|  |  |
| --- | --- |
| **TECHNICKÁ UNIVERZITA V KOŠICIACH**  **FAKULTA ELEKTROTECHNIKY A INFORMATIKY** | |
| **ROZPOZNÁVANIE NENÁVISTNEJ REČI POMOCOU VEĽKÝCH JAZYKOVÝCH MODELOV**  **Bakalárska práca** | |
|  | |
| **2025** | **Tetiana Mohorian** |
| **TECHNICKÁ UNIVERZITA V KOŠICIACH**  **fakulta ELEKTROTECHIKY a INFORMATIKY** | |
| **Rozpoznávanie nenávistnej reči pomocou veľkých jazykových modelov**  **Bakalárska práca** | |
|  | |
| Študijný program: | Počítačové siete |
| Študijný odbor: | Názov odboru (pozri zadávací list) |
| Školiace pracovisko: | KEaMT |
| Školiteľ: | Ing. Daniel Hládek, PhD. |
| Konzultant: | Ing. Daniel Hládek, PhD. |
|  | |
| **2025 Košice** | **Tetiana Mohorian** |

**Abstrakt v SJ**

Tato bakalárská praca sa zaoberá významom problému nenávisti online. Skúma, ktoré modely sú najvhodnejšie na rozpoznávanie agresie, s cieľom vycvičiť model na takéto rozpoznávanie v budúcnosti. Výskum sa uskutočnil na modeloch BERT, GPT, T5 a DISTIL BERT. Vyhodnotenie modelov pomocou metrík Recall, Precision, F1 ukázalo značné rozdiely v ich výsledkoch. Na zvýšenie presnosti rozpoznávania sa použilo niekoľko metód záberov, jemného ladenia a LoRA. Na trénovanie a overovanie modelov sa použil súbor údajov „TUKE-KEMT/hate\_speech\_slovak“. Obsahuje texty s nenávistnými prejavmi rozdelené na samotný text a značku (0 - bez nenávisti, 1 - prítomnosť). Výskum ukázal, že modely majú rôznu presnosť pri kategorizácii nenávistných prejavov.

**Kľúčové slova v SJ**

Kľúčové slovo1, kľúčové slovo2, kľúčové slovo3, kľúčové slovo4,...

Kľúčové slová sú slová, ktoré úzko definujú tému práce a pomocou nich je možné jednoduchšie vyhľadať prácu podľa jej odborného významu.

**Abstrakt v AJ**

This bachelor thesis deals with the importance of the problem of online hate. It examines which models are best suited to recognize aggression, with the aim of training a model to recognize such recognition in the future. The research was conducted on the BERT, GPT, T5 and DISTIL BERT models. Evaluation of the models using Recall, Precision, F1 metrics showed significant differences in their results. Several shot, fine tuning and LoRA methods were used to improve the recognition accuracy. The dataset "TUKE-KEMT/hate\_speech\_slovak" was used to train and validate the models. It contains the hate-speech texts divided into the text itself and the tag (0 - no hate, 1 - present). Research has shown that the models have varying accuracy in categorizing hate speech.

**Kľúčové slova v AJ**

Key word 1, Key word 2, Key word 3, Key word 4,...

**Zadanie práce**

Túto stranu nahraďte naskenovaným zadávacím listom.

Odporúčame skenovať na 200-300 DPI, Odtiene sivej

! v jednej vytlačenej ZP musí byť vložený originál zadávacieho listu !

**Čestné vyhlásenie**

Vyhlasujem, že som celú dizertačnú prácu vypracoval/a samostatne s použitím uvedenej odbornej literatúry.

Košice, 21. marca 2025 ..........................................

vlastnoručný podpis

**Poďakovanie**

Na tomto mieste môže byť vyjadrenie poďakovania napr. vedúcemu práce resp. konzultantom za pripomienky a odbornú pomoc pri vypracovaní práce. Nie je zvykom ďakovať za rutinnú kontrolu, menšiu spoluprácu alebo všeobecné rady. Vyjadrenie poďakovania v prípade využitia inej práce sa uskutočňuje formou citácie na konci hlavného textu práce a odkazy na citáciu sa musia uviesť aj na zodpovedajúcich miestach v texte.

**Obsah**

[Zoznam obrázkov 11](#_Toc192198529)

[Zoznam tabuliek 12](#_Toc192198530)

[Zoznam symbolov a skratiek 13](#_Toc192198531)

[Úvod 14](#_Toc192198532)

[1. Formulácia úlohy a cieľ práce 16](#_Toc192198533)

[2. Teoretický rozbor zvolenej témy 18](#_Toc192198534)

[2.1. Hlboké učenie, jeho koncepty a využitie v rôznych priemyselných odvetviach 18](#_Toc192198535)

[2.1.1. Čo je to hlboké učenie? 18](#_Toc192198536)

[2.1.2. Využitie hlbokého učenia v rôznych priemyselných odvetviach 19](#_Toc192198537)

[2.1.3. Základy hlbokého učenia v NLP 20](#_Toc192198538)

[2.1.4. Transformerová architektúra a jej kľúčové vlastnosti 21](#_Toc192198539)

[2.1.5. Využitie LLM v NLP úlohách 22](#_Toc192198540)

[2.1.6. Výzvy a limity veľkých jazykových modelov 23](#_Toc192198541)

[2.2. Prehľad veľkých jazykových modelov pre slovenčinu (otvorené aj uzavreté) 24](#_Toc192198542)

[2.2.1. Kritériá hodnotenia jazykových modelov 24](#_Toc192198543)

[2.2.2. Otvorené modely trénované na slovenčine 25](#_Toc192198544)

[2.2.3. Multilingválne otvorené modely s podporou slovenčiny 26](#_Toc192198545)

[2.2.4. Uzavreté veľké jazykové modely a ich podpora slovenčiny 27](#_Toc192198546)

[2.2.5. Porovnanie modelov a ich vhodnosť pre NLP úlohy v slovenčine 28](#_Toc192198547)

[2.3. Klasifikácia textu – NLP úlohy 29](#_Toc192198548)

[2.3.1. Základy textovej klasifikácie v NLP 29](#_Toc192198549)

[2.3.2. Prompting ako prístup k textovej klasifikácii 30](#_Toc192198550)

[2.3.3. Adaptívne trénovanie modelov pre textovú klasifikáciu 31](#_Toc192198551)

[2.3.4. Porovnanie jednotlivých prístupov v klasifikácii textu 32](#_Toc192198552)

[2.4. Problémy a možné budúce zlepšenia 32](#_Toc192198553)

[2.4.1. Hlavné výzvy pri detekcii nenávistnej reči 32](#_Toc192198554)

[2.4.2. Možnosti zlepšenia jazykových modelov pri klasifikácii nenávistnej reči 33](#_Toc192198555)

[2.4.3. Adaptívne trénovanie a nové techniky modelovania 34](#_Toc192198556)

[2.4.4. Budúce smery výskumu v oblasti NLP a hate speech detection 34](#_Toc192198557)

[3. Analýza stavu problematiky 36](#_Toc192198558)

[3.1. Existujúce modely vhodné pre slovenčinu 36](#_Toc192198559)

[3.1.1. SlovakBERT 36](#_Toc192198560)

[3.1.2. SlovakT5 36](#_Toc192198561)

[3.1.3. Multilinguálne modely (mT5, XLM-R) 37](#_Toc192198562)

[3.1.4. Veľké uzavreté modely (GPT-4, Gemini) 37](#_Toc192198563)

[3.2. Prehľad datasetov pre hate speech detection v slovenčine 38](#_Toc192198564)

[3.2.1. TUKE-KEMT Hate Speech dataset 38](#_Toc192198565)

[3.2.2. Iné dostupné datasetové zdroje 39](#_Toc192198566)

[3.2.3. Vlastnosti a výzvy datasetov pre hate speech detection v slovenčiny 39](#_Toc192198567)

[3.3. Prehľad metód na hate speech detection 40](#_Toc192198568)

[3.3.1. Klasifikácia textu 40](#_Toc192198569)

[3.3.2. Prompting 41](#_Toc192198570)

[3.3.3. Fine-tuning (doladenie) 41](#_Toc192198571)

[4. Návrh a implementácia riešenia zvolenej problematiky 43](#_Toc192198572)

[4.1. Výber modelu a metódy 43](#_Toc192198573)

[4.2. Trénovanie vybraného modelu pomocou LoRA 43](#_Toc192198574)

[4.3. Aplikácia few-shot a prompting (príklady promptov) 43](#_Toc192198575)

[4.4. Vyhodnotenie pomocou metrik Precision, Recall, F1 43](#_Toc192198576)

[4.5. Prehľad experimentov a výsledkov (tabuľky, grafy) 43](#_Toc192198577)

[4.6. Vývoj webovej aplikácie a Telegram bota 43](#_Toc192198578)

[5. 44](#_Toc192198579)

[5.1. Vkladanie tabuliek do dokumentu 44](#_Toc192198580)

[5.2. Vkladanie grafov do dokumentu 44](#_Toc192198581)

[5.3. Krížové odkazy na použitú literatúru a webový obsah 45](#_Toc192198582)

[5.4. Citovanie z použitej literatúry a zoznam použitej literatúry 45](#_Toc192198583)

[5.4.1. Pribežné ukladanie informácií o použitej literatúre 47](#_Toc192198584)

[6. Postup pri odovzdávaní a tlači záverečnej práce 48](#_Toc192198585)

[6.1. Predregistrácia záverečnej práce 49](#_Toc192198586)

[Záver 50](#_Toc192198587)

[Zoznam použitej literatúry 51](#_Toc192198588)

[Prílohy 52](#_Toc192198589)

Zoznam obrázkov

[Obr. 1 Vkladanie popisu pre obrázok 4](#_Toc382997503)

[Obr. 2 Vkladanie popisu k tabuľkám 4](#_Toc382997504)

[Obr. 3 Porovnanie percenta zhody fakúlt pre Bc štúdium za rok 2013 4](#_Toc382997505)

[Obr. 4 Vzor protokolu o kontrole originality 4](#_Toc382997506)

[Obr. 5 Obrázok grafického CD média 4](#_Toc382997507)

Zoznam tabuliek

[Tab. 1 Štatistické zhodnotenie percenta zhody za rok 2013 4](#_Toc382997508)

Zoznam symbolov a skratiek

Tento zoznam je nepovinný. Vypĺňa sa len v prípade značiek a symbolov, ktoré nie sú štandardami a nepatria do SI sústavy veličín.

SI Systeme International

SDHC Secure Digital High Capacity

Úvod

Internet a sociálne siete sa stali neoddeliteľnou súčasťou moderného života. Poskytujú ľuďom pohodlné nástroje na komunikáciu, výmenu informácií a sebarealizáciu. Avšak spolu s pozitívnymi aspektmi rozvoja digitálnych technológií vznikli aj vážne hrozby spojené s obsahom zverejňovaného materiálu. Jednou z týchto hrozieb je šírenie nenávistnej reči (hate speech), ktorá predstavuje vážne riziko pre jednotlivcov aj pre spoločnosť ako celok. Tento problém nadobudol globálny rozmer, keďže nenávistná reč nie je obmedzená na jeden jazyk, región alebo kultúru, ale stáva sa významným problémom v rôznych častiach sveta [3].

Nenávistná reč zahŕňa akékoľvek verejné vyjadrenia obsahujúce prvky nenávisti, diskriminácie alebo urážok voči určitým osobám alebo skupinám na základe ich rasy, etnického pôvodu, náboženstva, pohlavia, sexuálnej orientácie, politických názorov alebo iných charakteristík. Môže sa prejavovať ako priame urážky, výzvy na násilie alebo formy skrytých urážok, ako napríklad vtipy, memy alebo sarkastické komentáre. Tieto výroky môžu mať vážne následky, vrátane zhoršenia medziľudských vzťahov, šírenia nenávisti a polarizácie spoločnosti [4].

Šírenie nenávistnej reči na sociálnych sieťach sa stalo obzvlášť výrazným v posledných rokoch. Podľa údajov medzinárodných organizácií približne 35 % internetových používateľov stretlo s nenávistnou rečou v online priestore. V niektorých krajinách je toto číslo ešte vyššie, najmä v regiónoch, kde existujú etnické alebo politické konflikty. Znepokojujúce je aj to, že nenávistná reč sa často šíri medzi mladými ľuďmi, ktorí sa s ňou stretávajú na sociálnych sieťach, fórach a v online diskusiách. Tento jav má negatívny vplyv na duševné zdravie jednotlivcov a môže prispieť k vzniku konfliktov a násilných činov [6].

Súčasný problém spočíva v tom, že manuálna moderácia obsahu je neefektívna, najmä pri veľkom množstve príspevkov, ktoré sa denne objavujú na rôznych platformách. Napríklad, iba na Facebooku sa denne zverejňuje viac ako 100 miliárd príspevkov. Vzhľadom na tieto obrovské objemy sa stáva prakticky nemožné, aby ľudia alebo tím moderátorov stíhali spracovávať všetok obsah v reálnom čase. Okrem toho, tradičné metódy ako filtrovanie na základe kľúčových slov sú často nedostatočné, pretože nenávistná reč môže byť skrytá v rôznych formách, ako sú vtipy, memes alebo slangu [9].

