymazanie objektov hra, recenzia a zdroj recenzií. Tieto metódy sú sprístupnené iba autorizovaným používateľom. Dôvodom pre takúto reštrikciu je to, že systém bol primárne navrhnutý na poskytovanie vizualizácie výsledkov analýzy, no keďže bol braný ohľad aj na rozšíriteľnosť funkcionality, pri návrhu bolo myslené aj na možnosť pridávania objektov a použitie modulov pre analýzu a získavanie recenzií registrovaným používateľom systému.

Frontend

Frontend je kľúčovým prvkom, ktorý umožňuje používateľom analyzované hry vyhľadávať, filtrovať, radíť podľa rôznych parametrov a prezeráť si výsledky analýzy jednotlivých hier vo forme grafov.

Vyhľadávanie, filtrovanie a radenie hier je možné nájsť na stránke so zoznamom hier. Používateľ si môže zoznam filtrovať na základe časového rozmedzia vydania hry, min. a max. hodnoty skóre, podľa počtu recenzií a názvu hry. Zoznamu je možné zoradiť na základe skóre, počtu recenzií a dátumu vydania.

Prehľadná vizualizácia výsledkov analýzy hry je poskytnutá pri detailnom pohľade na hru. Detailný pohľad poskytuje súhrny prehľad o počte recenzií a aspektov podľa zdroja, kategórie a ich vývoja v čase. Taktiež poskytuje číselné ohodnotenie jednotlivých kategórií aspektov, ktoré umožňuje rýchlu identifikáciu silných a slabých stránok hry. Najčastejšie spomínané aspekty je možné nájsť pomocou slovného oblaku, ktorého obsah sa mení na základe vybranej kategórie a polarity.

Ako vizualizačný prvok pre súhrny prehľad o počte recenzií a aspektov bola zvolená kombinácia prekrývajúcich sa čiarových a plošných grafov. Čiary nesú údaje o počte aspektov vybranej polarite a plochy určujú objem recenzií pre jednotlivé zdroje. Používateľské rozhranie umožňuje dynamický výber vykresľovaných informácií a veľkosť časového intervalu, na základe ktorého je počet recenzií a aspektov agregovaný.

Vizualizačným prvkom pre sumarizáciu číselného hodnotenia kategórií aspektov je radiálny graf, čo umožňuje intuitívne vnímať celkové hodnotenie hry podľa veľkosti pokrytej plochy, ako aj to, v ktorých kategóriách hra zaostáva.

Kapitola 4

Implementácia

V tejto kapitole je podrobne prezentovaná realizácia navrhnutého riešenia pre zber, ukladanie, analýzu a vizualizáciu recenzií video hier a súvisiacich dát. Sú opísané konkrétne technológie, programovacie jazyky a nástroje, ktoré boli použité pri implementácii jednotlivých modulov systému. Taktiež je prezentovaná integrácia modulov s cieľom zabezpečiť správne fungovanie celého systému. Kapitola poskytuje podrobnosti o tom, ako boli prepojené jednotlivé úlohy z návrhu a výsledné riešenie, ktoré spĺňa stanovené požiadavky.

4.1 Použité nástroje

Implementácia navrhnutého systému si vyžadovala prácu s viacerými technológiami, knižnicami a nástrojmi. Najdôležitejšie z nich sú stručne opísané v nasledujúcich podsekciach.

Git

Git je distribuovaný systém kontroly verzií, ktorý sa používa na sledovanie zmien v zdrojovom kóde počas vývoja softvéru. V tomto projekte bol Git použitý pre správu verzii a zálohovanie dát na serveroch služby Github¹, čo uľahčoval prenos repozitára na iné zariadenia a zabezpečilo to kontinuitu práce na projekte.

PostgreSQL

PostgreSQL² je výkonný a škálovateľný objektovo-relačný databázový systém. Navyše poskytuje podporu vlastných dátových typov, možnosť full-textového vyhľadávania a je dostupný na serveri Athena 1³.

FastAPI

FastAPI⁴ je moderný, rýchly a vysoko výkonný webový rámec pre vytváranie API v programovacom jazyku Python. Bol zvolený pre jeho rýchlosť, jednoduchosť a výkonnosť. Tento rámec navyše poskytuje automatickú generáciu dokumentácie a overovanie parametrov, čo uľahčuje vývoj a udržiavanie aplikácie.

¹<https://github.com>

²<https://postgresql.org>

³<https://athena1.fit.vutbr.cz>

⁴<https://fastapi.tiangolo.com>

SQLAlchemy

SQLAlchemy⁵ je populárna knižnica pre prácu s databázami v programovacom jazyku Python. Poskytuje abstrakciu na vyššej úrovni pre komunikáciu s databázou a umožňuje pracovať s dátami bez znalosti konkrétnych SQL príkazov. V tejto práci bola použitá pre jej jednoduchosť a kompatibilitu s PostgreSQL. Ďalším dôvodom bola možnosť asynchrónnej implementácie metód určených pre prácu s databázou pomocou databázového adaptéra asynpcp⁶ podporujúceho asynchrónny beh databázovej relácie.

PyABSA

PyABSA je knižnica pre aspektovo-orientovanú analýzu sentimentu [32]. Túto knižnicu som sa rozhodol použiť kvôli jej jednoduchému rozhraniu pre tréningovanie a experimentovanie s modelmi. Navyše poskytuje implementáciu najnovších modelov v oblasti aspektovo-orientovanej analýzy a dátové sady, vhodné na experimentovanie s modelmi.

Alembic

Alembic⁷ je ľahký databázový migračný nástroj pre SQLAlchemy. V tejto práci bol použitý na správu zmien v databázových schémach, čo uľahčuje nasadenie a aktualizáciu aplikácie.

SvelteKit

SvelteKit⁸ je moderný rámec pre vytváranie webových aplikácií. Bol zvolený pre vývoj prezentačnej vrstvy aplikácie kvôli jeho rýchlosti, jednoduchosti a elegantnému prístupu k vývoju komponentov.

Layer Cake

Layer Cake⁹ je knižnica určená pre vizualizáciu dát, ktorá je založená na Svelte¹⁰ a D3.js¹¹. Jej použitie zjednodušilo proces vytvárania vrstvených a responzívnych vizualizácií, ktoré umožňujú užívateľom ľahko analyzovať dáta z rôznych uhlov pohľadu.

Docker

Docker¹² je platforma pre vývoj, distribúciu a spustenie kontajnerizovaných aplikácií. Docker bol v práci použitý na nasadenie a správu databázy použitej pri vývoji, čo uľahčovalo vývoj v rôznych prostrediach.

⁵<https://www.sqlalchemy.org>

⁶<https://www.psycopg.org>

⁷<https://alembic.sqlalchemy.org>

⁸<https://kit.svelte.dev>

⁹<https://layercake.graphics>

¹⁰<https://svelte.dev>

¹¹<https://d3js.org>

¹²<https://www.docker.com>

4.2 Štruktúra aplikácie

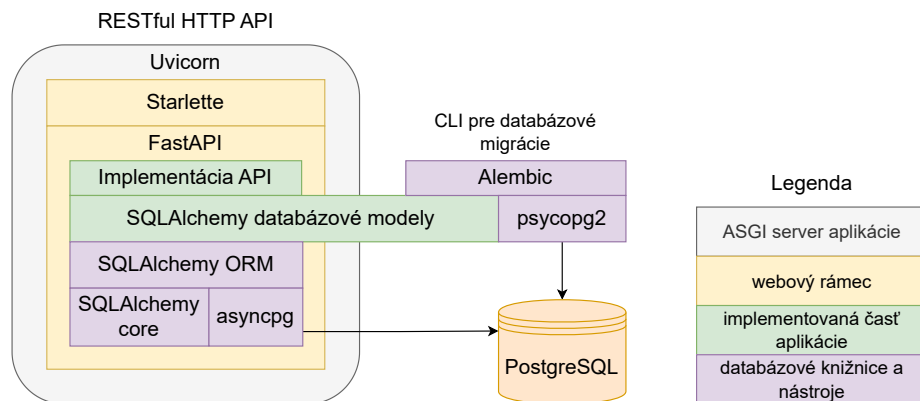
Aplikácia je rozdelená na backend a frontend, pričom samostatné moduly na získavanie recenzií a ich analýzu sú prístupné skrz balíka služieb poskytovaných backendom. Koreňový adresár projektu sa teda člení na:

- adresár `/app` obsahujúci moduly, submoduly, služby a spúšťacie skripty backendu,
- adresár `/frontend` obsahujúci komponenty, využívané balíčky a konfiguračné súbory prezentačnej vrstvy,
- adresár `/alembic` obsahujúci konfiguráciu a migračné skripty pre databázu s využitím nástroja Alembic.

Okrem týchto adresárov sa v koreňovom adresári aplikácie nachádzajú aj konfiguračné súbory pre nastavenie premenných prostredia, vytvorenie kontajnera s vývojovou databázou pomocou nástroja Docker a iné súbory potrebné pre konfiguráciu vývojového prostredia.

Backend spolu s modulmi pre získavanie dát a analýzu recenzií sú implementované v jazyku *Python 3.8*. Backend aplikácie nasleduje odporúčanú štruktúru rámca FastAPI, čo zaisťuje dobrú organizáciu kódu, modulárnosť a škálovateľnosť. FastAPI odporúča rozdelenie aplikácie do rôznych adresárov, ktoré zodpovedajú jednotlivým častiam aplikácie a zabezpečujú jednoduchú orientáciu a údržbu kódu. Štruktúre backendu zodpovedá toto členenie:

- **api**: adresár obsahujúci súbory definujúce API koncové body, ktoré zabezpečujú komunikáciu medzi frontendom a backendom.
- **core**: v tomto adresári sú umiestnené konfiguračné súbory zodpovedné za nastavenie prostredia, parametre aplikácie, pripojenie k databáze, autentifikáciu a autorizáciu užívateľov, ako aj middleware funkcie pre spracovanie požiadaviek.
- **crud**: adresár zahrňuje CRUD (vytvoriť, čítať, aktualizovať, zmazať) operácie pre prácu s dátami v databáze. CRUD operácie sú špecifikované pre každú entitu vrátane recenzií, hier, recenzentov, analyzovaných recenzií, extrahovaných aspektov, používateľov a špecifických operácií pre služby.
- **db**: adresár obsahuje súbory pre pripojenie k databáze a jej inicializáciu. Aplikácia využíva SQLAlchemy ako objektovo-relačný mapovací (ORM) nástroj pre prácu s databázou.
- **models**: obsahuje definície databázových modelov pre jednotlivé entity. Modely sú navrhnuté tak, aby zodpovedali štruktúre databázy a umožnili efektívnu prácu s dátami pomocou ORM.
- **schemas**: adresár obsahuje definície Pydantic schém, ktoré zabezpečujú validáciu dát pri komunikácii cez API. Schémy kontrolujú dáta a konvertujú ich medzi databázovými modelmi a JSON formátom použitým vo frontendovej časti.
- **services**: tento adresár zahrňuje služby integrujúce funkcionality modulov na zber recenzií a analýzu sentimentu do backendu. Služby komunikujú s modulmi, poskytujú im dáta alebo získavajú dáta pre uloženie do databázy či poskytnutie frontendu.



Obr. 4.1: Schéma aplikačného zásobníka webového API.

Frontend aplikácie je vytvorený s využitím moderného rámca pre vývoj webových aplikácií – SvelteKit. SvelteKit poskytuje základnú štruktúru pre projekt, ktorá umožňuje jednoduché pridávanie nových funkcionalít a údržbu existujúceho kódu. SvelteKit aplikácia sa rozdeľuje na tieto základné časti:

- **src/routes**: súbory a podadresáre pre jednotlivé stránky aplikácie. Každá stránka je reprezentovaná komponentom Svelte, čo zaisťuje jasnú štruktúru a modulárnosť aplikácie.
- **src/lib**: adresár obsahuje pomocné funkcie, konfiguráciu a iné súbory, ktoré sú využívané v rámci celej aplikácie. Podadresár **lib/components** obsahuje opakujúce sa a zdieľané komponenty, ktoré sú použité na viacerých stránkach alebo častiach aplikácie. Zdieľané komponenty zjednodušujú vývoj a znižujú náročnosť údržby kódu. Ďalšou dôležitou časťou adresára **lib** je podadresár **client**, kde sa nachádza kód implementujúci modul klienta, ktorý integruje služby poskytované API rozhraním.
- **src/theme**: tento adresár je určený pre definovanie štýlov aplikácie, vrátane preddefinovaných farieb, typografie, rozloženia a ďalších vizuálnych prvkov. Pre definovanie štýlov tejto aplikácie bol použitý preprocesorový skriptovací jazyk SCSS.
- **static**: obsahuje statické súbory, ako sú obrázky, štýly a fonty, ktoré sú súčasťou aplikácie.

Správa balíkov frontendu aplikácie je zabezpečená prostredníctvom nástroja NPM (Node Package Manager)¹³. NPM je populárny správca balíkov pre JavaScript a je súčasťou prostredia Node.js. Jeho hlavným účelom je uľahčiť inštaláciu, aktualizáciu a správu závislostí projektu. V aplikácii sú konfiguračné informácie o balíčkoch a závislostiach uložené v súbore **package.json**, ktorý sa nachádza v koreňovom adresári frontendu. Tento súbor obsahuje informácie o názve projektu, verzii, autorovi, licencií a hlavne o zozname balíkov a závislostí potrebných pre beh a vývoj aplikácie. Keďže frontendová časť aplikácie využíva SvelteKit, mnoho z týchto balíkov súvisí s funkciami a rozšíreniami SvelteKitu, ako aj s ďalšími knižnicami a nástrojmi potrebnými pre vývoj, ako napr. **d3.js** (pre vizualizáciu dát) či **SCSS** (pre štýlovanie).

¹³<https://www.npmjs.com/>

4.3 Modul pre získavanie dát

Táto sekcia popisuje modul systému určený k získavaniu dát. Na uskutočňovanie asynchrónnych HTTP požiadaviek je použitá knižnica *httplib*¹⁴. Implementáciu validátorov očakávaných odpovedí uľahčuje knižnica *pydantic*¹⁵. Adresár tohto modulu je umiestnený v adresári backend služieb (`services/scrapper`).

Pre každý zdroj dát je definovaná vlastná trieda dediaci a implementujúca základné metódy triedy `Scraper`. Triedy určené k získavaniu dát pre jednotlivé zdroje sú navrhnuté tak, aby boli čo najviac nezávislé, čo umožňuje ľahko pridať ďalšie zdroje alebo upraviť už existujúce zdroje bez toho, aby to ovplyvnilo celkovú funkcionálnosť scraperu. Každá trieda poskytuje metódy na získanie informácií o jednotlivých hrách a ich recenziách, ktoré sú jednoducho prispôsobiteľné a môžu byť upravené podľa konkrétnych potrieb.

Vďaka použitiu asynchrónnych metód a generátorov môže scraper efektívne získavať dáta viacerých zdrojov naraz, čím sa znižuje celkový čas potrebný na získanie požadovaných informácií.

Spracovanie HTTP požiadaviek

Na zabezpečenie správneho fungovania scraperov sú implementované metódy pre overenie a spracovanie odpovedí od serverov, ako aj na správu chýb a opätovné požiadavky. Metóda `get_retry` slúži na uskutočňovanie asynchrónnych požiadaviek s automatickým opätovným pokusom v prípade zlyhania alebo prekročenia časového limitu. O spracovanie odpovedí od serverov sa stará metóda `handle_response`, ktorá skontroluje, či je odpoveď platná, a ak je platná, použije preddefinovanú formátovaciu funkciu na extrahovanie požadovaných dát.

SteamScrapper

Trieda `SteamScrapper` slúži na získavanie údajov zo Steam API a je odvodená od triedy `Scraper`. Od ďalších typov zberačov dát sa odlišuje stratégiou získavania recenzií, ktorá sa zameriava na získavanie recenzií pre špecifickú hru alebo zoznam hier. Trieda implementuje nasledujúce metódy:

- `get_games`: získava zoznam hier. Výsledkom je zoznam identifikátorov hier (`appId`).
- `get_game_info`: na základe zadaného identifikátora hry (`appId`) získava informácie o hre. Výsledkom je objekt `SteamAppDetail` s informáciami o hre.
- `get_game_reviews`: získava recenzie pre hru na základe zadaného identifikátora hry (`appId`). Výsledkom je zoznam objektov `SteamReview` s recenziami.
- `get_games_info`: táto metóda získava informácie o viacerých hrách naraz na základe zadaných identifikátorov hier (`appId`). Táto metóda vracia zoznam objektov `SteamAppDetail` s informáciami o hrách.
- `games_page_generator`: asynchrónny generátor, ktorý postupne generuje zoznamy objektov `SteamAppDetail`.
- `game_reviews_page_generator`: asynchrónny generátor, ktorý generuje zoznam objektov `SteamReview` pre konkrétnu hru (`appId`).

¹⁴<https://www.python-httpx.org>

¹⁵<https://docs.pydantic.dev>

- `get_games_reviews`: získava recenzie pre viacero hier naraz na základe zadaných identifikátorov hier (`appId`) a parametrov filtra. Výsledkom je slovník, kde kľúčom je `appId` a hodnotou zoznam objektov `SteamReview`.

GamespotScraper

Úlohou triedy `GamespotScraper` je získavať dáta z Gamespot API a obsahuje nasledujúce metódy:

- `get_reviews_page`: asynchrónna metóda na získanie stránky s recenziami. Výsledok je dvojica (URL, `GamespotApiResponse`).
- `game_reviews_page_generator`: asynchrónny generátor, ktorý získava recenzie po jednotlivých stránkach a vracia zoznamy objektov `GamespotReview`.
- `get_all_reviews`: metóda, ktorá používa `game_reviews_page_generator` na získanie všetkých recenzií a vráti zoznam objektov `GamespotReview`.

DoupeScraper

Trieda `DoupeScraper` získava dáta zo stránky Doupe¹⁶ pomocou techniky web scraping. Medzi metódy tejto triedy patria:

- `get_reviews_page`: asynchrónna metóda na získavanie URL adries stránok s recenziami. Výsledok je dvojica (URL, zoznam objektov `DoupeReview`).
- `game_reviews_page_generator`: asynchrónny generátor, ktorý získava a spracúva stránky s recenziami. Výsledkom každej iterácie je zoznam prislúchajúcich objektov `DoupeReview`.
- `get_reviews_detail`: synchronná metóda na získanie detailov o recenziách pre zadaný zoznam objektov `DoupeReview`. Výsledok je zoznam objektov `DoupeReview` s doplnenými informáciami.
- `get_review_detail`: asynchrónna metóda na extrahovanie ďalších informácií o jednej recenzii pre zadaný objekt `DoupeReview`, ktorý už obsahuje URL adresu konkrétnej recenzie. Výsledok je dvojica (URL, objekt `DoupeReview` s doplnenými informáciami).

Overenie dát

Overenie dát je implementované v každej triede zberača dát prostredníctvom validátorov. Validátor je metóda, ktorá skontroluje, či daná odpoveď od servera je správna a či obsahuje požadované informácie. Každý typ zberača dát používa špecifický validátor, ktorý skontroluje očakávaný stavový kód a obsah odpovede. `SteamScraper` a `GamespotScraper` používajú metódu `json_response_validator`. `DoupeScraper` získava obsah extrahovaním informácií z HTML kódu stránky, preto používa metódu `html_response_validator`.

Validátor je použitý v metóde `handle_response`. Táto metóda sa stará o spracovanie odpovede od servera. Najprv aplikuje validátor na odpoveď, potom spracuje odpoveď prostredníctvom formátovacej metódy (`formatter`), ktorá extrahuje a spracuje dáta do požadovaného formátu a vráti očakávané objekty.

¹⁶<https://doupe.zive.cz/recenze>

Prepojenie s databázou

Rozhranie modulu je implementované v súbore `db_scraper.py`. Trieda `DBScraper` je navrhnutá tak, aby prepojila funkcionality jednotlivých tried typu `Scraper` s databázou a backendom, čím poskytuje rozhranie na získavanie informácií o hrách, recenziách a ďalších údajoch, a na ich ukladanie do databázy.

Pre vytvorenie inštancie triedy `DBScraper` je potrebné použiť metódu `create`, ktorej vstupnými argumentami sú objekty `scraper` a `session` (databázová relácia). Trieda `DBScraper` obsahuje hlavné metódy, ktoré sú zodpovedné za získavanie informácií o hrách a ich recenziách, a o ich následne ukladanie do databázy.

Medzi tieto metódy patrí:

- metóda `scrape_games` získava zoznam hier zo zdroja a ukladá ich do databázy,
- metóda `scrape_reviews_for_game` slúži na získavanie a ukladanie recenzií pre konkrétnu hru,
- metóda `scrape_all_reviews_for_not_updated_games` získava recenzie pre všetky hry, ktoré ešte neboli aktualizované, a ukladá ich do databázy,
- metóda `scrape_all_reviews` získava všetky recenzie zo zdroja a ukladá ich do databázy,
- metódy `add_games_to_db`, `add_reviews_to_db`, `add_reviewers_to_db` sú pomocné metódy slúžiace na ukladanie získaných objektov a ich metadát. Pre každý získaný objekt prebieha kontrola, či už nie je uložený v databáze.

4.4 Modul pre analýzu recenzií

Táto sekcia popisuje zaujímavé metódy a časti implementácie modulu určeného k aspektovo-orientovanej analýze sentimentu. V rámci štruktúry aplikácie je tento modul umiestnený v adresári `services/analyzer`.