Rozpoznávanie nenávistnej reči sa stalo aktuálnym výskumným problémom a výzvou pre odborníkov v oblasti spracovania prirodzeného jazyka (NLP). Moderné prístupy sa zameriavajú na využívanie metód strojového učenia, a predovšetkým hlbokého učenia, ktoré umožňujú analyzovať text v širšom kontexte, zohľadniť emocionálny tón, ironické vyjadrenia a zložité jazykové konštrukcie. V tomto smere sa čoraz viac pozornosti venuje využívaniu veľkých jazykových modelov (Large Language Models, LLM), ako sú BERT, GPT, Mistral alebo T5. Tieto modely sú trénované na rozsiahlych textových korpusoch a sú schopné generovať alebo analyzovať texty s vysokou presnosťou a kontextovým pochopením [7].

Veľké jazykové modely majú oproti tradičným metódam filtrovania niekoľko výhod. Okrem schopnosti zachytiť jemné nuansy jazyka a kontextu, môžu tieto modely analyzovať aj skryté významy, ktoré nie sú zjavné na prvý pohľad. Napríklad, modely ako BERT alebo GPT dokážu rozpoznať skrytú nenávistnú reč v textoch, ktoré sa zdajú byť nevinné, ale obsahujú diskriminačné alebo urážlivé podtóny. Okrem toho tieto modely môžu byť efektívne prispôsobené na rôzne jazyky a kultúrne špecifiká, čo je veľmi dôležité v globálnom internetovom prostredí [9].

V súčasnosti sa vývoj takýchto systémov automatického rozpoznávania nenávistnej reči stáva prioritou pre viaceré organizácie a vlády. Ich cieľom je znížiť šírenie nenávistného obsahu na sociálnych sieťach, ktoré môže mať ničivé dôsledky na spoločenskú harmóniu. Tento výskum je obzvlášť dôležitý v krajinách strednej a východnej Európy, vrátane Slovenska a Ukrajiny, kde sa šírenie nenávistnej reči na internete stáva čoraz vážnejším problémom, najmä v súvislosti s politickými a etnickými napätiami [7].

Vývoj automatických systémov na detekciu nenávistnej reči pomocou veľkých jazykových modelov je tiež dôležitý v kontexte ochrany slobody prejavu. Je nevyhnutné vyvážiť potrebu potláčať nenávistnú reč s ochranou práv jednotlivcov na slobodu slova. Preto je potrebné vyvinúť sofistikované modely, ktoré nielen efektívne identifikujú nenávistné vyjadrenia, ale aj zohľadňujú rôzne jazykové, kultúrne a sociálne faktory [6].

Týmto sa vývoj a adaptácia metód na automatické rozpoznávanie nenávistnej reči s využitím veľkých jazykových modelov stáva nielen vedeckou výzvou, ale aj spoločensky významným problémom. Riešenie tejto problematiky prispeje k vytváraniu bezpečnejšieho a tolerantnejšieho informačného priestoru a pomôže predchádzať negatívnym dôsledkom, ktoré môže nenávistná reč spôsobiť v spoločnosti. [5]

Súčasné výzvy si vyžadujú implementáciu technológií, ktoré dokážu rýchlo a presne odhaliť prejavy nenávisti, pričom zabezpečujú rešpektovanie slobody slova a minimalizovanie falošných varovaní [4].

1. Formulácia úlohy a cieľ práce

V dnešnej dobe sa automatické rozpoznávanie nenávistnej reči (hate speech) stalo kľúčovým problémom v oblasti spracovania prirodzeného jazyka (NLP) a strojového učenia. Nenávistná reč sa čoraz častejšie objavuje na sociálnych sieťach, v diskusiách na internetových fórach, v komentároch pod článkami, a často vedie k násilným alebo diskriminačným činom. Preto je dôležité vyvinúť efektívne metódy, ktoré umožnia automaticky identifikovať takéto prejavy a tým prispieť k vytváraniu bezpečnejšieho online prostredia. Táto bakalárska práca sa zameriava na rozpoznávanie nenávistnej reči za pomoci moderných veľkých jazykových modelov (LLM), ktoré sú schopné efektívne spracovávať a analyzovať texty v slovenčine.

Cieľom práce je zlepšiť existujúce metódy automatického rozpoznávania nenávistnej reči využitím pokročilých jazykových modelov a metód hodnotenia výkonu modelov. K tomu bude potrebné vykonať niekoľko kľúčových krokov:

1. Vypracovanie prehľadu veľkých jazykových modelov s podporou slovenčiny

Prvým krokom bude vykonať detailnú analýzu dostupných veľkých jazykových modelov, ktoré sú schopné spracovávať texty v slovenskom jazyku. Zameriame sa na populárne modely ako sú BERT [8], GPT [8], Mistral [3] a ďalšie, ktoré preukázali vysokú účinnosť pri spracovaní prirodzeného jazyka. Tieto modely sa často trénujú na obrovských množstvách textových dát a následne sú adaptované na špecifické úlohy, ako je klasifikácia textu, analýza sentimentu alebo, v našom prípade, detekcia nenávistnej reči. Prehľad sa bude týkať aj metodológií trénovania týchto modelov, ako aj ich výhod a obmedzení pri spracovaní slovenčiny. Výsledkom tejto analýzy bude výber najvhodnejšieho modelu na nasledujúci výskum.

1. Vypracovanie prehľadu dostupných textových korpusov pre rozpoznávanie nenávistnej reči

Ďalším krokom bude analýza a výber vhodných textových korpusov pre tréning a hodnotenie modelov. Rozpoznávanie nenávistnej reči je náročná úloha, ktorá si vyžaduje kvalitný a rozmanitý dataset obsahujúci rôzne formy nenávistných vyjadrení. Zameriame sa na dostupné korpusy, ktoré sú špecificky zamerané na identifikáciu nenávistnej reči v slovenčine, ako aj na korpusy, ktoré môžu byť prispôsobené pre tento účel. Pre tento účel bude potrebné analyzovať kvalitu týchto korpusov, ich veľkosť, pokrytie rôznych typov nenávistných prejavov a ich využiteľnosť pri tréningu modelov [11].

1. Výber modelu a metódy rozpoznávania nenávistnej reči

Na základe predchádzajúcich analýz budeme vyberať model a metódu, ktorá bude najefektívnejšia na rozpoznávanie nenávistnej reči. Pre tento účel budeme skúmať, ako rôzne modely, ako napríklad BERT [8], GPT [8] alebo Mistral [3], dokážu vykonávať klasifikáciu textu na základe predtrénovaných váh a následne sa adaptovať na úlohu detekcie nenávistnej reči. Bude potrebné rozhodnúť, či sa využije prístup, ktorý sa zameriava na jemné doladenie existujúceho modelu pomocou transfer learningu, alebo či bude potrebné trénovať model od nuly na vybraných dátach.

1. Vyhodnotenie presnosti rozpoznávania nenávistnej reči

Po implementácii vybraného modelu bude nevyhnutné vyhodnotiť jeho výkonnosť na testovacích dátach. Na tento účel použijeme niekoľko metód hodnotenia, ako sú presnosť (Precision), vyvolanie (Recall) a skóre F1 (F1-score), ktoré nám umožnia určiť, ako presne model dokáže identifikovať nenávistné vyjadrenia. Tieto metriky sú dôležité, pretože umožňujú vyvážené posúdenie výkonu modelu, berúc do úvahy nielen správnosť predpovedí, ale aj schopnosť modelu identifikovať všetky relevantné prípady nenávistnej reči [10].

1. Navrhnutie zlepšení vybranej metódy

Na základe výsledkov vyhodnotenia sa budú identifikovať slabé stránky zvoleného modelu a metodológie. V tejto fáze budú navrhnuté možné zlepšenia, ktoré by mohli viesť k zvýšeniu presnosti modelu. Môže ísť napríklad o optimalizáciu hyperparametrov, zlepšenie kvality trénovacích dát, rozšírenie datasetu o ďalšie príklady nenávistných vyjadrení, alebo využitie pokročilých techník, ako je metóda LoRA (Low-Rank Adaptation), ktorá môže zlepšiť schopnosť modelu adaptovať sa na nové úlohy s menším počtom trénovacích dát [9].

Celkovo cieľom tejto práce je prispieť k rozvoju efektívnych nástrojov na automatické rozpoznávanie nenávistnej reči, ktoré môžu byť použité na rôznych platformách a v rôznych aplikáciách na zlepšenie kvality online diskusií a prevenciu šírenia nenávistných prejavov. Výsledky tejto práce môžu mať praktický dopad na vývoj moderovacích nástrojov na sociálnych sieťach, forách a iných online platformách, kde sa problém nenávistnej reči stáva čoraz naliehavejší.

1. Teoretický rozbor zvolenej témy
   1. Hlboké učenie, jeho koncepty a využitie v rôznych priemyselných odvetviach
      1. Čo je to hlboké učenie?

Hlboké učenie (angl. deep learning) je špecifická podskupina strojového učenia (machine learning), ktorá sa zameriava na modelovanie a analýzu dát pomocou tzv. hĺbkových neurónových sietí (deep neural networks) [1].Tieto siete pozostávajú z viacerých vrstiev umelých neurónov, ktoré sa učia komplexné vzory a vzťahy v dátach. Práve počet vrstiev a ich schopnosť automaticky extrahovať čoraz abstraktnejšie reprezentácie dát je to, čo odlišuje hlboké učenie od tradičných metód strojového učenia [1].

Základným stavebným prvkom hlbokého učenia je neurónová sieť, ktorá je inšpirovaná fungovaním biologického mozgu. Neuróny v sieti prijímajú vstupy, aplikujú na ne váhy (weights), prechádzajú aktivačnými funkciami a produkujú výstupy, ktoré sa posúvajú do ďalšej vrstvy siete. Siete môžu mať desiatky až stovky vrstiev, pričom každá vrstva sa špecializuje na rozpoznávanie určitých aspektov vstupných dát – od jednoduchých čŕt (napríklad hrany v obrázkoch) až po komplexné koncepty (napríklad rozpoznanie tvárí alebo významu viet)[1].

Významným rysom hlbokého učenia je schopnosť automatického učenia reprezentácií bez potreby manuálneho návrhu vlastností (feature engineering), čo je často nutné pri klasických algoritmoch strojového učenia. Hlboké siete sa dokážu adaptovať na veľké množstvá nestruktúrovaných dát, ako sú obrázky, texty či zvuky, a dokážu z nich extrahovať relevantné informácie priamo počas tréningu[1].

Hlboké učenie zažilo obrovský rozmach v poslednom desaťročí, najmä vďaka rastúcemu výpočtovému výkonu (GPU a TPU akcelerátory), dostupnosti veľkých dátových súborov a inováciám v architektúre neurónových sietí. Modely ako AlexNet, VGG, ResNet či transformerové siete (napríklad BERT alebo GPT) priniesli prelomové výsledky v oblastiach ako počítačové videnie (computer vision), spracovanie prirodzeného jazyka (NLP) či rozpoznávanie reči [8].

Typické oblasti využitia hlbokého učenia zahŕňajú rozpoznávanie objektov na obrázkoch, automatický preklad textov, tvorbu odporúčacích systémov, predikciu správania systémov alebo analýzu medicínskych snímok. Vďaka tejto flexibilite sa hlboké učenie stalo jednou z kľúčových technológií modernej umelej inteligencie [1].

Napriek svojim výhodám však hlboké učenie nie je bez limitov. Tréning hlbokých modelov je extrémne náročný na výpočtové zdroje a vyžaduje veľké množstvo označených dát. Modely sú často vnímané ako „čierne skrinky“, pretože ich rozhodovací proces je ťažké interpretovať. Napriek týmto výzvam sa hlboké učenie naďalej vyvíja a posúva hranice možností v mnohých oblastiach [8].

* + 1. Využitie hlbokého učenia v rôznych priemyselných odvetviach

Hlboké učenie má široké využitie v rôznych priemyselných odvetviach, kde zefektívňuje procesy, optimalizuje výrobu, zlepšuje služby a umožňuje nové technologické pokroky [1]. V automobilovom priemysle sa hlboké učenie využíva predovšetkým v oblasti autonómnych vozidiel. Algoritmy spracovávajú obrazové a senzorové dáta na rozpoznávanie objektov, sledovanie chodcov a monitorovanie dopravného prostredia. Tieto technológie umožňujú samořiditeľným vozidlám bezpečne navigovať a reagovať na zmeny v okolí v reálnom čase [8].

V zdravotníctve je hlboké učenie kľúčovým nástrojom na analýzu medicínskych snímok, ako sú röntgeny, CT alebo MRI, kde modely dokážu identifikovať patologické zmeny, ako je rakovina, zápaly alebo iné ochorenia [1]. Taktiež sa využíva pri predikcii vývoja ochorení a výbere najvhodnejších liečebných postupov na základe historických a genetických dát pacientov [8].