Hlavná funkcionality modulu je implementovaná v súbore `db_analyzer.py`, ktorý navyše implementuje aj rozhranie pre spúšťanie z príkazového riadka a prepája jednotlivé komponenty tohto modulu s backendom. Logika analýzy recenzií je implementovaná metódou `analyze_db_reviews`, kde sa na začiatku podľa úlohy a identifikátora modelu inicializuje natrénovaný model, ktorý sa bude používať na analýzu. Potom sa získa počet recenzií, ktoré ešte neboli spracované vybraným modelom. Nespracované recenzie sa ďalej spracovávajú po dávkach, ktorých veľkosť určuje parameter `batch_size`. Dávkové spracovanie je umožnené vďaka implementácii metódy `batch_predict` triedy `ABSAGenerator`. Pri analýze sa rozlišujú krátke a dlhé recenzie. Krátke recenzie sa analyzujú priamo, zatiaľ čo dlhé recenzie (nad 200 znakov) sú rozdelené na vety a analyzujú sa po častiach. Toto členenie vyplýva z dĺžok recenzií vyskytujúcich sa v sade dát, na ktorej boli modely doladené (možné vidieť na obrázku 3.2). Ďalším dôvodom rozdeľovania dlhých recenzií sú menšie pamäťové nároky pri spracovávaní dávky. Rozdeľovanie recenzie tiež zabráni strate informácií pri zarovnaní jednotlivých vzoriek dávky. Výsledky analýzy sú nakoniec uložené do databázy alebo v prípade, že je zvolená voľba `dump`, do súboru. Spracovanie výsledkov analýzy uložených v súboroch je implementované metódou `insert_from_dumped_file`.

Predspracovanie textu recenzií

Predtým, ako model analyzuje text recenzie, je potrebné odstrániť nežiaduce formátovanie textu recenzie a očistiť ho od zbytočných medzier, interpunkcie a špeciálnych znakov. Tieto kroky sú implementované metódami, ktoré sa nachádzajú v súbore `utils.py`. Špecifickou časťou predspracovania textu pre modely a úlohy použité v tejto práci je prevod textu recenzie do formátu definovaného úlohou.

Tento formát sa skladá z týchto častí:

- inštrukcia: slovný popis úloh,
- príklady: majú podobu dvojice vstup, výstup,
- vstup: značka označujúca začiatok textu, ktorý má byť analyzovaný,
- výstup: značka, ktorá označuje koniec vstupu.

Formát každej úlohy má svoju špecifickú inštrukciu a aj očakávaný tvar výstupu. Definície inštrukcií pre jednotlivé úlohy sa nachádzajú v súbore `acos/instruction.py`.

Jazykové modely

Triedy implementujúce rozhranie generatívnych jazykových modelov sa nachádzajú v súbore `acos/model.py`. Pri ich implementácii boli využité knižnice `PyABSA`¹⁷, `Pytorch`¹⁸ a `Transformers`¹⁹. Hlavnou triedou je trieda `ABSAGenerator`, ktorá je potomkom triedy `T5Generator`, a je určená pre spoločnú extrakciu určitých častí zo štvorice: aspekt, kategória aspektu, názor, polarita sentimentu. V tejto triede sa nachádza pomocná metóda `_get_instructor`, ktorá je zodpovedná za prípravu vstupu pre rôzne typy spoločnej extrakcie určenej parametrom `task`. Ďalšia metóda `decode_quadruple_from_output` dekóduje výstup modelu na štruktúru obsahujúcu aspekt, kategóriu, názor a polaritu. Trieda `ABSAGenerator` obsahuje aj metódy `predict` a `batch_predict`, ktoré umožňujú predikciu výstupu z modelu pre jednotlivé texty alebo dávky textov.

4.5 Vizualizácia

Pre vytvorenie moderného a responzívneho používateľského zážitku bola použitá knižnica `Svelte Material UI (SMUI)`²⁰, ktorá je implementáciou materiálového dizajnu pre `Svelte`. `SMUI` poskytuje širokú škálu predpripravených komponentov a štýlov, ktoré zabezpečujú konzistentný vzhľad a správanie celého používateľského rozhrania.

Hlavnými prvkami vizualizácie sú zoznam videohier a detailný pohľad na konkrétnu videohru. Preto budú nasledujúce podsekcie zamerané najmä na implementáciu týchto častí.

Komunikácia s API

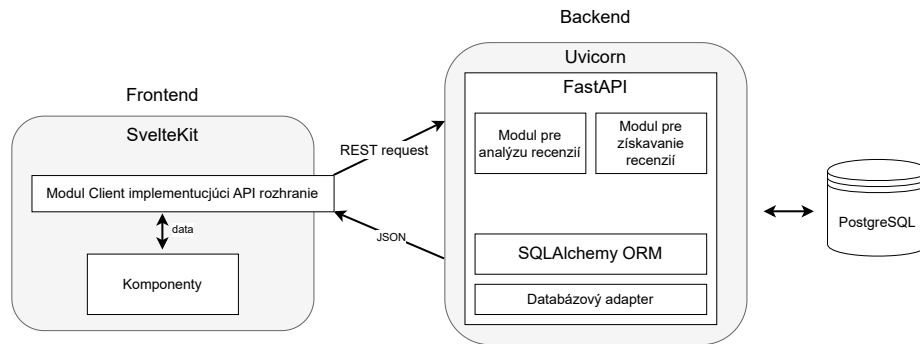
Dáta sú prezentačnej vrstve predávané prostredníctvom API rozhrania. Funkcionalita API rozhrania a biznis logika je implementovaná backendom, ktorý poskytuje nasledujúce služby:

¹⁷<https://github.com/yangheng95/PyABSA>

¹⁸<https://pytorch.org>

¹⁹<https://huggingface.co/docs/transformers>

²⁰<https://sveltematerialui.com>



Obr. 4.2: Zjednodušená schéma architektúry aplikácie.

- **users:** je zodpovedná za správu používateľských účtov. Umožňuje vytvárať, čítať, aktualizovať a mazať používateľské účty. Medzi hlavné operácie, ktoré sú k dispozícii, patria registrácia nových používateľov, získavanie informácií o existujúcich používateľoch a úprava používateľských údajov.
- **auth:** stará sa o autentifikáciu a autorizáciu používateľov. Poskytuje metódy na prihlásenie, odhlásenie a obnovenie hesla. Po úspešnom prihlásení sa vytvorí a vráti prístupový token, ktorý sa potom používa na overenie oprávnení pri prístupe k chráneným prostriedkom a metódam API.
- **games:** zabezpečuje správu a prístup k dátam o videohrách. Umožňuje vytvárať, čítať, aktualizovať a mazať záznamy o hrách. Okrem toho poskytuje metódy na vyhľadávanie hier podľa rôznych kritérií, ako sú názov hry, dátum vydania, hodnotenie a počet recenzií. Koncové body `games/[id]/summary/v2/[time_interval]` a `games/[id]/summary/aspects/[time_interval]` poskytujú počty aspektov agregovaných na základe kategórie, zdroja, polarity sentimentu a časového intervalu, ktoré sú frontendom prezentované používateľom vo forme grafov.
- **reviews:** sa venuje správe recenzií a hodnotení videohier. Umožňuje pridávať, čítať, aktualizovať a mazať recenzie. Taktiež poskytuje metódy na získanie recenzií pre konkrétnu hru alebo používateľa a na vyhodnotenie celkového hodnotenia hry na základe recenzií.

Pre zjednodušenie práce s REST API a jeho koncovými bodmi je použitý modul klienta, ktorý je generovaný na základe súboru `openapi.json`. Tento súbor obsahuje špecifikáciu REST API podľa štandardu OpenAPI. OpenAPI je štandardný formát pre popis API, ktorý umožňuje jednoduché zdieľanie a generovanie klientov pre rôzne jazyky a platformy. Súbor `openapi.json` obsahuje všetky potrebné informácie o API, ako sú dostupné koncové body, metódy, parametre, návratové hodnoty a autentifikácia. Na základe špecifikácie v tomto súbore je možné automaticky generovať klientský modul, ktorý zjednodušuje prácu s API a zabezpečuje správne typovanie a validáciu dát. Generovanie klienta môže byť realizované pomocou nástroja, akým je napríklad OpenAPI Generator²¹.

²¹<https://openapi-generator.tech>

Responzivita a reaktívne programovanie

Responzivita aplikácie je dosiahnutá pomocou reaktívneho programovania, ktoré je prirodzenou súčasťou Svelte. V Svelte sa na to používajú objekty zvané **stores**, ktoré musia definovať metódu **subscribe**. Pre uľahčenie práce s kontraktom týchto úložísk sa používa špeciálna syntax **\$store**, určená na odkazovanie na reaktívne úložiská. Táto syntax nám umožňuje sledovať zmeny v úložiskách a automaticky aktualizovať komponenty, ktoré sú na týchto úložiskách závislé.

Vo vytvorenej aplikácii je transformácia dát grafu závislá na reaktívnej syntaxi, ktorá sleduje zmeny v nastaveniach grafu, ako aj vo filtroch a zoradení zoznamu hier. Keďže Svelte automaticky sleduje zmeny v reaktívnych premenných, všetky zmeny nastavení, filtrov a zoradenia sú okamžite aplikované na dáta grafu, čo vedie k rýchlej a plynulej aktualizácii vizualizácií. Kód implementujúci reaktívne úložiská sa nachádza v adresári `src/lib/stores`.

Implementácia zoznamu hier

Zoznam hier je implementovaný ako Svelte komponent, ktorý sa stará o vykresľovanie stránky zoznamu hier a umožňuje používateľom hry vyhľadávať, filtrovať a radiť. V rámci adresára prezentačnej vrstvy sa kód implementujúci tento komponent nachádza v súbore `src/routes/games/+page.svelte`.

Pre zobrazenie jednotlivých hier je použitý komponent `GameCard.svelte`. Tento komponent zobrazuje informácie o hre a umožňuje prechod na detailný pohľad na hru. Na navigáciu medzi rôznymi stránkami zoznamu hier slúži panel s tlačidlami na konci zoznamu.

Filtrovanie a zoradenie zoznamu

Funkcionalita filtrovania a zoradenia sú implementované backendom. Pri každej zmene filtra alebo zoradenia po interakcii používateľa s frontendom sa odošle požiadavka na koncový bod `games`, ktorý je určený na získanie filtrovaného a zoradeného zoznamu. Jednotlivé nastavenia sa predávajú v parametroch cesty požiadavky.