V finančnom sektore umožňuje hlboké učenie detekciu podvodov na základe analýzy transakčných dát a správania klientov. Algoritmy môžu rýchlo identifikovať nezvyčajné vzory a upozorniť na podozrivé aktivity. Okrem toho sa používa aj pri hodnotení kreditného rizika a tvorbe prediktívnych modelov, ktoré pomáhajú pri rozhodovaní o investíciách a obchodných stratégiách [6].

V obchodnom sektore hlboké učenie zohráva dôležitú úlohu pri optimalizácii zásobovania a distribúcie produktov. Modely môžu predpovedať dopyt po výrobkoch na základe historických údajov a externých faktorov, ako sú počasie alebo sezónne trendy. Tento prístup umožňuje firmám efektívnejšie riadiť sklady a znižovať náklady spojené s nadbytočnými zásobami [7].

V poľnohospodárstve sa hlboké učenie využíva na monitorovanie a analýzu poľnohospodárskych plodín pomocou dronov a satelitných snímok. Modely dokážu detegovať choroby, škodcov a problémy s rastlinami ešte pred tým, než sa stanú viditeľné voľným okom, čím umožňujú včasnú intervenciu a optimalizáciu využívania pesticídov a hnojív [7].

V energetike hlboké učenie pomáha pri predikcii spotreby energie a optimalizácii distribúcie. Modely môžu analyzovať historické dáta o spotrebe a zohľadniť faktory ako počasie alebo sviatky, aby predpovedali dopyt a zabezpečili rovnováhu medzi výrobou a spotrebou. Taktiež sa využíva na predikciu porúch v infraštruktúre, čo pomáha pri údržbe a minimalizácii výpadkov [5].

Každé z týchto odvetví využíva špecifické aplikácie hlbokého učenia na zlepšenie svojich procesov a poskytovanie nových produktov a služieb, ktoré by inak neboli možné. Tieto technológie sú kľúčové pre transformáciu priemyselných sektorov a zvyšovanie ich konkurencieschopnosti [1].

* + 1. Základy hlbokého učenia v NLP

Spracovanie prirodzeného jazyka (NLP) je oblasť umelej inteligencie, ktorá sa zaoberá interakciou medzi počítačmi a ľudským jazykom. Hlboké učenie sa v posledných rokoch stalo kľúčovou technológiou v NLP, pretože umožňuje automatické učenie sa zo štruktúrovaných aj neštruktúrovaných jazykových dát a dosahuje vynikajúce výsledky v rôznych úlohách, ako je strojový preklad, analýza sentimentu, generovanie textu alebo rozpoznávanie entít [1][8].

Jedným z najvýznamnejších prístupov v hlbokom učení pre NLP je použitie neural networks (NN), konkrétne recurrent neural networks (RNN), long short-term memory (LSTM) sietí a novších modelov ako transformers. Tradičné prístupy spracúvali slová jednotlivými krokmi, ale tieto modely dokážu spracovávať celé sekvencie slov, pričom si pamätajú predchádzajúce informácie. Tento prístup zlepšuje porozumenie kontextu v dlhších textoch [8].

Recurrent Neural Networks (RNN) sú typom neurónových sietí, ktoré sú schopné spracovávať sekvenčné dáta tým, že informácie prenášajú medzi jednotlivými časovými krokmi. RNN sú veľmi užitočné pri úlohách, kde záleží na sekvenčnom kontexte, ako je analýza textov alebo generovanie textu. Avšak, klasické RNN modely čelili problému so zabúdáním dlhodobých závislostí. Tento problém bol vyriešený použitím LSTM a GRU (Gated Recurrent Units), ktoré lepšie uchovávajú informácie v dlhších sekvenciách [8].

Neskôr, transformerové modely ako BERT (Bidirectional Encoder Representations from Transformers) a GPT (Generative Pretrained Transformer) revolucionalizovali NLP. Tieto modely využívajú mechanizmus self-attention, ktorý umožňuje modelu zamerať sa na dôležité slová v celej sekvencii textu, čím sa výrazne zlepšuje pochopenie kontextu. Transformerové modely spracovávajú celé sekvencie textu naraz, na rozdiel od RNN, ktoré spracovávajú text krok za krokom. Tento prístup umožňuje efektívnejšie trénovanie na veľkých dátach a dosahovanie vysokej presnosti [8][9].

V NLP sa hlboké učenie využíva na rôzne úlohy. Jednou z nich je strojový preklad, kde modely ako Google Translate používajú rekurentné a transformerové modely na preklad textov medzi rôznymi jazykmi. Ďalšou dôležitou úlohou je rozpoznávanie entít (NER), kde modely identifikujú špecifické informácie v texte, ako sú mená, dátumy alebo lokality. Analýza sentimentu je ďalšia populárna úloha, pri ktorej modely určujú, či je text pozitívny, negatívny alebo neutrálny[9].

Významnou výhodou hlbokého učenia v NLP je aj schopnosť generovania textu. Modely ako GPT sú schopné vytvárať koherentné a kontextuálne relevantné texty, čo otvára možnosti pre automatizované generovanie článkov, odpovedí na otázky, alebo aj tvorbu kreatívneho obsahu [8][9].

Základy hlbokého učenia v NLP teda spočívajú v aplikácii rôznych typov neurónových sietí na úlohy, ktoré vyžadujú analýzu jazyka a jeho kontextu. S využitím moderných technológií, ako sú transformers, dosahuje NLP v súčasnosti vynikajúce výsledky v širokom spektre aplikácií [8][9].

* + 1. Transformerová architektúra a jej kľúčové vlastnosti

Transformerová architektúra je jedným z najvýznamnejších pokrokov v oblasti spracovania prirodzeného jazyka (NLP) a strojového učenia. Bola predstavená v práci Attention is All You Need od Vaswaniho a kol. v roku 2017 a od tej doby sa stala základom pre mnoho moderných modelov, ako sú BERT, GPT, T5, či BART. Transformer sa vyznačuje efektívnosťou a flexibilitou pri spracovaní sekvencií dát, čo umožňuje dosahovať vynikajúce výsledky pri rôznych úlohách, ako je strojový preklad, analýza sentimentu, rozpoznávanie entít, generovanie textu a mnoho ďalších [8][9] .

Kľúčovou vlastnosťou transformerovej architektúry je mechanizmus pozornosti (attention), konkrétne self-attention. Tento mechanizmus umožňuje modelu analyzovať každé slovo v kontexte celej sekvencie, nie len v súvislosti s predchádzajúcimi slovami. Na rozdiel od tradičných RNN a LSTM, ktoré spracovávajú sekvenciu slovo po slove, transformer umožňuje paralelný výpočet, čím zlepšuje efektivitu pri trénovaní na veľkých dátach [8][9].

Self-attention sa zameriava na vzťahy medzi všetkými slovami v sekvencii. Každé slovo v texte môže "venovať pozornosť" iným slovám v tej istej sekvencii, čím získava informácie o tom, ako sú jednotlivé slová navzájom závislé. Tento proces zahŕňa tri hlavné komponenty: query, key a value. Tieto reprezentácie sa používajú na výpočet váh, ktoré určujú, akú pozornosť si jednotlivé slová zaslúžia v rámci danej sekvencie [8].

Transformerová architektúra je postavená na dvoch hlavných častiach: enkodér a dekodér. Enkodér spracováva vstupnú sekvenciu, pričom každý enkodér obsahuje vrstvy self-attention a feedforward sietí, ktoré umožňujú modelu získať komplexné reprezentácie vstupného textu. Dekodér potom generuje výstupnú sekvenciu na základe týchto reprezentácií, pričom využíva mechanizmus pozornosti aj pri generovaní textu [8].

Jeden z hlavných dôvodov, prečo je transformer tak efektívny, je jeho schopnosť pracovať s paralelným výpočtom. Tradičné RNN a LSTM modely majú tendenciu spracovávať sekvenciu krok za krokom, čo znamená, že výpočty nie sú paralelizovateľné. Naproti tomu transformer spracováva celú sekvenciu súčasne, čo výrazne urýchľuje proces trénovania, najmä pri práci s veľkými datasetmi [8][9].

Okrem mechanizmu pozornosti, transformer tiež využíva pozicionálne enkódovanie, ktoré sa pridáva k vstupným dátam, aby model rozumel poradí slov v sekvencii. To je nevyhnutné, pretože transformer nepracuje s časovou postupnosťou, ako to robia RNN a LSTM, a preto musí vedieť, kde sa nachádza každé slovo v kontexte celej vety [8].

Jednou z ďalších výhod transformerovej architektúry je škálovateľnosť. Vzhľadom na to, že transformer umožňuje paralelné spracovanie sekvencií a jeho architektúra je modulárna, je možné prispôsobiť model rôznym veľkostiam, čo vedie k jeho efektívnosti pri trénovaní aj na veľmi veľkých datasetoch. Tento aspekt je kľúčový pri trénovaní modelov ako GPT-3 alebo BERT, ktoré obsahujú miliardy parametrov a sú schopné dosahovať špičkové výsledky v rôznych úlohách [8][9].

Všetky tieto vlastnosti spoločne robia transformer veľmi efektívnym nástrojom pre spracovanie textu a sú kľúčom k jeho úspechu v rôznych aplikáciách NLP [8][9] .

* + 1. Využitie LLM v NLP úlohách

Veľké jazykové modely (LLM, z anglického Large Language Models) sa stali revolučnými nástrojmi v oblasti spracovania prirodzeného jazyka (NLP). Tieto modely, ako sú GPT-3, BERT, T5 a BART, dosahujú vynikajúce výsledky v rôznych NLP úlohách, ako je strojový preklad, generovanie textu, analýza sentimentu, sumarizácia textu a rozpoznávanie entít [8][9].

LLM sa trénujú na obrovských množstvách textových dát, čo im umožňuje zachytiť rozsiahle jazykové vzory a komplexné vzťahy medzi slovami a frázami v rôznych kontextoch. Ich schopnosť chápať a generovať prirodzený jazyk je výsledkom pokročilých architektúr, ako sú transformery, ktoré umožňujú modelom spracovávať text efektívne, a to aj pri veľkých objemoch dát. Tento pokrok v oblasti modelov znamená, že LLM môžu byť prispôsobené rôznym úlohám v NLP [8][9].

Jednou z najpopulárnejších úloh, v ktorej LLM excelujú, je generovanie textu. Modely ako GPT-3 sú schopné generovať koherentné a kontextuálne relevantné texty na základe zadaných pokynov. Tento proces je známy ako text completion alebo prompt-based generation, kde model na základe začiatku textu predpovedá jeho pokračovanie. LLM sú tiež široko využívané na generovanie kreatívnych textov, ako sú články, poviedky, či marketingové materiály [9].

Ďalšou významnou úlohou je strojový preklad. Modely ako T5 a mnoho jazykové BERT varianty sa používajú na preklad textov medzi rôznymi jazykmi. Tieto modely sú schopné spracovať dlhé vety a zachovať význam aj pri preklade do iných jazykov, čo je výhodné v profesionálnych aplikáciách ako je automatizovaný preklad webových stránok alebo dokumentov [8].

LLM sú taktiež veľmi silné pri analýze sentimentu a klasifikácii textu. Tieto modely môžu identifikovať emocionálne tóny textov, ako je pozitívny, negatívny alebo neutrálny sentiment, čo je užitočné pri analýze zákazníckych recenzií, sociálnych médií alebo prieskumov. Na podobný spôsob môžu byť použité aj na identifikáciu tém, kde modely určujú hlavnú tému alebo kategóriu textu (napríklad šport, politika, zábava) [8].

Okrem toho, LLM excelujú v úlohách sumarizácie textu, kde sú schopné vytvoriť krátke zhrnutie dlhších dokumentov alebo článkov. Tento proces, nazývaný abstraktná sumarizácia, je ideálny pre aplikácie, ktoré vyžadujú rýchle pochopenie dlhých textov bez nutnosti ich čítania v plnom rozsahu, ako sú právne dokumenty, vedecké články alebo novinové správy [8].

LLM sú taktiež neoceniteľné v rozpoznávaní entít (NER), kde modely identifikujú a klasifikujú dôležité informácie v texte, ako sú mená, miesta, organizácie, dátumy a ďalšie. Tento proces je užitočný pri analýze veľkých textových súborov, ako sú správy alebo dokumenty, kde je potrebné extrahovať konkrétne informácie [8].

Využitie LLM v NLP úlohách je obrovské a stále sa rozširuje. Ich schopnosť pracovať s rôznymi jazykmi, rozumieť kontextu a generovať prirodzený text otvára nové možnosti pre aplikácie v oblasti umelých inteligencií, ako sú chatboti, virtuálni asistenti a automatické prekladanie, čím sa stávajú kľúčovými nástrojmi v moderných technológiách [8][9].

* + 1. Výzvy a limity veľkých jazykových modelov

Veľké jazykové modely (LLM), ako sú GPT-3, BERT a ďalšie pokročilé modely, predstavujú významný pokrok v oblasti spracovania prirodzeného jazyka. Napriek ich obrovským úspechom v rôznych úlohách, ako sú generovanie textu, preklad, sumarizácia a analýza sentimentu, tieto modely čelí niekoľkým výzvam a limitom, ktoré ovplyvňujú ich výkon, etické používanie a udržateľnosť [8][9].

Jednou z hlavných výziev je výpočtová náročnosť a energetická spotreba pri trénovaní veľkých jazykových modelov. Tréning modelov s miliardami parametrov vyžaduje obrovské množstvo výpočtového výkonu, čo vedie k vysokým nákladom na hardvér a energiu. Napríklad trénovanie modelov ako GPT-3 môže stáť milióny dolárov a využívať tisíce grafických procesorov (GPU) počas niekoľkých týždňov. Tento problém predstavuje významnú bariéru pre malé a stredné organizácie, ktoré si nemôžu dovoliť tieto náklady, čím dochádza k centralizácii vývoja týchto technológií [8][9].

Ďalšou výzvou je generalizácia modelov. Hoci LLM dosahujú vysoké výkony na rôznych úlohách, nie vždy sú schopné správne generalizovať na špecifické problémy alebo nezvyčajné vstupy. Napríklad môžu generovať nepresné alebo neadekvátne odpovede, keď sú vystavené neznámym alebo neštandardným vstupom. To je často spôsobené tým, že modely sa trénujú na širokom spektru dát, ale môžu mať problém so špecifickými jazykovými štruktúrami alebo doménami, ktoré neboli dostatočne zastúpené v trénovacích dátach [8][9].

Bias a etické otázky sú ďalším dôležitým aspektom pri používaní LLM. Tieto modely sa učia z obrovských korpusov textu, ktoré obsahujú ľudské predsudky a stereotypy. Ak modely neboli riadne filtrované alebo monitorované, môžu generovať obsah, ktorý je zaujatý, diskriminačný alebo nevhodný. Tento problém je obzvlášť závažný v prípadoch, keď modely generujú obsah, ktorý môže ovplyvniť verejné názory alebo byť použitý na manipuláciu s informáciami, ako je šírenie dezinformácií alebo hate speech [6][8].

Ďalšou prekážkou je rozpoznávanie kontextu a dlhodobá koherencia. Hoci LLM excelujú pri analýze krátkych textov, môžu mať problémy s udržaním koherencie v dlhších textoch. Pri generovaní textov alebo rozhovoroch môže model stratiť kontext alebo prepojenie medzi rôznymi časťami textu, čo vedie k nesúvislým alebo nesprávnym odpovediam. Tento problém je obzvlášť viditeľný pri interakciách, ktoré si vyžadujú udržanie dlhodobého kontextu, ako sú dialógy alebo komplexné analýzy textov [8].

Interpretačná schopnosť je ďalšou výzvou. LLM, aj keď produkujú výstupy, ktoré sa javia ako logické a súvislé, nie vždy sú schopné vysvetliť, ako dospeli k danému záveru. Tento problém je známy ako problém čiernej skrinky, kde modely sú schopné poskytovať výsledky bez jasného pochopenia, ako tieto výsledky vznikli. V aplikáciách, kde je potrebné dôkladne vysvetliť rozhodovanie modelu, ako je zdravotná diagnostika alebo právne poradenstvo, je táto nedostatok interpretácie vážnym problémom [9].

Na záver, etické a regulačné výzvy spojené s použitím LLM sú stále aktuálne. Existuje množstvo otázok týkajúcich sa ochrany súkromia, spravodlivosti a zodpovednosti pri používaní týchto technológií. S rastúcim využívaním LLM v rôznych oblastiach, ako sú automatizované systémy rozhodovania alebo tvorba obsahu, je nevyhnutné, aby sa vyvinuli etické rámce a regulačné mechanizmy, ktoré zabezpečia spravodlivé, transparentné a zodpovedné používanie týchto modelov [8][9].

* 1. Prehľad veľkých jazykových modelov pre slovenčinu (otvorené aj uzavreté)
     1. Kritériá hodnotenia jazykových modelov

Hodnotenie jazykových modelov je kľúčové na určenie ich kvality, efektivity a vhodnosti pre konkrétne úlohy v oblasti spracovania prirodzeného jazyka (NLP). Existuje viacero kritérií, ktoré sa používajú na hodnotenie modelov. Tieto kritériá zahŕňajú nielen technické aspekty modelu, ale aj jeho schopnosť prispôsobiť sa konkrétnym potrebám a požiadavkám úloh v rôznych jazykoch, vrátane slovenčiny [8][9][10].

Jedným z najdôležitejších kritérií je presnosť (accuracy), ktorá určuje, ako presne model vykonáva svoju úlohu, či už ide o klasifikáciu textu, preklad alebo generovanie odpovedí. Pre úlohy klasifikácie sa najčastejšie používa metrika F1 skóre, ktorá zohľadňuje rovnováhu medzi presnosťou a odvolaním. Presnosť meria počet správne predikovaných pozitívnych prípadov v porovnaní so všetkými predikovanými pozitívnymi výsledkami, zatiaľ čo odvolanie (recall) sleduje, koľko skutočne pozitívnych prípadov bolo správne identifikovaných [8][10].

Ďalšou dôležitou metrikou je koherencia textu, najmä pri generatívnych modeloch. Koherencia určuje, ako dobre model zachováva súvislosť a logickú štruktúru pri generovaní textu, čo je zásadné pri úlohách, ako je generovanie textov alebo automatizované rozhovory [8][9].

Rýchlosť a výpočtová náročnosť sú ďalšie faktory, ktoré ovplyvňujú hodnotenie modelov. Modely, ktoré majú vysokú výpočtovú náročnosť, môžu byť nepraktické pre reálne aplikácie, pretože vyžadujú silný výpočtový výkon a dlhšie časy spracovania. Optimalizované modely, ktoré dokážu udržiavať vyváženie medzi výkonmi a efektívnosťou, sú považované za lepšie [9][10].

Schopnosť generalizovať je ďalší dôležitý parameter. Modely, ktoré dokážu správne fungovať na nových a neznámych dátach, sú považované za robustné a efektívne. Schopnosť modelu adaptovať sa na rôzne jazyky, dialekty alebo domény môže byť rozhodujúca pri aplikáciách, ako je strojový preklad alebo analýza sentimentu [8][9].

Nakoniec, interpretovateľnosť modelov, ktorá označuje schopnosť vysvetliť, prečo model prišiel k určitému rozhodnutiu, je čoraz dôležitejšia, najmä v etických a regulačných kontextoch. Transparentnosť v rozhodovacom procese je dôležitá na zabezpečenie dôvery používateľov a etického používania modelov [10].

* + 1. Otvorené modely trénované na slovenčine

Otvorené jazykové modely, ktoré sú špeciálne trénované na slovenčine, zohrávajú dôležitú úlohu v oblasti spracovania prirodzeného jazyka (NLP) pre tento jazyk. Tieto modely sú k dispozícii pre komunitu a môžu byť voľne použité na rôzne úlohy, ako je analýza sentimentu, preklad, extrakcia informácií, klasifikácia textu a ďalšie aplikácie. Otvorenosť týchto modelov umožňuje výskumníkom a vývojárom vylepšovať ich a prispôsobiť špecifickým potrebám v slovenskom jazyku [11][12].

Jeden z najznámejších otvorených modelov pre slovenčinu je Slovenský BERT (SlovakBERT). Tento model je variantom populárneho BERT (Bidirectional Encoder Representations from Transformers), ktorý bol trénovaný na veľkom množstve slovenských textov. Slovenský BERT bol trénovaný na textoch z rôznych domén, ako sú noviny, blogy a internetové fóra, čím sa zabezpečilo, že model rozumie rôznym jazykovým nuansám a kontextom v slovenčine. Tento model je efektívny pri úlohách, ako je analýza sentimentu, extrakcia entít a textová klasifikácia [12].

Ďalším príkladom otvoreného modelu je Slovenský RoBERTa, ktorý vychádza z BERT architektúry, ale je vylepšený a optimalizovaný pre rýchlejší tréning a lepší výkon na rôznych NLP úlohách. Tento model je tiež široko používaný v slovenskom jazyku a je k dispozícii na použitie a ďalší výskum [12].

Okrem BERT a RoBERTa existujú aj ďalšie modely, ako slovenský T5 a XLM-RoBERTa, ktoré sa zameriavajú na špecifické úlohy, ako je strojový preklad alebo viacjazyčné porozumenie. Modely ako T5, ktoré sú založené na transformerovej architektúre, môžu byť prispôsobené na rôzne úlohy pomocou prístupov ako „text-to-text“, čo umožňuje flexibilitu pri riešení úloh ako sumarizácia alebo generovanie textu [12].

Hoci otvorené modely pre slovenčinu stále zaostávajú za tými, ktoré sú dostupné pre väčšie jazyky, ako je angličtina, vývoj v tejto oblasti pokračuje a nové modely sa stále objavujú. Tieto modely sú cenným nástrojom pre vývojárov a výskumníkov, ktorí sa snažia zlepšiť kvalitu a efektivitu NLP aplikácií pre slovenčinu. Otvorenosť týchto modelov podporuje aj komunitu vývojárov, ktorí môžu zdieľať vylepšenia a prispievať k ich ďalšiemu rozvoju [12].

* + 1. Multilingválne otvorené modely s podporou slovenčiny

Multilingválne modely predstavujú významný krok vpred v oblasti spracovania prirodzeného jazyka, najmä pre jazyky, ktoré nie sú vždy široko podporované. Tieto modely sú navrhnuté tak, aby podporovali viacero jazykov, čím poskytujú flexibilitu pri práci s rôznymi jazykmi v rámci jednej architektúry. Pre slovenčinu to znamená, že aj napriek jej menšej popularite v NLP komunite sú k dispozícii pokročilé nástroje, ktoré umožňujú pracovať s týmto jazykom v širšom kontexte [12].

Jedným z najvýznamnejších multilingválnych modelov je XLM-RoBERTa, ktorý je variantom RoBERTa a bol trénovaný na množstve rôznych jazykov, vrátane slovenčiny. Tento model sa stal populárnym v oblasti NLP úloh, ako je textová klasifikácia, analýza sentimentu a extrakcia entít, vďaka svojej schopnosti spracovávať viacero jazykov naraz. XLM-RoBERTa je schopný poskytovať výkonné výsledky aj pre slovenčinu, aj keď sa na tomto jazyku netrénoval tak intenzívne, ako na väčších jazykoch, ako je angličtina. Tento model je veľmi užitočný v prípadoch, keď je potrebné pracovať s viacjazyčnými dátami alebo keď nie je k dispozícii špecifický jazykový model pre daný jazyk [12].

Ďalším významným modelom v tejto oblasti je mBERT (Multilingual BERT), ktorý je trénovaný na 104 jazykoch vrátane slovenčiny. Podobne ako XLM-RoBERTa, mBERT umožňuje riešiť rôzne NLP úlohy v slovenčine aj iných jazykoch. Je veľmi efektívny pri úlohách, ktoré si vyžadujú porozumenie textu v rôznych jazykoch, ako je strojový preklad, klasifikácia textu, alebo generovanie textu [12].

Výhodou multilingválnych modelov je ich univerzálnosť. Namiesto vytvárania špecifických modelov pre každý jazyk môžu výskumníci a vývojári použiť jeden model, ktorý pokrýva viacero jazykov, čo zjednodušuje prácu pri projektoch, ktoré zahŕňajú rôzne jazyky alebo vyžadujú adaptáciu na rôzne jazykové a kultúrne kontexty. Pre slovenčinu sú tieto modely dôležité najmä v oblastiach ako strojový preklad, analýza sentimentu, otázky a odpovede, alebo generovanie textu, kde sa často stretávame s potrebu podporovať viac jazykov súčasne [12].

Avšak, aj keď multilingválne modely prinášajú značné výhody, majú aj svoje obmedzenia. Na rozdiel od špecifických modelov trénovaných na konkrétny jazyk, môžu multilingválne modely vykazovať nižší výkon v prípade jazykov, ktoré sú menej zastúpené v tréningových dátach, ako je tomu v prípade slovenčiny. Napriek tomu, tieto modely stále ponúkajú dobrú rovnováhu medzi univerzálnosťou a výkonnosťou a sú veľmi cenné pri práci so slovenčinou v rôznych NLP aplikáciách [12].

* + 1. Uzavreté veľké jazykové modely a ich podpora slovenčiny

Uzavreté veľké jazykové modely sú komerčné nástroje, ktoré sú vyvinuté a spravované rôznymi technologickými spoločnosťami. Na rozdiel od otvorených modelov, ktoré sú k dispozícii na voľné použitie a prispôsobenie, uzavreté modely sú zvyčajne prístupné prostredníctvom platených API alebo predplatných. Tieto modely sú často trénované na obrovských množstvách dát a majú vysoký výkonnostný potenciál, no používateľom obmedzujú prístup k samotnej architektúre modelu alebo jeho tréningovým dátam [8][9][12].