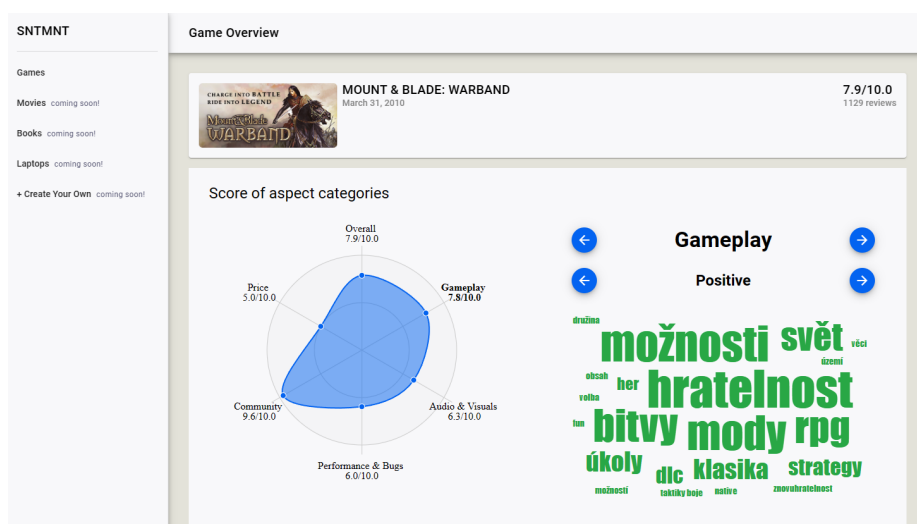
Na úrovni prezentačnej vrstvy sú nastavenia filtra implementované pomocou komponentu `Accordion`, ktorý tvorí sekciu nastavení filtrov. Používateľ môže recenzie filtrovať nastavením požadovaných filtrovacích parametrov, ktorými sú:

- časové rozmedzie vydania hry: používateľ môže filtrovať hry na základe dátumu vydania, kde je možné zvoliť minimálny a maximálny dátum.
- minimálna a maximálna hodnota skóre: filtrovanie hier podľa hodnotenia umožňuje používateľovi nájsť hry s určitou úrovňou popularity alebo kvality. Používateľ môže zadať minimálne a maximálne hodnotenie, ktoré hry musia dosiahnuť, aby sa zobrazili v zozname.
- počet recenzií: používateľ môže filtrovať hry podľa počtu recenzií. Táto možnosť umožňuje získať prehľad o hrách s väčším alebo menším množstvom spätnej väzby od hráčov a môže byť užitočná pri identifikácii populárnych či menej známych hier.
- názov hry: vyhľadávanie hier podľa ich názvu je realizované pomocou komponentu `Autocomplete`. Tento komponent je umiestnený na hornom paneli zoznamu videohier a je vždy viditeľný. Po zadaní textu sa pomocou triedy `GamesService`, ktorá implementuje API rozhranie pre koncový bod hier, získa zoznam zodpovedajúcich názvov hier uložených v databáze aplikácie.

Sekcia nastavení filtra ďalej obsahuje tlačidlo **apply**, ktorým používateľ aplikuje nové kritéria filtra, a tlačidlo **reset**, ktorým sa obnovia pôvodné nastavenia filtra. Pre zmenu zoradenia sú určené tlačidlá na hornom paneli zoznamu hier. Zoznam je možné zoradiť na základe skóre, počtu recenzií a dátumu vydania.

Implementácia detailného pohľadu

Detailný pohľad sa skladá z viacerých vizualizačných komponentov zoskupených v komponente stránky detailného pohľadu (`src/routes/games/[id]/+page.svelte`). Komponent stránky sa stará o získanie dát z webového rozhrania aplikácie, ktoré ďalej posúva príslušajúcim komponentom určeným na ich vizualizáciu. Hlavnými vizualizačnými komponentami sú komponent prehľadu a komponent pre zobrazenie časových grafov. Pre manipuláciu a prácu s dátami je použitá knižnica D3.js, zatiaľ čo základné časti grafov, ako sú osi, plochy a čiary, sú implementované s využitím knižnice Layer Cake. Layer Cake poskytuje rámec s reaktívnymi premennými pre rôzne nastavenia grafu, ako sú šírka, výška, škály a domény hodnôt.



Obr. 4.3: Vizualizácia radiálneho grafu a slovného oblaku vo webovej aplikácii

Komponent prehľadu

Komponent prehľadu (`Overview.svelte`) je zodpovedný za vizualizáciu skóre hry v jednotlivých kategóriach aspektov pomocou radiálneho grafu a za vykreslenie slovného oblaku najviac spomínaných aspektov pre konkrétnu kategóriu a polaritu sentimentu.

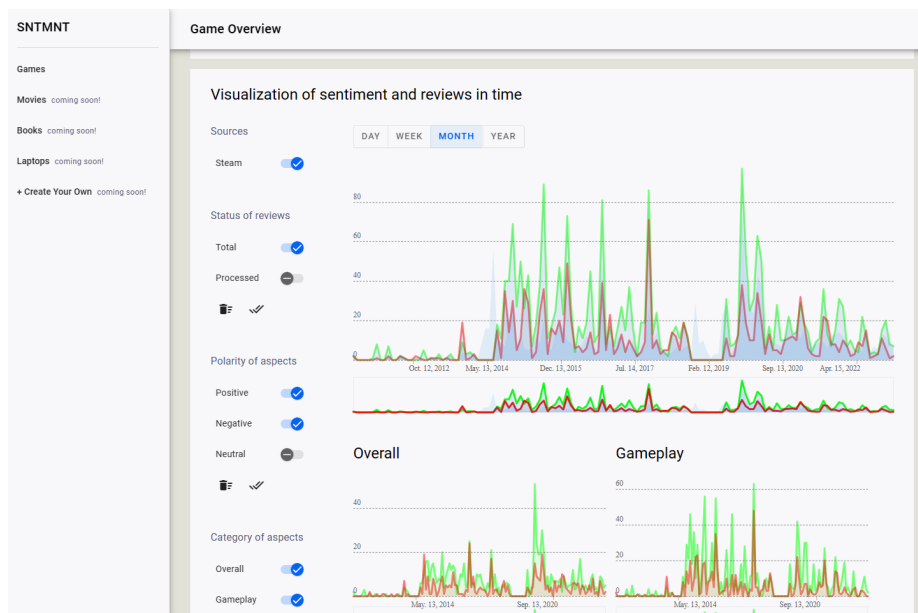
Komponent pre slovný oblak exportuje dve premenné: `data` a `selectedCategory`. Tie sú použité na definovanie vstupných údajov pre vizualizáciu slovného oblaku. Ďalej obsahuje niekoľko reaktívnych premenných a funkcií, ktoré slúžia na aktualizáciu vizualizácie a navigáciu medzi rôznymi kategóriami a polaritami. Funkcia zodpovedná za vykreslenie obsahu slovného oblaku (`updateWordcloud`) sa volá vždy, keď sa zmení aktuálna kategória alebo polarita. Táto funkcia spracováva vstupné údaje, vytvára škály pre farby a veľkosti slov a nakoniec inicializuje inštanciu triedy `layout` z `d3-cloud` (balík určený k vytváraniu slovných oblakov), ktorá sa stará o vykresľovanie slovného oblaku. Slovný oblak je vytvorený

tak, že každé slovo zobrazuje svoju frekvenciu vo veľkosti písma a farbe. Výsledkom je vizualizácia, ktorá umožňuje rýchle získanie prehľadu o najčastejších slovách a ich dôležitosti v rámci danej kategórie a polarity.

Radiálny graf, ktorý zobrazuje skóre jednotlivých kategórií hry, tvoria komponenty `AxisRadial.svelte` a `RadarChart.svelte`. Skóre je vypočítané na strane prezentačnej vrstvy a slúži na to funkcia `calculateCategoryScores`, ktorá na výpočet skóre z agregovaných dát používa vzorec 3.3.

Komponent pre zobrazenie časových grafov

Tento komponent je zodpovedný za vizualizáciu dát v čase a používa viaceré podkomponenty, ktoré spolu tvoria komplexnú vizualizáciu časových radov. Dáta pre tieto grafy sú získané pomocou koncových bodov `games/[id]/summary/v2` (pre agregované časové rady počtu recenzií) a `games/[id]/summary/aspects` (pre agregované časové rady počtu aspektov). Získané dáta sa pred vykreslením transformujú podľa nastavení používateľa.



Obr. 4.4: Vizualizácia grafov časového vývoja sentimentu vo webovej aplikácii

Grafy sú rozdelené do dvoch sekcií. Prvá sekcia zobrazuje graf časového vývoja počtu recenzií a celkového počtu aspektov určitej polarite podľa zdroja. Farba plochy pod krivkou odlišuje jednotlivé zdroje recenzií a farba krivky rozlišuje jednotlivé polarite. Tento graf tvoria komponenty `Multiline.svelte` a `MultiArea.svelte`, ktoré zobrazujú čiary a plochy pod krivkami pre rôzne zdroje a polarite aspektov. Druhá sekcia zobrazuje podrobnejšie grafy pre jednotlivé kategórie aspektov. Tieto grafy sú vytvorené pomocou komponentu `SyncedBrush.svelte`, ktorý zobrazuje časový vývoj aspektov určitej polarite. Pre každú kategóriu aspektov je vykreslený samostatný graf. Používateľ môže pomocou panelu s nastaveniami umiestneným vedľa grafov filtrovať zobrazenie dát podľa zdrojov, stavu recenzií (spracované alebo všetky), polarite aspektov a kategórie aspektov. Tieto nastavenia ovplyvňujú vizualizáciu v hlavnom grafe, ako aj v synchronizovaných grafoch nižšie. Na zobrazenie hodnôt v aktuálnom bode sa používa komponent `SharedTooltip.svelte`, ktorý sa zobrazí pri vstupe kurzora do blízkeho okolia bodu.

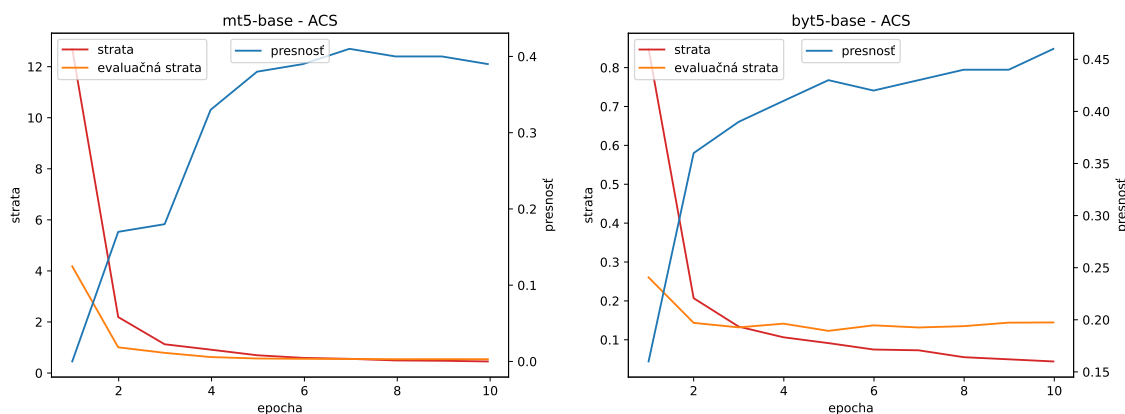
Kapitola 5

Experimenty a vyhodnotenie

Táto kapitola ponúka prehľad o experimentoch vykonaných v rámci ladenia generatívnych jazykových modelov. Ďalej sa zameriava na posúdenie použiteľnosti systému skúmaním trendov pomocou vizualizácie výsledkov analýzy a následného zhodnotenia v porovnaní so skutočnosťou. Na záver sa kapitola venuje testovaniu používateľského rozhrania webovej aplikácie, ktoré prebiehalo formou osobných stretnutí s účastníkmi testovania.