Niektoré z najznámejších uzavretých veľkých jazykových modelov, ktoré podporujú slovenčinu, zahŕňajú GPT-3 (Generative Pre-trained Transformer 3) od OpenAI, BERT od Google, a Claude od Anthropic. Tieto modely sa často používajú na rôzne aplikácie NLP, ako je generovanie textu, sumarizácia, strojový preklad, a otázky a odpovede. Napriek tomu, že sú trénované predovšetkým na anglických textoch, majú schopnosť spracovávať aj ďalšie jazyky, vrátane slovenčiny [8][9] .

GPT-3 je jedným z najvyspelejších uzavretých modelov, ktorý poskytuje veľmi silné výsledky v oblasti generovania textu a porozumenia. Tento model je schopný pracovať so slovenčinou, aj keď jeho primárna sila spočíva v anglických textoch. GPT-3 je veľmi efektívny pri generovaní textu, ale môže mať obmedzenú presnosť v prípade jazykov, ktoré nie sú hlavne zastúpené v jeho trénovacích dátach, ako je slovenčina. Napriek tomu, jeho flexibilita a schopnosť vykonávať rôzne úlohy ho robí populárnym v mnohých aplikáciách [9].

Claude, model od spoločnosti Anthropic, sa zameriava na zodpovedanie otázok, generovanie textu a ďalšie úlohy, ktoré sú podobné tým, ktoré vykonáva GPT-3. Hoci je tento model primárne navrhnutý na spracovanie angličtiny, jeho schopnosť pracovať s viacerými jazykmi, vrátane slovenčiny, umožňuje jeho použitie v rôznych kontextoch. Claude tiež zdieľa obmedzenia podobné tým, ktoré má GPT-3 pri spracovaní menej bežných jazykov, ako je slovenčina [9].

Tieto uzavreté modely sú často vysoce optimalizované a poskytujú vysokú presnosť a výkon pri spracovaní textov v rôznych jazykoch, vrátane slovenčiny [8][9]. Avšak, ich použitie môže byť nákladné, najmä v prípade väčších aplikácií, ktoré vyžadujú vysoký objem výpočtových zdrojov. Okrem toho je možné, že modely nebudú úplne prispôsobené na špecifické jazykové vlastnosti a nuansy slovenčiny, čo môže mať vplyv na ich výkon pri riešení určitých úloh [9].

Jedným z významných benefitov uzavretých modelov je ich robustnosť a rýchlosť. Tieto modely sú podporované veľkými technologickými spoločnosťami, ktoré zaručujú neustálu aktualizáciu a optimalizáciu. V porovnaní s otvorenými modelmi sú uzavreté modely často lepšie prispôsobené komerčným aplikáciám, kde je potrebná stabilita, spoľahlivosť a škálovateľnosť [9].

* + 1. Porovnanie modelov a ich vhodnosť pre NLP úlohy v slovenčine

Porovnanie rôznych jazykových modelov a ich vhodnosť pre úlohy spracovania prirodzeného jazyka (NLP) v slovenčine je kľúčové pri výbere najlepšieho modelu pre konkrétny projekt alebo aplikáciu. Slovenský jazyk, ako menší jazyk v globálnom meradle, čelí určitým výzvam pri práci s NLP modelmi. Modely, ktoré boli trénované prevažne na angličtine, môžu vykazovať obmedzenú presnosť pri spracovávaní slovenčiny. Preto je dôležité zvážiť viacero faktorov, ako je kvalita modelu, jeho jazyková podpora, dostupnosť pre konkrétne úlohy a technické požiadavky [8][9][12] .

Otvorené modely, ako napríklad XLM-RoBERTa a mBERT, sú výbornou voľbou pre aplikácie, ktoré vyžadujú podporu viacerých jazykov vrátane slovenčiny. Tieto modely sú trénované na rozsiahlych viacjazyčných dátach, ktoré zahrnujú aj slovenský jazyk, a preto sú schopné vykonávať úlohy ako analýza sentimentu, klasifikácia textu, extrakcia entít a strojový preklad. Aj keď môžu mať nižší výkon v porovnaní s modelmi špecifickými pre slovenčinu, ich univerzálnosť a široká jazyková podpora ich robí ideálnymi pre aplikácie, ktoré zahŕňajú viacero jazykov alebo pre projekty s obmedzenými dátami pre konkrétny jazyk [12].

Na druhej strane, uzavreté modely ako GPT-3 a Claude poskytujú vynikajúce výsledky pri generovaní textu a vykonávaní komplexných úloh v rôznych jazykoch. Avšak, ich výkonnosť v slovenčine môže byť ovplyvnená tým, že tieto modely sú primárne trénované na angličtine, čo môže viesť k nižšej presnosti v prípade špecifických úloh v slovenčine. Uzavreté modely sú však veľmi silné v generovaní textu, otázkach a odpovediach, a ich výkon je zvyčajne stabilný a robustný. Tieto modely sú vhodné najmä pre aplikácie, ktoré vyžadujú vysokú kvalitu generovania textu, ako je automatické generovanie článkov, tvorba obsahu a komunikácia so zákazníkmi [9].

Pri porovnávaní modelov je tiež potrebné zohľadniť technické požiadavky a dostupnosť zdrojov. Otvorené modely môžu byť nasadené a prispôsobené na konkrétne úlohy s menšími nákladmi, avšak môžu vyžadovať väčšie výpočtové kapacity pri spracovávaní veľkého množstva dát. Uzavreté modely často poskytujú prístup prostredníctvom API, čo umožňuje ich jednoduchšiu integráciu do aplikácií, ale tento prístup môže byť nákladný a závislý na externých službách [9].

Záverom, výber modelu závisí od konkrétnych potrieb projektu. Ak je potrebná flexibilita a podpora viacjazyčných dát, otvorte modely ako XLM-RoBERTa a mBERT sú výborné voľby. Naopak, pre úlohy generovania textu alebo riešenie komplexných otázok s vysokou presnosťou môžu uzavreté modely ako GPT-3 a Claude poskytnúť lepšie výsledky, aj keď s obmedzeniami pri spracovávaní slovenčiny [9].

* 1. Klasifikácia textu – NLP úlohy
     1. Základy textovej klasifikácie v NLP

Textová klasifikácia je jednou z najzákladnejších a najdôležitejších úloh v oblasti spracovania prirodzeného jazyka (NLP). Je to proces priradenia kategórie alebo štítku k textovým dátam na základe ich obsahu. Tento proces môže byť aplikovaný na rôzne úlohy, ako je sentimentálna analýza, spamová detekcia, kategorizácia správ, rozpoznávanie tém a ďalšie. Textová klasifikácia je fundamentálnou úlohou pre množstvo aplikácií v reálnom svete, od analýzy zákazníckych recenzií až po automatické filtrovanie e-mailov [8][9][12].

Textová klasifikácia zvyčajne zahŕňa niekoľko krokov: predspracovanie textu, extrakciu vlastností a následné použitie klasifikačných algoritmov na predikciu kategórie. Predspracovanie textu môže zahŕňať kroky ako tokenizáciu, lematizáciu a odstránenie zastaraných slov (stop words). Po predspracovaní sa textové dáta zvyčajne transformujú do číselných vektorov pomocou techník ako bag-of-words, TF-IDF alebo embeddingov, ako sú Word2Vec alebo GloVe [8][12].

Klasifikačné algoritmy, ktoré sa často používajú v NLP úlohách, zahŕňajú metódy strojového učenia ako logistickú regresiu, podporu vektorových strojov (SVM), rozhodovacie stromy, a novšie prístupy ako neurónové siete, vrátane rekurentných neurónových sietí (RNN) a konvolučných neurónových sietí (CNN). S príchodom moderných modelov, ako je Transformer, sa pre textovú klasifikáciu čoraz viac používajú modely ako BERT alebo GPT, ktoré dokážu efektívne spracovať veľké množstvo textových dát a poskytnúť vysoko kvalitné výsledky [8][9].

V textovej klasifikácii je dôležité určiť správne metriky hodnotenia modelu. Najbežnejšie používané metriky sú presnosť (accuracy), precíznosť, recall a F1-skóre, ktoré hodnotia, ako dobre model identifikuje jednotlivé kategórie. Pre úlohy s nevyváženými dátami sa často používa aj AUC (Area Under Curve) alebo ROC krivky [10].

* + 1. Prompting ako prístup k textovej klasifikácii

Prompting sa stal dôležitým prístupom v oblasti textovej klasifikácie s príchodom veľkých jazykových modelov (LLM), ako sú GPT a BERT. Tento prístup spočíva v tom, že sa model požiadavá na konkrétnu úlohu alebo úlohy pomocou štruktúrovaných textových vstupov, ktoré modelu poskytujú informácie o tom, čo má robiť. Prompting môže byť efektívny spôsob, ako získať modely na riešenie úloh bez potreby trénovania od začiatku. Existujú rôzne formy promptingových prístupov, z ktorých najčastejšie používané sú zero-shot a few-shot učenie [8][9] .

* + - 1. Zero-shot learning

Zero-shot learning je prístup, pri ktorom model vykonáva úlohu bez toho, aby bol špecificky trénovaný na konkrétnych príkladoch úlohy. Model dostáva prompt, ktorý obsahuje dostatok informácií na to, aby sa mohol rozhodnúť alebo vykonať požiadavku aj bez priameho trénovania na tejto úlohe. Tento prístup sa stal veľmi populárny v kontexte veľkých jazykových modelov ako GPT, ktoré sú trénované na obrovskom množstve textových dát z rôznych domén, čím im umožňujú vykonávať širokú škálu úloh aj bez explicitného učenia sa týchto úloh [8][9.

V textovej klasifikácii sa zero-shot prístup využíva napríklad pri zadávaní promptov, ktoré modelu hovoria, že sa má rozhodnúť, do akej kategórie patrí určitý text, aj keď model nebol špecificky trénovaný na danú kategóriu. Tento prístup je obzvlášť užitočný, keď nie sú dostupné špecifické trénovacie dáta pre konkrétnu úlohu [8].

* + - 1. Few-shot learning

Few-shot learning je podobný zero-shot learningu, ale líši sa tým, že model dostane niekoľko príkladov úlohy, ktoré mu pomôžu lepšie pochopiť, čo sa od neho očakáva. Tento prístup zlepšuje schopnosť modelu prispôsobiť sa novým úlohám na základe minimálneho množstva trénovacích dát. Few-shot learning sa často používa pri aplikáciách, kde je ťažké alebo nákladné získať veľké množstvo trénovacích dát [9].

Pri few-shot learningu model dostáva niekoľko príkladov kategórií a následne je požiadaný o klasifikáciu nového textu na základe týchto príkladov. Tento prístup výrazne zlepšuje výkonnosť modelov v prípadoch, keď nie je k dispozícii dostatok dát na tradičné trénovanie [9].

* + - 1. Výhody a nevýhody promptingového prístupu

Výhody promptingového prístupu zahŕňajú flexibilitu a úsporu času. Modely ako GPT-3, ktoré sú vycvičené na veľkých množstvách dát, sú schopné vykonávať rôzne úlohy na základe správne konštruovaných promptov, čo môže byť veľmi efektívne, ak nie je potrebné trénovať modely od začiatku. Prompting je tiež veľmi užitočný v situáciách, keď sú potrebné rýchle odpovede na rôzne úlohy bez potreby rozsiahlej prípravy [8].

Nevýhodou je, že prompting môže byť menej presný v porovnaní s prístupmi, ktoré zahŕňajú explicitné trénovanie modelu. Modely môžu mať obmedzené pochopenie nuáns v špecifických úlohách, čo môže ovplyvniť ich výkon. Okrem toho, promptovanie si vyžaduje expertné spracovanie a testovanie rôznych formulácií promptov, čo môže byť časovo náročné [9].

* + 1. Adaptívne trénovanie modelov pre textovú klasifikáciu

Adaptívne trénovanie modelov je prístup, ktorý umožňuje modelom prispôsobiť sa novým úlohám alebo doménam s minimálnymi požiadavkami na tréningové dáta. Tento prístup je obzvlášť užitočný pri práci s veľkými jazykovými modelmi, ktoré boli predtrénované na obrovských množstvách dát. Adaptívne trénovanie môže byť vykonávané rôznymi spôsobmi, pričom medzi najbežnejšie používané metódy patrí fine-tuning, LoRA a PEFT [9].

* + - 1. Fine-tuning

Fine-tuning je proces, pri ktorom sa model, ktorý bol predtrénovaný na obrovských množstvách dát, prispôsobí konkrétnej úlohe pomocou menšieho množstva špecifických trénovacích dát. Tento proces zvyčajne zahŕňa pokračovanie tréningu modelu na špecializovaných dátach, aby sa zlepšila jeho výkonnosť pri konkrétnej úlohe. Fine-tuning je veľmi efektívny, pretože umožňuje modelu využívať znalosti získané počas predtrénovania a prispôsobiť sa novým požiadavkám [9].