5.1 Porovnanie generatívnych jazykových modelov

Porovnania sú založené na výsledkoch z procesu ladenia, ktoré sú vyhodnocované pomocou metrick: strata, evaluačná strata a presnosť na testovacej množine počas každej epochy. Na základe týchto metrick bola následne vybraná doposiaľ najlepšia verzia modelu, ktorá bola manuálne vyhodnotená na testovacej množine 142 recenzií. Manuálne vyhodnotenie poskytne lepší obraz o použiteľnosti generatívnych jazykových modelov pre úlohu extrakcie štvoríc (aspekt, kategória, názor, sentiment), keďže správnosť vygenerovaného výstupu modelu je veľmi ťažké určiť len na základe porovnania s referenčným výstupom.



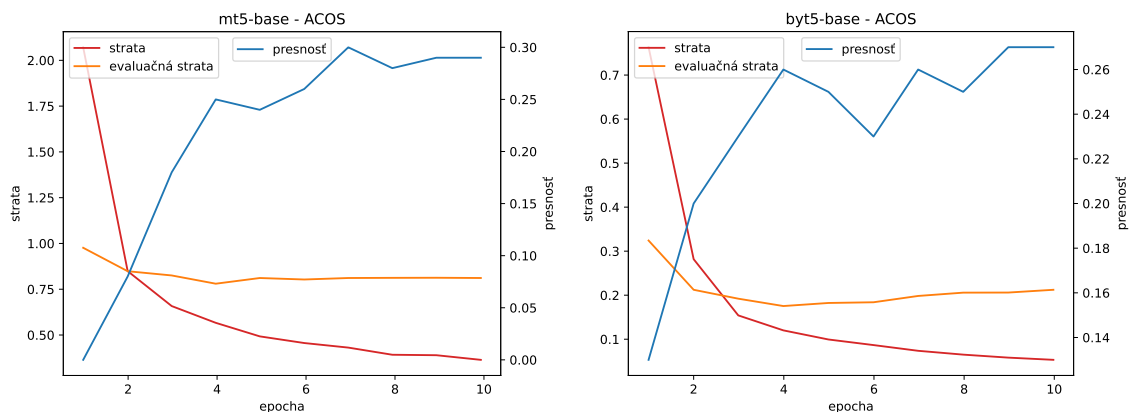
Obr. 5.1: Porovnanie vývoja straty a presnosti modelu mT5-base a ByT5-base v priebehu 10 epoch na úlohe extrakcie trojíc (aspekt, kategória, sentiment).

Ako štartovací bod boli zvolené predtrénované modely mT5-base a ByT5-base, dostupné na HuggingFace¹. Vybrané modely sú predtrénované na dátovej sade mC4 (Multilingual

¹<https://huggingface.co/models>

Colossal Clean Crawled Corpus)². Každý model bol ladený po dobu 10 epoch, pričom sa po každej epoche vyhodnotila jeho presnosť na testovacej množine. Na ladenie bola použitá trieda `Seq2SeqTrainer` z knižnice `transformers`³. Strata sa počíta na základe krížovej entropie medzi každým prvkom vygenerovanej sekvencie a prvkom na rovnakom indexe v referenčnej sekvencii (zlatá značka). Presnosť bola pri tomto vyhodnotení určovaná stratégiou presnej zhody vygenerovanej sekvencie s referenčným výstupom postupne po jednotlivých prvkoch štvoric (resp. trojíc). Nevýhodou tejto stratégie je to, že generatívny model častokrát vygeneruje výraz pre aspekt (resp. názor), ktorý má rovnaký význam ako referenčný aspekt (resp. názor), no aj takáto predikcia je vyhodnotená ako zlá. Preto je výsledná presnosť nižšia.

Modely boli doladené na vlastných dátových sadách, ktoré boli pripravené pre úlohu extrakcie aspektových trojíc (aspekt, kategória, sentiment) a štvoric (aspekt, kategória, názor, sentiment). Pre vytvorenie trénovacej množiny pre úlohu extrakcie trojíc boli použité vzorky z oboch dátových sád, takže obsahuje celkom 1217 recenzií (bližšie informácie o dátových sádach je možné nájsť v tabuľkách 3.3 a 3.2). Trénovacia množina pre úlohu extrakcie štvoric je menšia a tvorí ju 827 recenzií.



Obr. 5.2: Porovnanie vývoja straty a presnosti modelu mT5-base a ByT5-base v priebehu 10 epoch na úlohe extrakcie štvoric (aspekt, kategória, názor, sentiment).

Výsledky porovnaní, ktoré je možné vidieť na obrázkoch 5.1 a 5.2, ukazujú veľmi podobný vývoj metrik oboch modelov. Strata modelu ByT5 je nižšia, no presnosť na testovacej množine je porovnateľná s modelom mT5. Na základe týchto experimentov bol pre úlohu extrakcie štvoric vybraný model mT5, ktorý je menej náročný na výpočetné zdroje [29].

Model bol ďalej manuálne vyhodnotený na množine 142 recenzií. Presnosť (accuracy) modelu pre extrakciu celej štvorice je 60,13% a presnosť extrakcie trojice je 63,92%. Pri extrakcii názoru a aspektu je častou chybou rozdelenie zložitejšieho názoru na viacero aspektov. Napríklad v recenzii „*Síce se jedná o starší hru, ale stále má své kouzlo.*“ sú extrahované dva aspekty „*hra*“ a „*kouzlo*“, no extrahovaný by mal byť len aspekt reprezentovaný štvoricou (hra, celková spokojnosť, starší ale stále má své kouzlo, pozitívny). Tento typ chyby sa vyskytuje v 34% prípadov z celkového počtu 82 prípadov, kedy bol zle extrahovaný názor alebo aspekt. Ďalšou chybou pri extrakcii je vynechanie aspektov v recenzii obsahujúcej mnoho vymenovaných aspektov. Príkladom je recenzia „*super gameplay kom-*

²<https://huggingface.co/datasets/mc4>

³<https://huggingface.co/docs/transformers>

bat obsah mapy“, kde model extrahoval len aspekt „gameplay“ a „kombat“ a určil rovnaký názor pre oba aspekty („super“).

Tabuľka 5.1: Vyhodnotenie klasifikácie pre jednotlivé triedy kategórie aspektu a sentimentu

	Presnosť	Precíznosť	Úplnosť	F1	Počet
Celková spokojnosť	95,47%	90,24%	82,22%	86,05%	41
Hratelnosť	94,34%	93,52%	92,66%	93,09%	108
Optimalizácia a chyby	95,47%	90,38%	87,04%	88,68%	52
Komunita	99,62%	50,00%	100,00%	66,67%	2
Cena	98,11%	83,33%	100,00%	90,91%	30
Audiovizuál	98,11%	90,63%	93,55%	92,06%	32
Pozitívny	89,43%	84,75%	90,91%	87,72%	118
Negatívny	90,94%	96,32%	87,33%	91,61%	136
Neutrálny	96,23%	27,27%	60,00%	37,50%	11

Podrobnejšie metriky výkonnosti modelu pri klasifikácii kategórie aspektu a sentimentu je možné vidieť v tabuľke 5.1. Na základe týchto výsledkov sa dá usúdiť, že model pri klasifikácii kategórie aspektu dosahuje dobré výsledky, no nevie uspokojivo klasifikovať neutrálny sentiment. Pre zlepšenie klasifikácie tohto sentimentu je potrebné získať viac recenzií spomínajúcich neutrálne aspekty, keďže počet neutrálnych aspektov je v sade pre extrakciu štvoríc veľmi nevyvážený oproti pozitívnym a negatívnym aspektom (ako je možné vidieť v tabuľke 3.2). Zlé klasifikácie triedy s nízkym zastúpením potvrdzujú aj makro priemerované hodnoty metrik pre klasifikáciu sentimentu (tabuľka 5.2), ktoré sú výrazne nižšie ako mikro priemerované hodnoty.

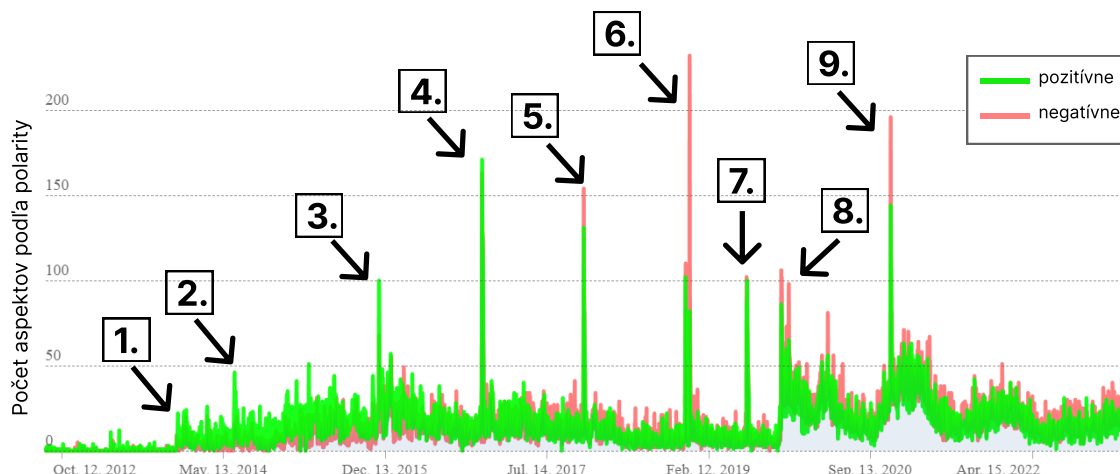
Tabuľka 5.2: Vyhodnotenie klasifikácie kategórie a sentimentu pomocou mikro a makro priemerovania

		Precíznosť	Úplnosť	F1
Kategória	mikro	90,57%	90,57%	90,57%
	makro	83,02%	92,58%	86,24%
Sentiment	mikro	88,30%	88,30%	88,30%
	makro	79,41%	69,45%	72,28%

5.2 Analýza trendov a porovnanie s realitou

Cieľom tejto analýzy a porovnania je zistiť, či je možné vytvorený systém a zvolené vizualizácie použiť na identifikovanie významných udalostí týkajúcich sa konkrétnej videohry. Príklady takýchto udalostí môžu zahŕňať aktualizácie obsahu hry, kontroverzné vyjadrenia zo strany vývojárov alebo prestížne turnaje, ktoré pritiahnu pozornosť hráčskej komunity. Pre toto porovnanie bola vybraná hra Counter-Strike: Global Offensive. Táto hra je podľa počtu 87266 analyzovaných recenzií najviac recenzovanou hrou českej komunity hráčov na platforme Steam. Preto môže poskytnúť pohľad na zaujímavé trendy.

Z prieskumu článkov, poznámok na blogu vývojárov a turnajov v čase zodpovedajúcom jednotlivým označeným vrcholom na obrázku 5.3 vyplýva, že 5 z 9 vrcholov súvisí s vý-



Obr. 5.3: Na grafe je označených 9 udalostí, ktoré mali zásadný vplyv na počet recenzií a polaritu sentimentu v nich spomínaných aspektov. Ďalej si je možné všimnúť zvýšený počet negatívnych aspektov po veľkej aktualizácii hry (6. vrchol), ktorý pretrváva aj v súčasnosti.

Tabuľka 5.3: Legenda udalostí

Udalosť	Dátum
1. Výpredaj hier na platforme Steam	27. november 2013
2. Výpredaj hier na platforme Steam	19. jún 2014
3. Finále turnaja Frankfurt Major 2015 (Dota 2)	21. november 2015
4. Výpredaj hier na platforme Steam	23. november 2016
5. Výpredaj hier na platforme Steam	22. november 2017
6. Danger Zone a free-to-play zmena	6. december 2018
7. ESL One Cologne 2019	2.–7. júl 2019
8. Zákaz obchodovania s populárnym predmetom	30. október 2019
9. Výpredaj hier na platforme Steam	25. november 2020

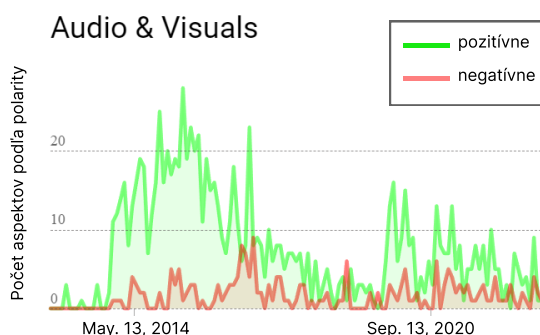
predajom hier na platforme Steam. Na základe tohto zistenia sa dá usúdiť, že takéto akcie majú veľký podiel na aktivite hráčov. Zvýšenie počtu recenzií pripisujem novým hráčom, ktorí sa vďaka zvýhodnenej cene rozhodli hru vyskúšať. V prislúchajúcich recenziách sa akcia a cena vôbec nespomínajú a vyjadrujú sa len o hre a jej aspektoch. Zaujímavým je 3. vrchol, ktorého časové obdobie zodpovedá finále prvého oficiálneho turnaja v hre Dota 2. Dota 2 a Counter-Strike: Global Offensive zdieľajú rovnakého vydavateľa. Ďalším vrcholom zodpovedajúcim prestížnemu turnaju je 7. vrchol. Zvýšený počet recenzií sa aj v týchto prípadoch dá pripísať novým hráčom. Zaujímavé je taktiež, že tieto recenzie turnaje nespomínajú.

Výnimkou sú 6. a 8. vrchol, ktoré reagujú na podstatné zmeny v hre. 6. vrchol sa týka veľkej aktualizácie, ktorá priniesla veľa nového obsahu a taktiež urobila hru dostupnou zdarma. To zvýšilo počet hráčov a taktiež aj recenzií, no tie boli veľmi negatívne. Najspomínanejšími boli kategórie: celková spokojnosť, optimalizácia a chyby, hrateľnosť (presnú štatistiku je možné vidieť v tabuľke 5.4). Používatelia mali negatívny postoj k sprístupneniu

Tabuľka 5.4: Počet aspektov podľa kategórie a polaritu sentimentu pre 6. vrchol

Kategória	Pozitívne	Negatívne
Celkom	82 (26,11%)	232 (73,89%)
Celková spokojnosť	44 (14,01%)	99 (31,53%)
Optimalizácia a chyby	9 (2,87%)	64 (20,38%)
Hratelnosť	27 (8,60%)	59 (18,79%)
Komunita	2 (0,64%)	5 (1,59%)
Audiovizuál	0 (0,00%)	5 (1,59%)

hry zdarma a najmä k zlej technickej stránke hry po aktualizácii⁴. Vrchol 8. udalosti presne zodpovedá novej aktualizácii, v ktorej vývojári zásadne zasiahli do trhu s predmetmi v hre. To vyvolalo zvýšenú aktivitu komunity na sociálnych sieťach, no používatelia sa k tejto aktualizácii v recenziách nevyjadrujú. Tento fakt naznačuje, že znova ide o nových hráčov.



Obr. 5.4: Graf časového vývoja sentimentu pre audiovizuálnu stránku hry.

V grafoch časového vývoja sentimentu pre kategórie je možné vidieť detailnejšie trendy v recenziách. Pri vybranej hre je zaujímavým vývoj sentimentu pre audiovizuálnu kategóriu (obrázok 5.4), kde je jednoznačne viditeľný klesajúci trend pozitívneho hodnotenia tejto kategórie. To naznačuje, že audiovizuálna stránka hry bola v dobe vydania na veľmi dobrej úrovni, no postupom času sa už nevyvíjala, a preto ju hráči v recenziách už často nechválili.

Porovnanie číselného hodnotenia hry

Vypočítané skóre hier bolo porovnané s číselným hodnotením na webovej stránke Metacritic⁵, ktorá poskytuje číselné hodnotenie na škále od 0,0 do 10,0 bodov.

Porovnať bolo možné 63 zo 68 analyzovaných hier, ktoré boli hodnotené používateľmi webovej stránky Metacritic. Metacritic poskytuje aj vlastné hodnotenie, ktoré je založené na recenziách videoherných kritikov. Takéto hodnotenie malo 53 zo 68 hier. Priemerná odchýlka medzi vypočítaným skóre a skóre Metacritic bola 1,6 bodu. Odchýlka medzi vypočítaným skóre a používateľským skóre bola 1,2 bodu (výsledky porovnania sú v tabuľke 5.5).

⁴<https://www.pcgamer.com/csgo-receives-14000-negative-steam-reviews-in-a-single-day-after-going-free-to-play/>

⁵<https://www.metacritic.com>

Δ	Metacritic	Používatelia
premier	1,6	1,2
medián	1,6	1,0
min	0,0	0,0
max	4,0	3,5

Tabuľka 5.5: Porovnanie rozdielu vypočítaného skóre so skóre na web stránke Metacritic

Zhodnotenie výsledkov

Z výsledkov prieskumu vyplýva, že vytvorený systém sa dá efektívne využiť na identifikáciu postojov hráčov voči rozhodnutiam vývojárov, aktualizáciám či iným udalostiam, a to aj na úrovni vopred stanovených kategórií aspektov. Číselné skóre hry taktiež zodpovedá realite a poskytuje používateľom rýchle zhodnotenie kvality hry.

5.3 Testovanie používateľského rozhrania

Cieľom testovania bolo zistiť intuitívnosť navigácie, jednoduchosť použitia filtrov, zoradenia a vyhľadávania v zozname hier. Ďalším cieľom bolo určiť, či sú jednotlivé grafy a vizualizácie zrozumiteľné a či sa s nimi dá ľahko pracovať. Testovanie prebiehalo formou osobného stretnutia s účastníkmi testovania a zúčastnilo sa ho 5 osôb. Použitý bol prehliadač Google Chrome vo verzii 112.0 a aplikácia bola dostupná na webovej adrese <https://athena1.fit.vutbr.cz:6769>.

Predstavenie účastníkov testovania a spätná väzba

Prvý účastník (20-25, vysokoškolské vzdelanie) má veľký záujem o videohry a pravidelne navštevuje webové stránky ako Metacritic a Steam. Recenzie sú pre neho veľmi dôležité pri rozhodovaní sa, či si hru vyskúša. Účastník uviedol, že získal všetky potrebné informácie z vizualizácií, bol spokojný s aplikáciou a nemal žiadne návrhy na zlepšenie.

Druhý účastník (25-30, vysokoškolské vzdelanie) má tiež veľký záujem o videohry a pravidelne navštevuje webové stránky zamerané na videohry. Recenzie považuje za dôležitý zdroj informácií pri rozhodovaní sa, či si hru vyskúša. V spätnej väzbe uviedol, že získal užitočné informácie z vizualizácií, ako sú skóre rôznych aspektov hry a sentiment komunity k týmto aspektom. Ďalej sa mu páčilo, že na základe grafu časového vývoja kategórie optimalizácia a chyby, bol schopný detegovať výpadky alebo zlé aktualizácie hry. Navrhoval pridať a filtrovať viac hodnôt na grafe vývoja sentimentu.

Tretí účastník (25-30, stredoškolské vzdelanie) má mierny záujem o videohry a občas navštevuje webové stránky zamerané na videohry. Pre neho sú recenzie dôležité pri rozhodovaní sa o vyskúšaní hry. Bol spokojný s informáciami získanými z vizualizácií, ako napríklad ako sa vývojári starajú o jednotlivé aspekty hry. K slovnému oblaku ďalej uviedol, že ľahko poukázal na obsah hry a taktiež aj humor recenzentov. Chválil jednoduchosť a zreteľnosť stránky a navrhoval pridať možnosť filtrovania hier podľa kategórie.

Štvrtý účastník (20-25, vysokoškolské vzdelanie) má mierny záujem o videohry a občas navštevuje webové stránky zamerané na videohry. Pre neho sú recenzie dôležité pri rozhodovaní sa o vyskúšaní hry. V spätnej väzbe uviedol, že získal užitočné informácie z vizualizácií, napríklad ako sa hra časovo rozvíjala. Páčilo sa mu, že vedel zistiť, aké konkrétne aspekty

hry vývojári zlepšujú a ako sú komunitou prijaté. Účastník tiež uviedol, že mu vizualizácie pomohli rozhodnúť sa, či si hru kúpi.

Piaty účastník (20-25, stredoškolské vzdelanie) má mierny záujem o videohry a občas navštevuje webové stránky zamerané na videohry. Pre neho sú recenzie dôležité pri rozhodovaní sa o vyskúšaní hry. Tento účastník uviedol, že na základe vizualizácií zistil silné a slabé stránky hier a ako sa časom zlepšovali. Zo slovného oblaku vedel vyčítať, aké boli chválené časti konkrétneho obsahu hry a na základe grafov časového vývoja zistil presný dátum nepodarenej aktualizácie, čo považoval za veľmi užitočné. Používateľ uviedol, že narážal na ťažkosti pri pochopení niektorých grafov a navrhoval zlepšenia. Najviac mu chýba možnosť filtrovania hier podľa kategórie a taktiež by zjednodušil grafy časového vývoja sentimentu, pri ktorých by pridal aj podrobné opisy.