* + - 1. LoRA (Low-Rank Adaptation)

LoRA je technika adaptívneho trénovania, ktorá sa zameriava na zníženie počtu parametrov, ktoré je potrebné aktualizovať počas fine-tuningu. Tento prístup umožňuje efektívnejšie a rýchlejšie prispôsobenie modelov, pretože sa zameriava len na malú časť parametrov, ktoré sú potrebné na prispôsobenie modelu novej úlohe. LoRA je zvlášť užitočná, keď je potrebné trénovať modely s obmedzenými výpočtovými prostriedkami [9].

* + - 1. PEFT (Parameter-Efficient Fine-Tuning)

PEFT je ďalšou technikou, ktorá sa zameriava na efektívnosť trénovania modelov. Tento prístup sa vyhýba úplnému pretrénovaniu všetkých parametrov modelu a namiesto toho sa zameriava na prispôsobenie len tých parametrov, ktoré sú na danú úlohu najviac relevantné. PEFT je veľmi užitočný pri práci s veľkými modelmi, ktoré majú obrovské množstvo parametrov, čím sa znižuje potreba rozsiahlej výpočtovej sily [9].

* + 1. Porovnanie jednotlivých prístupov v klasifikácii textu

Pri porovnávaní rôznych prístupov v textovej klasifikácii je dôležité zvážiť niekoľko faktorov, ako sú presnosť, rýchlosť tréningu, náklady na dáta a flexibilita modelu. Zero-shot a few-shot učenie ponúkajú rýchle a flexibilné riešenia, ale môžu byť menej presné v porovnaní s modelmi, ktoré prešli adaptívnym trénovaním ako fine-tuning alebo LoRA. Tieto metódy môžu poskytnúť lepšiu presnosť, ale vyžadujú viac tréningových dát a výpočtových zdrojov. V konečnom dôsledku výber prístupu závisí od konkrétneho prípadu použitia, dostupnosti dát a výpočtových zdrojov [9] .

* 1. Problémy a možné budúce zlepšenia
     1. Hlavné výzvy pri detekcii nenávistnej reči

Detekcia nenávistnej reči je jednou z najzložitejších úloh v oblasti spracovania prirodzeného jazyka (NLP), ktorá sa snaží identifikovať a klasifikovať texty obsahujúce nenávistné, urážlivé alebo diskriminačné vyjadrenia [5][6]. Hlavné výzvy pri tejto úlohe zahŕňajú:

* Ambiguity a kontextová závislosť: Nenávistná reč môže byť veľmi závislá od kontextu. To, čo môže byť považované za nenávistné v jednom kontexte, môže byť úplne neškodné v inom. Napríklad urážky, ktoré sú súčasťou vtipu, nemusia byť vždy nebezpečné, ale môžu byť v niektorých situáciách vnímané ako nenávistné [6].
* Jazyková a kultúrna variabilita: Klasifikácia nenávistnej reči môže byť ovplyvnená jazykovými a kultúrnymi rozdielmi. Výrazy považované za urážlivé alebo nenávistné v jednej kultúre nemusia byť rovnaké v inej. Tento problém sa ešte zhoršuje pri multilingválnych systémoch, kde môže byť ťažké správne identifikovať urážky v rôznych jazykoch [6][7].
* Nezvyčajné a jemné prejavy nenávistnej reči: Nenávistná reč nemusí byť vždy explicitná. Môže byť vyjadrená formou kódovaných alebo skrytých správ, čo môže byť náročné na detekciu pre tradičné modely spracovania textu. Tento problém môže zahŕňať aj ironizovanie, sarkazmus alebo prejavy nenávisti zamaskované pod povrchom iných argumentov [6].
* Zneužívanie systému: Ďalšou výzvou je, že niektorí používatelia môžu vedome manipulovať so systémom, aby sa vyhli detekcii nenávistnej reči. Tento problém môže zahŕňať použitie synonym, skrytých odkazov alebo špeciálnych znakov, aby sa obchádzali bezpečnostné mechanizmy [5][6].

Pre úspešné riešenie týchto výziev je potrebné zlepšiť existujúce metódy a vyvinúť nové technológie na spracovanie kontextuálnych, jemných a multilingválnych aspektov nenávistnej reči [6][7].

* + 1. Možnosti zlepšenia jazykových modelov pri klasifikácii nenávistnej reči

Jazykové modely v súčasnosti dosahujú veľmi dobré výsledky pri klasifikácii nenávistnej reči, avšak stále existujú oblasti, kde môžu byť zlepšené [6][7]. Medzi možnosti zlepšenia patrí:

* Lepšie modelovanie kontextu: Pre detekciu nenávistnej reči je kľúčové modelovať širší kontext textu. Súčasné modely, ako BERT a GPT, sú silné v porozumení krátkemu kontextu, ale pri dlhších diskusiách alebo komplexných interakciách medzi viacerými používateľmi môžu mať problémy s presným pochopením zámery textu. Integrácia modelov, ktoré lepšie chápu kontext, môže výrazne zlepšiť výsledky [6][7].
* Multimodálne prístupy: Detekcia nenávistnej reči sa často neobmedzuje len na text. V kombinácii s ďalšími údajmi, ako sú obrázky alebo zvuky (napr. v komentároch na videách), môžu multimodálne prístupy poskytovať bohatší pohľad na to, čo sa deje v komunikácii. Integrácia multimodálnych dát do modelov môže viesť k lepšiemu pochopeniu prejavov nenávistnej reči [6][7].
* Lepšie generalizovanie a zlepšenie generalizačných schopností modelov: Dôležité je, aby modely dokázali generalizovať aj mimo doménu, v ktorej boli trénované. Aktuálne modely môžu mať problémy s aplikovaním svojej učenia na nové dáta, čo môže viesť k zníženiu ich výkonnosti pri neznámych prejavoch nenávistnej reči. Zlepšenie generalizácie je kľúčové pre širokú aplikovateľnosť modelov 6][7].
* Pokročilé techniky proti manipulácii so systémom: Zlepšenie robustnosti modelov voči manipulácii a obchádzaniu detekcie je ďalšou dôležitou oblasťou. Tieto techniky môžu zahŕňať pokročilé metódy detekcie skrytých prejavov nenávistnej reči, využívanie ďalších detekčných vrstiev alebo kombinovanie viacerých techník spracovania textu [5][6].

Pokrok v týchto oblastiach môže pomôcť vytvoriť robustnejšie a presnejšie systémy pre klasifikáciu nenávistnej reči v rôznych jazykoch a kultúrach.

* + 1. Adaptívne trénovanie a nové techniky modelovania

Adaptívne trénovanie sa stalo kľúčovým prístupom v oblasti NLP, najmä pri trénovaní modelov pre špecifické úlohy, ako je detekcia nenávistnej reči [8]. Niektoré z nových techník, ktoré môžu výrazne zlepšiť výkonnosť modelov, zahŕňajú:

* Fine-tuning: Fine-tuning je proces, pri ktorom sa model predtrénovaný na veľkom množstve dát prispôsobí konkrétnej úlohe, ako je detekcia nenávistnej reči. Tento proces môže výrazne zlepšiť výkon modelov, keď sa tréningová úloha špecializuje na daný problém [8].
* LoRA (Low-Rank Adaptation): LoRA je technika, ktorá umožňuje znížiť počet parametrov, ktoré musia byť upravené počas adaptívneho trénovania, čo vedie k rýchlejšiemu a efektívnejšiemu trénovaniu. Táto metóda môže byť užitočná pri práci s veľkými jazykovými modelmi, kde sa často vyžaduje rozsiahla výpočtová kapacita[9] .
* PEFT (Parameter-Efficient Fine-Tuning): PEFT sa zameriava na to, aby model prispôsobil len tie parametre, ktoré sú kľúčové pre konkrétnu úlohu, čím sa znižuje potreba rozsiahleho trénovania. Tento prístup môže byť obzvlášť užitočný v prípadoch, keď sú trénovacie dáta limitované [9].
* Multitask learning: Multitask learning je prístup, kde sa model trénuje na viacerých úlohách naraz, čo mu umožňuje zdieľať vedomosti medzi úlohami. Tento prístup môže byť efektívny pri detekcii nenávistnej reči, pretože modely môžu zdieľať vedomosti medzi rôznymi formami prejavov nenávisti [8][9].

Tieto techniky môžu pomôcť zlepšiť výkonnosť modelov pri detekcii nenávistnej reči a zároveň umožniť rýchlejší a efektívnejší tréning [6][9].

* + 1. Budúce smery výskumu v oblasti NLP a hate speech detection

Výskum v oblasti NLP a detekcie nenávistnej reči pokračuje rýchlym tempom. Medzi kľúčové budúce smery výskumu patria:

Vylepšenie multimodálnych prístupov: S rastúcim množstvom multimediálnych dát, ako sú obrázky, videá a zvukové nahrávky, bude dôležité vyvinúť techniky, ktoré dokážu integrovať tieto rôzne formy dát na efektívnu detekciu nenávistnej reči. Multimodálne prístupy môžu prispieť k lepšiemu porozumeniu kontextu a jemným nuansám prejavov nenávistnej reči [6].

Zlepšenie modelov pre multilingválnu detekciu: Detekcia nenávistnej reči v rôznych jazykoch je stále výzvou, najmä keď sa zohľadňuje jazyková a kultúrna rôznorodosť. Výskum zameraný na multilingválne modely, ktoré dokážu správne identifikovať nenávistnú reč v rôznych jazykoch, bude kľúčový pre globálne aplikácie [7].

Pokročilé techniky pre pochopenie kontextu: Detekcia nenávistnej reči vyžaduje pochopenie širšieho kontextu, ako sú sociálne a politické faktory, ktoré môžu ovplyvniť vnímanie textov. Pokročilé modely, ktoré budú schopné lepšie chápať a interpretovať tento kontext, môžu významne zlepšiť presnosť detekcie [6][7].

Etické otázky a bias v modeloch: Výskum bude musieť zohľadniť aj etické otázky spojené s detekciou nenávistnej reči, ako sú problémy s biasom v trénovacích dátach a v modeloch. Zabezpečenie, aby modely boli spravodlivé a nezaujato klasifikovali texty, je kľúčové pre ich širokú akceptáciu a aplikáciu [6][7].

Pokrok v týchto oblastiach prispeje k vytvoreniu robustných a spravodlivých systémov na detekciu nenávistnej reči [6][7].

1. Analýza stavu problematiky
   1. Existujúce modely vhodné pre slovenčinu

V tejto časti sa zameriame na existujúce jazykové modely, ktoré sú vhodné pre spracovanie slovenčiny a môžu byť použité na úlohy, ako je detekcia nenávistnej reči. Tieto modely sú postavené na rôznych architektúrach a metodikách a sú schopné efektívne riešiť rôzne úlohy spracovania prirodzeného jazyka (NLP) [8][9][12].

* + 1. SlovakBERT

SlovakBERT je jazykový model založený na populárnej architektúre BERT (Bidirectional Encoder Representations from Transformers). Tento model bol prispôsobený pre slovenčinu, čo znamená, že bol trénovaný na slovenskej textovej dátovej báze a je schopný rozpoznať jazykové štruktúry špecifické pre tento jazyk. BERT je model navrhnutý na predtrénovanie v dvoch krokoch: najskôr sa model učí z neoznačeného textu pomocou úlohy maskovaného jazykového modelu (Masked Language Modeling), a následne sa doladí na špecifické úlohy pomocou anotovaných dát (fine-tuning) [12].

SlovakBERT sa osvedčil v rôznych úlohách spracovania textu, ako je klasifikácia textu, rozpoznávanie pomenovaných entít (Named Entity Recognition, NER), analýza sentimentu a extrakcia informácií. Pri detekcii nenávistnej reči je jeho silnou stránkou schopnosť rozpoznať jemné nuansy v texte, ako je identifikácia urážlivých alebo diskriminačných výrazov, ktoré môžu byť nejednoznačné alebo obsahovať skrytú nenávistnú reč. Tento model je teda veľmi účinný na úlohy, ktoré vyžadujú hlboké pochopenie syntaxe a sémantiky textu [12].

* + 1. SlovakT5

SlovakT5 je model, ktorý je založený na architektúre T5 (Text-to-Text Transfer Transformer). Tento model prispôsobený pre slovenčinu využíva prístup "text-to-text", čo znamená, že všetky úlohy sú formulované ako prevod textu na text. Model teda dostane ako vstup text vo forme zadania a vygeneruje výstup vo forme odpovede alebo iného textového výstupu [8].

T5 je veľmi flexibilný model, ktorý umožňuje vykonávanie rôznych úloh spracovania textu vrátane prekladu, sumarizovania, generovania textu a klasifikácie. Pre úlohu detekcie nenávistnej reči je tento model veľmi efektívny, pretože dokáže z textu identifikovať nenávistné alebo urážlivé výrazy a transformovať tento text do požiadavky alebo výstupu, ktorý označuje jeho povahu. SlovakT5 je preto ideálnym kandidátom na úlohy, kde je potrebné nielen klasifikovať text, ale aj generovať ďalší relevantný obsah na základe analýzy textu [8].