Zhrnutie výsledkov testovania

Súčasťou spätnej väzby boli aj otázky, ktoré si vyžadovali ohodnotiť určitú časť používateľského rozhrania webovej aplikácie na škále od 1 do 5, pričom ak nie je inak uvedené, 1 znamená veľmi nespokojný a 5 veľmi spokojný:

1. Ako by ste ohodnotili intuitívnosť navigácie v rámci webovej aplikácie? (1–5)
2. Ako by ste ohodnotili jednoduchosť použitia filtrov pri výbere hier zo zoznamu? (1–5)
3. Ako by ste ohodnotili jednoduchosť použitia zoradenia pre zoznam hier? (1–5)
4. Na škále od 1 do 5, kde 1 znamená veľmi zrozumiteľné a 5 veľmi nezrozumiteľné, ako zrozumiteľné sú pre vás grafy a vizualizácie v podrobnom zobrazení hry?
5. Ako by ste zhodnotili celkovú kvalitu vizualizácií a grafov v podrobnom zobrazení hry? Ohodnoťte na škále od 1 do 5, kde 1 znamená veľmi nízka kvalita a 5 veľmi vysoká kvalita.
6. Ako dôležité sú pre vás vizualizácie a grafy pri rozhodovaní sa o výbere hier? Ohodnoťte na škále od 1 do 5, kde 1 znamená veľmi nezáleží a 5 veľmi záleží.

Priemerné hodnotenie jednotlivých aspektov používateľského rozhrania je možné vidieť v tabuľke 5.6.

Číslo otázky	Hodnotený aspekt	Priemerné skóre
1.	intuitívnosť navigácie	4.6
2.	jednoduchosť použitia filtrov	4.2
3.	jednoduchosť použitia zoradenia	4.8
4.	zrozumiteľnosť grafov a vizualizácií	3.8
5.	celková kvalita vizualizácií a grafov	4.0
6.	dôležitosť vizualizácií a grafov pri rozhodovaní	3.6

Tabuľka 5.6: Výsledok hodnotenia aspektov webovej aplikácie respondentmi

Na základe spätnej väzby sa dá povedať, že webová aplikácia poskytuje intuitívnu navigáciu a jednoduché nástroje pre vyhľadávanie, filtrovanie a radenie hier. Respondenti považovali zvolené grafické vizualizácie za zrozumiteľné a ľahko pochopiteľné, no niektorým chýbali detailné popisy vizualizácií. Taktiež by uvítali rozšírenie možností filtrovania o ďalšie kritéria.

Kapitola 6

Záver

Cieľom tejto práce bolo navrhnúť a implementovať systém pre aspektovo-orientovanú analýzu sentimentu používateľských recenzií videohier. Výsledky analýzy by mali používateľom poskytnúť bližšie informácie o hrách a ich aspektoch, pomocou vhodne zvolených vizualizácií. Problematika aspektovo-orientovanej analýzy si vyžadovala preskúmanie mnohých úloh a ich súčasných riešení pomocou strojového učenia. Na základe požiadaviek na to, čo by malo byť výstupom analýzy a aké informácie o hrách by mal systém poskytovať, sa táto práca zamerala na úlohu extrakcie štvoríc (aspekt, kategória, názor, sentiment). Na základe prieskumu súčasných metód bola k riešeniu tejto úlohy vybraná metóda založená na inštrukčnom učení a generatívnych jazykových modeloch.

Výsledkom tejto práce je funkčný systém, ktorý automatizuje proces získavania, spracovania a analýzy recenzií videohier. Analýza je vykonávaná predtrénovaným modelom mT5¹, ktorý bol doladený na vlastnej dátovej sade recenzií videohier určenej pre úlohu extrakcie aspektových štvoríc. Na testovacej sade dosiahol presnosť 60%. Systém navyše ponúka webovú aplikáciu, ktorá zabezpečuje vizualizácie výstupov analýzy pomocou intuitívneho používateľského rozhrania. Vizualizácie poskytujú ucelený pohľad na silné a slabé stránky konkrétnej videohry pomocou číselného hodnotenia a slovných oblakov najviac spomínaných aspektov. Ďalšou súčasťou vizualizácií sú interaktívne grafy časového vývoja sentimentu. Tie umožňujú používateľom sledovať oblasti, v ktorých sa hra zlepšovala alebo naopak zhoršovala v priebehu času. Taktiež sa pomocou nich dajú identifikovať zlé aktualizácie alebo iné zmeny, ktoré zásadne ovplyvňujú spokojnosť hráčov.

Na základe experimentov a porovnávania informácií získaných analýzou vizualizácií s realitou, sa dá konštatovať, že navrhnutý systém splnil stanovené ciele tejto práce. Avšak vývoj takéhoto systému je neustály proces a existuje priestor pre ďalšie zlepšenia a rozšírenia. Jedným zo spôsobov, ako by sa systém mohol zlepšiť, je zvýšiť úspešnosť modelu v klasifikácií aspektových štvoríc. To by bolo možné dosiahnuť zlepšením dátovej sady, v ktorej by bolo potrebné vyrovnať rozdiely medzi nevyváženými triedami aspektov a odstrániť nejednoznačné vzorky. Taktiež by bolo vhodné klasifikovať aspekty do jemnejších kategórií. Medzi možné budúce smerovanie práce patrí kombinovanie aspektovo-orientovanej analýzy sentimentu s ďalšími zdrojmi informácií, ako sú napríklad dáta o predajnosti hier, počte hráčov alebo štatistiky zo sociálnych sietí. Takéto spojenie môže priniesť ešte presnejší a komplexnejší obraz o stave a oblúbenosti videohier.

¹<https://huggingface.co/google/mt5-base>

Literatúra

- [1] ADALOGLOU, N. *Understanding the receptive field of deep convolutional networks* [online]. 2020 [cit. 2023-03-28]. Dostupné z: <https://theaisummer.com/receptive-field/>.
- [2] BA, J., KIROS, J. R. a HINTON, G. E. *Layer Normalization* [online]. 2016, Revidované 21. 7. 2016 [cit. 2023-04-16]. Dostupné z: <https://doi.org/10.48550/arXiv.1607.06450>.
- [3] BAHDANAU, D., CHO, K. a BENGIO, Y. *Neural Machine Translation by Jointly Learning to Align and Translate* [online]. September 2014, Revidované 19. 5. 2016 [cit. 2023-04-18]. Dostupné z: <https://arxiv.org/abs/1409.0473>.
- [4] CAI, H., XIA, R. a YU, J. Aspect-Category-Opinion-Sentiment Quadruple Extraction with Implicit Aspects and Opinions. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* [online]. Online: Association for Computational Linguistics, August 2021, s. 340–350. DOI: 10.18653/v1/2021.acl-long.29. Dostupné z: <https://aclanthology.org/2021.acl-long.29>.
- [5] DEVLIN, J., CHANG, M., LEE, K. a TOUTANOVA, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [online]. 2018, Revidované 24. 5. 2019. Dostupné z: <http://arxiv.org/abs/1810.04805>.
- [6] GAO, T., FANG, J., LIU, H., LIU, Z., LIU, C. et al. LEGO-ABSA: A Prompt-based Task Assemblable Unified Generative Framework for Multi-task Aspect-based Sentiment Analysis. In: *Proceedings of the 29th International Conference on Computational Linguistics* [online]. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Október 2022, s. 7002–7012 [cit. 2023-03-26]. Dostupné z: <https://aclanthology.org/2022.coling-1.610>.
- [7] GLEZ PEÑA, D., LOURENÇO, A., LÓPEZ FERNÁNDEZ, H., REBOIRO JATO, M. a FDEZ RIVEROLA, F. Web scraping technologies in an API world. *Briefings in Bioinformatics*. Apríl 2013, zv. 15, č. 5, s. 788–797. DOI: 10.1093/bib/bbt026. ISSN 1467-5463. Dostupné z: <https://doi.org/10.1093/bib/bbt026>.
- [8] GOODFELLOW, I., BENGIO, Y. a COURVILLE, A. Deep Learning. In: MIT Press, November 2016, s. 526. Adaptive Computation and Machine Learning series. ISBN 9780262035613.
- [9] IOFFE, S. a SZEGEDY, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* [online]. Február 2015, Revidované 2. 3. 2015 [cit. 2023-04-16]. Dostupné z: <https://doi.org/10.48550/arXiv.1502.03167>.

- [10] LIN, D., BEZEMER, C.-P., ZOU, Y. a HASSAN, A. E. An Empirical Study of Game Reviews on the Steam Platform. *Empirical Software Engineering*. 1. vyd. Berlín, Nemecko: Springer Nature. Február 2019, zv. 24, s. 170–207. DOI: 10.1007/s10664-018-9627-4. ISSN 1573-7616. Dostupné z: <https://doi.org/10.1007/s10664-018-9627-4>.
- [11] LIU, B. *Sentiment Analysis and Opinion Mining*. San Rafael, CA: Morgan & Claypool Publishers, máj 2012. Synthesis Lectures on Human Language Technologies, č. 1. ISBN 978-3-031-02145-9. Dostupné z: <http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016>.
- [12] LIU, J., TENG, Z., CUI, L., LIU, H. a ZHANG, Y. Solving Aspect Category Sentiment Analysis as a Text Generation Task. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* [online]. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, November 2021, s. 4406–4416 [cit. 2023-03-23]. DOI: 10.18653/v1/2021.emnlp-main.361. Dostupné z: <https://aclanthology.org/2021.emnlp-main.361>.
- [13] LIU, N. a ZHAO, J. A BERT-Based Aspect-Level Sentiment Analysis Algorithm for Cross-Domain Text. *Computational Intelligence and Neuroscience*. Hindawi. 2022, zv. 2022, s. 8726621, [cit. 2023-03-24]. DOI: 10.1155/2022/8726621. ISSN 1687-5273. Dostupné z: <https://doi.org/10.1155/2022/8726621>.
- [14] MEDHAT, W., HASSAN, A. a KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* [online]. 2014, zv. 5, č. 4, s. 1093–1113, Revidované 8. 4. 2014, [cit. 2023-03-16]. DOI: 10.1016/j.asej.2014.04.011. ISSN 2090-4479. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S2090447914000550>.
- [15] MITCHELL, R. Web Scraping with Python: Collecting More Data from the Modern Web. In: 2. vyd. USA: O’Reilly Media, Apríl 2018, s. 175–180. ISBN 978-1491985571.
- [16] MUHAMMAD, A., WIRATUNGA, N. a LOTHIAN, R. Contextual sentiment analysis for social media genres. *Knowledge-Based Systems*. 2016, zv. 108, s. 92–101. DOI: 10.1016/j.knosys.2016.05.032. ISSN 0950-7051. New Avenues in Knowledge Bases for Natural Language Processing.
- [17] NI, R. a CAO, H. Sentiment Analysis based on GloVe and LSTM-GRU. In: *2020 39th Chinese Control Conference (CCC)*. 2020, s. 7492–7497. DOI: 10.23919/CCC50068.2020.9188578.
- [18] PEPPER, J. a WANG, L. Generative Aspect-Based Sentiment Analysis with Contrastive Learning and Expressive Structure. In: *Findings of the Association for Computational Linguistics: EMNLP 2022* [online]. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, December 2022, s. 6089–6095 [cit. 2023-04-16]. Dostupné z: <https://aclanthology.org/2022.findings-emnlp.451>.
- [19] PONTIKI, M., GALANIS, D., PAVLOPOULOS, J., PAPAGEORGIU, H., ANDROUTSOPOULOS, I. et al. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In: *Proceedings of the 8th International Workshop on Semantic Evaluation*