* + 1. Multilinguálne modely (mT5, XLM-R)

Okrem špecializovaných modelov pre slovenčinu existujú aj multilinguálne modely, ktoré boli trénované na širokej škále jazykov a sú schopné spracovať text v rôznych jazykoch vrátane slovenčiny. Tieto modely môžu byť výhodné v prípade, že potrebujeme model schopný spracovať texty aj v iných jazykoch, čo môže byť užitočné pri analýze textov, ktoré obsahujú zmiešané jazykové prostredie [12].

* mT5 (Multilingual T5) je multilingválny variant modelu T5, ktorý bol trénovaný na veľkom množstve jazykov vrátane slovenčiny. Model využíva rovnaký prístup "text-to-text", čo znamená, že je schopný riešiť rôzne NLP úlohy ako preklad, sumarizovanie a klasifikáciu textu. Jeho výhodou je univerzálnosť a schopnosť pracovať s viacerými jazykmi naraz, čo môže byť veľmi užitočné v prípadoch, keď sa texty nenachádzajú iba v slovenčine, ale aj v iných jazykoch. Pre úlohy detekcie nenávistnej reči v slovenčine môže byť model mT5 veľmi efektívny, najmä ak je potrebné riešiť viacjazyčné prostredie [8].
* XLM-R (XLM-RoBERTa) je ďalší multilingválny model, ktorý je založený na architektúre RoBERTa, ktorá je variantom BERT. Tento model bol trénovaný na 100 jazykoch, vrátane slovenčiny, a je veľmi efektívny pri riešení úloh ako klasifikácia textu a extrakcia informácií, aj pri obmedzených dátach pre konkrétny jazyk. XLM-R je veľmi silný model na úlohy, ktoré vyžadujú robustnosť voči viacerým jazykom, a je schopný zvládnuť aj náročné úlohy ako detekcia nenávistnej reči v textoch, ktoré môžu obsahovať slang, neformálny jazyk alebo nejasné formulácie [12].
  + 1. Veľké uzavreté modely (GPT-4, Gemini)

Okrem špecializovaných a multilingválnych modelov existujú aj veľké generatívne modely, ako sú GPT-4 a Gemini, ktoré majú široké uplatnenie v oblasti generovania textu a spracovania prirodzeného jazyka. Tieto modely sú veľmi silné v generovaní textu, ale môžu byť tiež efektívne na úlohy klasifikácie, ako je detekcia nenávistnej reči [9].

* GPT-4, vyvinutý spoločnosťou OpenAI, je veľmi silný generatívny model, ktorý je schopný vykonávať rôzne úlohy vrátane klasifikácie textov, generovania textu, analýzy sentimentu a ďalších úloh spracovania textu. Hoci nebol špecificky trénovaný na slovenčinu, jeho schopnosti v zero-shot alebo few-shot learning umožňujú aplikovať ho aj na úlohy v slovenčine. Pre detekciu nenávistnej reči môže byť model GPT-4 veľmi účinný, pretože dokáže generovať relevantné odpovede alebo identifikovať nenávistné výrazy, aj keď nebol priamo trénovaný na slovenčinu [9].
* Gemini, vyvinutý spoločnosťou Google DeepMind, je podobný modelu GPT-4, ale zameriava sa na veľmi vysokú presnosť a univerzálnosť pri riešení rôznych úloh spracovania textu. Tento model je vhodný na rôzne NLP úlohy, vrátane analýzy textov a detekcie nenávistnej reči. Gemini sa považuje za model, ktorý je schopný poskytovať veľmi presné výsledky aj v prípadoch, kde sú požiadavky na vysokú kvalitu generovaných výstupov alebo analýzy [9].
  1. Prehľad datasetov pre hate speech detection v slovenčine

Pri vývoji modelov na detekciu nenávistnej reči je jedným z kľúčových aspektov dostupnosť kvalitných datasetov. Tieto datasety poskytujú vzory textov, ktoré môžeme použiť na trénovanie a testovanie modelov. Pre slovenčinu existuje niekoľko datasetov, ktoré sú vhodné pre úlohy detekcie nenávistnej reči. Tento prehľad sa zameria na niektoré z najvýznamnejších datasetov, ktoré sú k dispozícii a môžu byť využité na detekciu nenávistnej reči v slovenskom jazyku [11].

* + 1. TUKE-KEMT Hate Speech dataset

TUKE-KEMT Hate Speech dataset je jeden z najznámejších a najviac používaných datasetov pre detekciu nenávistnej reči v slovenčine. Tento dataset bol vytvorený v rámci projektu na Technickej univerzite v Košiciach (TUKE) a Katedre elektroniky a multimediálnych technológií (KEMT). Dataset obsahuje texty zo sociálnych médií, kde sú označené konkrétne texty ako nenávistné alebo ne-nechcené (hate speech) [11].

Dataset je rozdelený do dvoch hlavných kategórií: nenávistná reč (hate speech) a ne-nenávistná reč (non-hate speech). Texty v datasetoch zahŕňajú rôzne formy nenávistného jazyka, ako sú rasistické, sexistické, homofóbne a iné formy diskriminácie. Tento dataset je veľmi užitočný pre trénovanie modelov, ktoré sa zameriavajú na identifikáciu urážlivých a nenávistných vyjadrení v slovenskom jazyku [11].

Jednou z výhod tohto datasetu je, že je anotovaný odborníkmi a obsahuje rôzne typy nenávistnej reči. Okrem toho dataset obsahuje aj popisné informácie o texte, ako je jeho dĺžka, počet znakov, počet slov, čo môže byť užitočné pre analýzu a doladenie modelu [11].

Vlastnosti datasetu:

* Veľkosť: Obsahuje tisíce príkladov textov z rôznych domén [11].
* Typy textu: Texty pochádzajúce z rôznych zdrojov, predovšetkým zo sociálnych médií [11].
* Kategórie: Kategorizované ako nenávistná a ne-nenávistná reč [11].
* Použitie: Tento dataset je vhodný na trénovanie a testovanie modelov na detekciu nenávistnej reči a pre výskumné účely v oblasti analýzy textu [11].
  + 1. Iné dostupné datasetové zdroje

Okrem datasetu TUKE-KEMT Hate Speech existujú aj ďalšie datasetové zdroje, ktoré môžu byť využité pre detekciu nenávistnej reči v slovenčine. Tieto datasety môžu byť menšie, ale stále užitočné, najmä pre testovanie modelov alebo pre špecifické úlohy [11]. Niektoré z nich zahŕňajú:

* SLOVAK-HATE dataset: Tento dataset je zameraný na detekciu nenávistnej reči v slovenčine a obsahuje texty, ktoré sú označené ako nenávistné a ne-nenávistné. Obsahuje texty z rôznych internetových diskusií, fóra a sociálnych sietí. Tento dataset môže byť užitočný pre trénovanie modelov, ktoré sa zameriavajú na analýzu sentimentu a klasifikáciu textov [11].
* Slovenský dataset pre detekciu nenávistnej reči na Twitteri: Tento dataset bol vytvorený pomocou skríningových nástrojov na získavanie tweetov obsahujúcich nenávistné výrazy a urážky. Obsahuje texty, ktoré boli manuálne označené ako nenávistné alebo ne-nenávistné. Tento dataset je špecifický pre slovenčinu a vhodný na analýzu textov, ktoré sú stručné a majú špecifický jazykový štýl charakteristický pre sociálne siete [11].
* Prekladačské datasety: Niektoré datasety, ktoré boli pôvodne určené na prekladanie textov, môžu byť tiež využité na detekciu nenávistnej reči, najmä ak obsahujú texty z rôznych jazykov. Tieto datasety môžu byť vhodné na trénovanie multilingválnych modelov, ktoré dokážu detekovať nenávistnú reč v rôznych jazykoch, vrátane slovenčiny [11].
* Reddit Hate Speech Dataset: Tento dataset je založený na diskusiách a komentároch z populárnej platformy Reddit. Hoci je tento dataset primárne v angličtine, niektoré z textov môžu obsahovať slovenské výrazy a frázy. Tento dataset môže byť užitočný pre modely, ktoré sa zameriavajú na detekciu nenávistnej reči v rôznych jazykoch [11].
  + 1. Vlastnosti a výzvy datasetov pre hate speech detection v slovenčiny

Pri výbere datasetu pre detekciu nenávistnej reči v slovenčine je potrebné zvážiť niekoľko faktorov [11]:

* Kvalita anotácie: Dataset musí byť správne anotovaný odborníkmi, aby bolo možné efektívne trénovať modely. Nesprávne alebo nejednoznačné označenie môže viesť k zníženiu výkonu modelu [11].
* Diverzita textov: Pre modely, ktoré sa používajú na detekciu nenávistnej reči, je dôležité, aby dataset obsahoval rôzne formy nenávistného jazyka, ako sú osobné útoky, diskriminácia, rasizmus a iné. Rôzne domény textov, ako sú sociálne médiá, diskusné fóra, spravodajské články a blogy, môžu prispieť k lepšiemu pokrytiu a robustnosti modelu [11].
* Veľkosť datasetu: Čím väčší dataset, tým viac dát má model k dispozícii na tréning, čo vedie k lepšiemu pochopeniu vzorcov a štruktúr v texte. Pre efektívne trénovanie modelov na detekciu nenávistnej reči je potrebné mať veľký dataset s rôznorodými príkladmi [11].
* Jazyková štruktúra: Slovenčina má špecifickú gramatiku a syntax, preto je dôležité, aby dataset bol prispôsobený slovenským jazykovým štruktúram. Preto datasety zamerané na slovenčinu (ako TUKE-KEMT a Slovenský dataset pre hate speech) sú lepšou voľbou ako datasety z iných jazykov [11].
  1. Prehľad metód na hate speech detection

Detekcia nenávistnej reči (hate speech detection) je jednou z najdôležitejších úloh v oblasti spracovania prirodzeného jazyka (NLP). S cieľom efektívne rozpoznať nenávistné vyjadrenia v texte, výskumníci a vývojári používajú rôzne metódy a prístupy. Tento prehľad sa zameria na niektoré z hlavných metód, ktoré sú využívané v úlohách detekcie nenávistnej reči, ako sú klasifikácia, prompting a fine-tuning [5][6].

* + 1. Klasifikácia textu

Klasifikácia textu je základnou metódou používanou na detekciu nenávistnej reči. Tento prístup zahŕňa proces, pri ktorom je text automaticky klasifikovaný do niekoľkých preddefinovaných kategórií, napríklad "nenávistná reč" (hate speech) a "ne-nenávistná reč" (non-hate speech). Modely trénované na túto úlohu sa naučia rozoznať vzory a charakteristiky, ktoré sú typické pre nenávistné vyjadrenia.

Pre klasifikáciu sa často používajú rôzne druhy modelov, ako sú:

* Štatistické modely: Tradičné metódy ako Naivní Bayes alebo SVM (Support Vector Machine) sa môžu použiť pre textovú klasifikáciu. Tieto metódy sú efektívne na menších datasetoch, ale sú menej presné na komplexné úlohy ako detekcia nenávistnej reči [5].
* Neuronové siete: Pokročilejšie metódy, ako sú rekurentné neuronové siete (RNN) alebo konvolučné neuronové siete (CNN), sa čoraz častejšie používajú pre túto úlohu. Tieto modely sú schopné spracovať dlhšie texty a extrahovať zložitejšie vzory, ktoré sú potrebné pre rozpoznávanie nenávistných komentárov [6].
* Transformerové modely: Moderné prístupy na detekciu nenávistnej reči využívajú transformerové architektúry, ako je BERT, RoBERTa, T5 a ďalšie. Tieto modely sú schopné pochopiť zložité vzťahy v texte a poskytujú veľmi presné výsledky pri detekcii nenávistných vyjadrení. Trénovanie takýchto modelov však vyžaduje väčšie množstvo dát a výpočtových prostriedkov [6][8].

Pri klasifikácii textu sa modely trénujú na rôznych datasetoch obsahujúcich nenávistnú reč. Výsledky klasifikácie môžu byť hodnotené pomocou rôznych metrik, ako sú presnosť (accuracy), precíznosť (precision), zber (recall) a F1-skóre. Tieto metriky pomáhajú určiť, ako dobre model rozpoznáva nenávistnú reč a zároveň minimalizuje falošné pozitíva [10].

* + 1. Prompting

Prompting je technika, ktorá sa stala veľmi populárnou v súvislosti s veľkými jazykovými modelmi ako GPT-3 alebo GPT-4 [9]. V tomto prístupe sa modelom zadáva špecifický "prompt" (predloha), ktorý definuje úlohu, ktorú má model vykonať. V kontexte detekcie nenávistnej reči je prompt formulovaný tak, aby model rozpoznal, či daný text obsahuje nenávistné vyjadrenie.