- (*SemEval 2014*) [online]. Dublin, Ireland: Association for Computational Linguistics, August 2014, s. 27–35 [cit. 2023-03-21]. DOI: 10.3115/v1/S14-2004. Dostupné z: <https://aclanthology.org/S14-2004>.
- [20] QIU, G., LIU, B., BU, J. a CHEN, C. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*. Cambridge, MA: MIT Press. Marec 2011, zv. 37, č. 1, s. 9–27. DOI: 10.1162/coli_a_00034. Dostupné z: <https://aclanthology.org/J11-1002>.
- [21] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S. et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* [online]. 2020 [cit. 2023-04-18]. Dostupné z: <https://doi.org/10.48550/arXiv.1910.10683>.
- [22] SCARIA, K., GUPTA, H., SAWANT, S. A., MISHRA, S. a BARAL, C. *InstructABSA: Instruction Learning for Aspect Based Sentiment Analysis* [online]. Február 2023 [cit. 2023-03-21]. Dostupné z: <https://arxiv.org/abs/2302.08624>.
- [23] SORIA OLIVAS, E., GUERRERO, J. D. M., MARTINEZ SOBER, M., MAGDALENA BENEDITO, J. R. a SERRANO LOPEZ, A. J., ed. *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques*. Hershey, PA: Information Science Reference, august 2009.
- [24] URRIZA, I. M. a CLARIÑO, M. A. A. Aspect-Based Sentiment Analysis of User Created Game Reviews. In: *2021 24th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*. 1. vyd. Singapur, Singapur: IEEE, 2021, sv. 1, s. 76–81. DOI: 10.1109/O-COCOSDA202152914.2021.9660559. ISSN 2472-7695. Dostupné z: <https://ieeexplore.ieee.org/document/9660559>.
- [25] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L. et al. Attention is All you Need. In: GUYON, I., LUXBURG, U. V., BENGIO, S., WALLACH, H., FERGUS, R. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, sv. 30. Dostupné z: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [26] WANG, Y., HUANG, M., ZHU, X. a ZHAO, L. Attention-based LSTM for Aspect-level Sentiment Classification. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, s. 606–615. DOI: 10.18653/v1/D16-1058. Dostupné z: <https://aclanthology.org/D16-1058>.
- [27] WILSON, T., WIEBE, J. a HOFFMANN, P. Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. *Computational Linguistics* [online]. September 2009, zv. 35, č. 3, s. 399–433, Revidované 8. 3. 2008, [cit. 2023-03-23]. DOI: 10.1162/coli.08-012-R1-06-90. ISSN 0891-2017. Dostupné z: <https://doi.org/10.1162/coli.08-012-R1-06-90>.
- [28] XU, L., BING, L., LU, W. a HUANG, F. Aspect Sentiment Classification with Aspect-Specific Opinion Spans. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* [online]. Online: Association for Computational Linguistics, November 2020, s. 3561–3567, Revidované 16. 11. 2020

- [cit. 2023-03-23]. DOI: 10.18653/v1/2020.emnlp-main.288. Dostupné z: <https://aclanthology.org/2020.emnlp-main.288>.
- [29] XUE, L., BARUA, A., CONSTANT, N., AL RFOU, R., NARANG, S. et al. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics* [online]. Marec 2022, zv. 10, s. 291–306, Revidované 8. 3. 2022, [cit. 2023-04-16]. DOI: 10.1162/tacl_a_00461. ISSN 2307-387X. Dostupné z: https://doi.org/10.1162/tacl_a_00461.
- [30] XUE, W. a LI, T. Aspect Based Sentiment Analysis with Gated Convolutional Networks. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* [online]. Melbourne, Australia: Association for Computational Linguistics, Júl 2018, s. 2514–2523 [cit. 2023-03-23]. DOI: 10.18653/v1/P18-1234. Dostupné z: <https://aclanthology.org/P18-1234>.
- [31] XUE, W. a LI, T. Aspect Based Sentiment Analysis with Gated Convolutional Networks. *CoRR* [online]. 2018, abs/1805.07043, [cit. 2023-04-18]. Dostupné z: <http://arxiv.org/abs/1805.07043>.
- [32] YANG, H. a LI, K. *PyABSA: Open Framework for Aspect-based Sentiment Analysis* [online]. 2022, Revidované 23. 2. 2023 [cit. 2023-03-23]. Dostupné z: <https://arxiv.org/abs/2208.01368>.
- [33] YOSINSKI, J., CLUNE, J., BENGIO, Y. a LIPSON, H. *How transferable are features in deep neural networks?* [online]. 2014 [cit. 2023-03-20]. Dostupné z: <http://arxiv.org/abs/1411.1792>.
- [34] ZHANG, L., WANG, S. a LIU, B. *Deep Learning for Sentiment Analysis : A Survey* [online]. 2018, Revidované 30. 1. 2018 [cit. 2023-03-23]. Dostupné z: <http://arxiv.org/abs/1801.07883>.

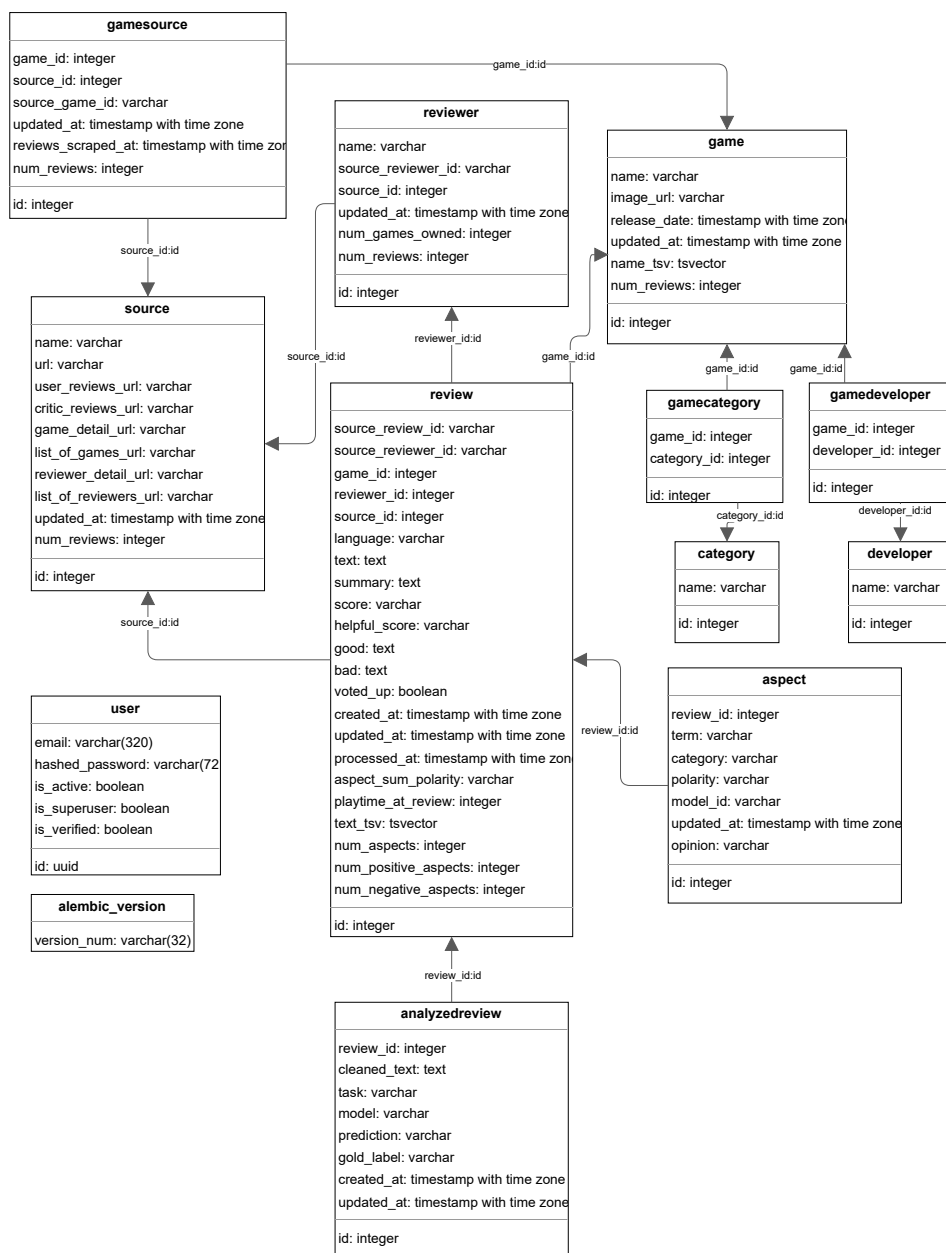
Príloha A

Obsah priloženého pamäťového média

- `/tex/` – adresár obsahujúci zdrojové súbory technickej správy
- `/xsabo103.pdf` – technická správa vo formáte PDF
- `/app/` – adresár obsahujúci zdrojové kódy aplikácie
- `/readme.txt` – popis aplikácie a návod na inštaláciu

Príloha B

Relačný model databázy



Príloha C

Plagát

Aspektově orientovaná analýza postojů z uživatelských recenzí počítačových her

František Sabol

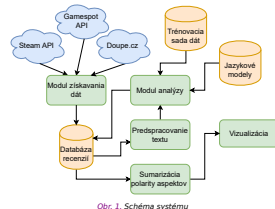
Vedúci práce: doc. RNDr. Pavel Smrč, Ph.D.

Intuitívne používateľské rozhranie a vizualizácia výsledkov aspektovo-orientovanej analýzy



Ciele práce

Cieľom práce bolo vytvoriť systém, ktorý bude zhromažďovať používateľské recenzie videohier z distribučných platforiem. Tie následne analyzuje na úrovni aspektu a používateľom poskytne prehľadnú vizualizáciu výsledkov analýzy.



Obr. 1. Schéma systému

Prínos pre používateľa

Webová aplikácia a vizualizácie pomáhajú získať rýchly prehľad o názore komunity na silné alebo slabé stránky hry.

Extrakcia aspektových štvorcí

Systém aplikuje generatívne jazykové modely na extrakciu štvorcí (aspekt, kategória, názor, sentiment). Presnosť klasifikácie celej štvorice na testovacej množine dosahuje presnosť 60%.

Tabuľka 1. Vyhodnotenie klasifikácie kategórie a sentimentu pomocou mikro a makro priradení

		Precíznosť	Úplnosť	F1
Kategória	mikro	90,57%	90,57%	90,57%
	makro	83,02%	92,58%	86,24%
Sentiment	mikro	88,30%	88,30%	88,30%
	makro	79,41%	69,45%	72,28%

Formát použitých inštrukcií