#### Například prompt môže byť: "Určte, či nasledujúci text obsahuje nenávistnú reč: [vložte text]". Model potom poskytne odpoveď typu "áno" alebo "nie" na základe analýzy daného textu. [9]

Výhody prompting prístupu:

* Flexibilita: Prompting umožňuje rýchlo prispôsobiť model na rôzne úlohy bez potreby opätovného trénovania modelu [9].
* Zero-shot a few-shot učenie: Pomocou správne formulovaných promptov môžu byť modely schopné vykonať úlohu, aj keď neboli explicitne trénované na konkrétnom datasete. Zero-shot učenie znamená, že model je schopný rozpoznať nenávistnú reč bez predchádzajúceho tréningu na tejto úlohe, zatiaľ čo few-shot učenie umožňuje modelu učiť sa z malého počtu príkladov [9].

Nevýhody:

* Obmedzené pokrytie: Prompting môže byť menej efektívny v prípadoch, kde je potrebné rozpoznať veľmi komplexné alebo jemné formy nenávistnej reči [9].
* Závislosť od formulácie promptu: Kvalita výstupu závisí od správnej formulácie promptu. Nesprávne formulované prompty môžu viesť k nesprávnym alebo nepresným výsledkom [9].
  + 1. Fine-tuning (doladenie)

Fine-tuning je proces, pri ktorom sa predtrénovaný model prispôsobí konkrétnej úlohe, ako je detekcia nenávistnej reči. Tento prístup je veľmi efektívny, keď sú k dispozícii kvalitné datasety, pretože modely ako BERT, T5 alebo mT5 už obsahujú základné jazykové schopnosti, ktoré sú ďalej vylepšené pre konkrétnu úlohu [8].

Proces fine-tuningu:

* Predtrénovanie modelu: Model je predtrénovaný na veľkom korpuse textov, aby získal základné jazykové schopnosti [8].
* Doladenie na špecifický dataset: Model je následne doladený na menší dataset s označenými príkladmi nenávistnej reči. Tento krok zahŕňa učenie modelu na rozpoznávanie vzorcov špecifických pre úlohu detekcie nenávistnej reči [8].

Výhody fine-tuningu:

* Vysoká presnosť: Doladené modely dosahujú veľmi vysokú presnosť pri detekcii nenávistnej reči, pretože model je optimalizovaný pre túto úlohu [8].
* Flexibilita: Fine-tuning umožňuje prispôsobiť model rôznym jazykom, doménam a formám nenávistnej reči [8].

Nevýhody:

* Potrebné veľké dáta: Fine-tuning vyžaduje kvalitný dataset s anotovanými príkladmi, čo môže byť náročné a časovo náročné [8].
* Výpočtová náročnosť: Fine-tuning je náročný na výpočtové prostriedky a čas [8].
  1. Spôsoby hodnotenia modelov a metriky

Hodnotenie výkonu modelov v oblasti detekcie nenávistnej reči je kľúčovým krokom, ktorý umožňuje určiť, aké účinné sú modely pri vykonávaní tejto úlohy. Na vyhodnotenie používame rôzne metriky, ako sú Precision, Recall a F1-score, ktoré poskytujú ucelený obraz o schopnostiach modelu. Okrem týchto tradičných metrík, ktoré sa používajú na vyhodnotenie klasifikačných modelov, existujú aj špecifické nástroje a knižnice na hodnotenie jazykových modelov, ako je lm\_eval\_harness a iné [10].

* + 1. Klasické metriky hodnotenia

Precision (Presnosť) – Tento ukazovateľ hodnotí, koľko z textov, ktoré model označil ako nenávistné, je skutočne nenávistných. Je to dôležité pre minimalizáciu falošných pozitívnych výsledkov, keď model nesprávne označí bezpečný text za nenávistný [10].

Recall (Zachytiteľnosť) – Tento ukazovateľ meria, aký podiel skutočne nenávistných textov model dokáže správne identifikovať. Dôležité je, aby model nezmeškal veľkú časť nenávistných textov, aj keď to môže viesť k vyšším falošným pozitívam [10].

F1-score – Tento ukazovateľ je harmonický priemer Precision a Recall. Je veľmi užitočný pri hodnotení modelov, kde je potrebné dosiahnuť rovnováhu medzi presnosťou a zachytiteľnosťou, pretože poskytuje komplexný obraz o výkone modelu [10].

* + 1. lm\_eval\_harness – Knižnica na hodnotenie modelov

lm\_eval\_harness je knižnica vyvinutá na efektívne hodnotenie jazykových modelov, najmä tých, ktoré sú zamerané na úlohy textovej klasifikácie, generovania textu, sumarizovania a ďalšie úlohy spracovania prirodzeného jazyka. Táto knižnica umožňuje hodnotenie modelov na rôznych typoch úloh, a to s využitím preddefinovaných datasetov a metód hodnotenia.

Hlavným cieľom lm\_eval\_harness je poskytnúť univerzálny rámec na testovanie výkonu jazykových modelov, ktorý zahŕňa:

* Štandardizované hodnotenie: Umožňuje používateľom porovnať rôzne modely a ich výkonnosť na rovnakých úlohách a datasetoch [10].
* Podpora rôznych metrík: Knižnica podporuje širokú škálu metrík na hodnotenie výkonu modelu, vrátane Accuracy, Precision, Recall, F1-score, a ďalších metód, ako je napríklad perplexity alebo BLEU score pre generatívne úlohy [10].
* Podpora rôznych modelov: lm\_eval\_harness podporuje množstvo rôznych modelov, ako sú BERT, GPT-2, GPT-3, T5, RoBERTa, a ďalšie. To umožňuje jednoducho vykonávať benchmarky medzi rôznymi jazykovými modelmi a získať výsledky, ktoré sú dôležité pri rozhodovaní, ktorý model je najvhodnejší pre danú úlohu [10].
* Použitie tejto knižnice zjednodušuje proces hodnotenia modelov a poskytuje štruktúrovaný spôsob, ako analyzovať výkonnosť rôznych jazykových modelov pri rôznych úlohách [10].
  + 1. Iné metódy hodnotenia

Okrem lm\_eval\_harness existujú aj ďalšie prístupy na hodnotenie jazykových modelov, ktoré môžeme využiť pre detekciu nenávistnej reči [10]:

* Manual Evaluation (Manuálne hodnotenie): V niektorých prípadoch môže byť potrebné ručne hodnotiť výsledky modelu. Táto metóda zahŕňa kontrolu predikcií modelu ľudskými hodnotiteľmi, ktorí rozhodnú, či model správne identifikoval nenávistnú reč. Tento prístup je časovo náročný, ale môže poskytnúť veľmi podrobné a presné hodnotenie [10].
* Cross-validation: Táto metóda zahŕňa rozdelenie datasetu na viacero častí a hodnotenie modelu na rôznych častiach dát, čím sa minimalizuje riziko pretrénovania. Cross-validation poskytuje lepšiu predstavu o schopnostiach modelu generalizovať na nevidené dáta [10].
* A/B Testing: Tento prístup umožňuje porovnať rôzne verzie modelov a určiť, ktorý model dosahuje lepšie výsledky na testovacej sade. A/B testovanie je obzvlášť užitočné pri porovnávaní nových metód trénovania alebo rôznych architektúr modelov [10].

1. Návrh a implementácia riešenia zvolenej problematiky

Táto kapitola sa zameriava na návrh a implementáciu riešenia vybranej problematiky, pričom sa špecificky venujeme modelom strojového učenia a ich adaptácii pomocou metódy LoRA. Cieľom je vytvoriť efektívny model na detekciu nenávistnej reči v slovenskom jazyku, pričom využijeme pokročilé techniky doladenia veľkých jazykových modelov. Analyzujeme výber modelu, jeho trénovanie, aplikáciu few-shot učenia a vyhodnotenie výkonnosti pomocou metrických ukazovateľov.

V súčasnosti je detekcia nenávistnej reči kľúčovou úlohou v oblasti spracovania prirodzeného jazyka (NLP). Pre efektívne riešenie tejto úlohy je dôležité vybrať vhodné jazykové modely, ktoré majú dostatočnú kapacitu porozumenia kontextu v slovenskom jazyku. Naša práca sa zameriava na optimalizáciu modelov s využitím metód doladenia, ktoré minimalizujú výpočtovú náročnosť pri zachovaní vysokej presnosti.

* 1. Výber modelu a metódy

Pri výbere modelu na úlohu detekcie nenávistnej reči sme sa zamerali nielen na architektúru modelu, ale aj na efektívnosť metód, ktoré umožňujú optimalizovať výkon modelov pri spracovaní slovenského jazyka. Kľúčové metódy, ktoré sme použili, zahŕňajú few-shot learning, vyváženosť dát a doladenie modelov pomocou LoRA a PEFT. Tieto prístupy nám umožnili zlepšiť výsledky aj pri obmedzených dátach a znížiť výpočtovú náročnosť pri trénovaní modelov.

* + 1. Výber modelu
       1. Monolingválne modely

Pri výbere modelov sme sa rozhodli pre monolingválne modely, ktoré boli špecificky trénované na spracovanie textov v slovenčine. Takéto modely, ako Slovak-T5-base, Slovak-T5-small a SlovakBERT, vykazujú lepšie výsledky pri detekcii nenávistnej reči v slovenčine, pretože sa prispôsobujú jazyku a jeho špecifikám. Tieto modely sú schopné využiť jazykové vzory a kontextové informácie charakteristické pre slovenský jazyk, čím dosahujú vyššiu presnosť v porovnaní s modelmi, ktoré sú trénované na viacerých jazykoch.

Slovak-T5-base a Slovak-T5-small sú modely postavené na architektúre T5, ktorá sa osvedčila pri rôznych úlohách spracovania textu. Slovak-T5-base je väčšia verzia modelu, ktorá je schopná poskytnúť lepšie výsledky pri náročnejších úlohách, avšak s vyššími nárokmi na výpočtové zdroje. Slovak-T5-small je menšia verzia, ktorá si vyžaduje menej výpočtových prostriedkov, ale stále poskytuje silné výsledky v analýze slovenských textov. Tieto modely boli trénované špecificky na slovenskom jazyku, čo im umožnilo lepšie pochopiť jemnosti a nuansy tohto jazyka v porovnaní s multilingválnymi modelmi.

SlovakBERT je ďalší model špecifický pre slovenčinu a je založený na architektúre BERT. Tento model bol trénovaný na veľkých množstvách slovenských textov a je veľmi efektívny pri úlohách, ako je analýza sentimentu a klasifikácia textov. Jeho silnou stránkou je schopnosť rozpoznať a porozumieť kontextu v slovenskom jazyku, čo prispieva k jeho úspechu pri detekcii nenávistnej reči v slovenčine.

* + - 1. Multilingválne modely

Okrem monolingválnych modelov sme zvážili aj multilingválne modely, ako sú mT5-small, mT5-base a mT5-large, ktoré sú trénované na rôznych jazykoch. Tieto modely majú širšiu generalizačnú schopnosť, ale ich výkon na konkrétnom jazyku, ako je slovenčina, môže byť nižší v porovnaní s modelmi špecifickými pre tento jazyk. Výhodou multilingválnych modelov je ich flexibilita a schopnosť pracovať s textami v rôznych jazykoch, čo je dôležité pri práci s viacjazyčnými dátami. Avšak, ako sa ukázalo v našich testoch, modely ako mT5 vykazovali nižšiu presnosť pri analýze slovenských textov v porovnaní s modelmi špecifickými pre slovenčinu.

mT5-small, mT5-base a mT5-large sú modely založené na architektúre T5, ktoré boli trénované na rôznych jazykoch, vrátane slovenčiny. Modely mT5 majú výhodu v tom, že sú schopné pracovať s viacerými jazykmi, čo je ideálne pri analýze textov v rôznych jazykoch v rámci jedného projektu. Avšak pri špecifických úlohách, ako je detekcia nenávistnej reči v slovenčine, sa ukázalo, že tieto modely nedosahujú rovnakú úroveň presnosti ako monolingválne modely, ktoré boli špecificky trénované na slovenskom jazyku.

* + 1. Metódy
       1. Few-shot learning

Few-shot learning (FSL) je metodológia, ktorá umožňuje modelu učiť sa z malého množstva trénovacích príkladov. Tento prístup je veľmi užitočný v prípadoch, keď máme obmedzený počet dát pre konkrétnu úlohu, ako je napríklad detekcia nenávistnej reči. Tradičné metódy strojového učenia často vyžadujú veľké množstvo dát na to, aby sa model naučil identifikovať relevantné vzory. Avšak v reálnych situáciách môžeme mať len obmedzený počet príkladov pre špecifické jazyky alebo kategórie.

V našich experimentoch sme implementovali few-shot learning s cieľom znížiť potrebu rozsiahleho trénovania modelu, pričom sme vychádzali z existujúcich, predtrénovaných modelov, ako sú mT5 alebo SlovakBERT. Tieto modely poskytujú dobrý základ, ktorý sa ďalej prispôsobuje na špecifickú úlohu detekcie nenávistnej reči. V praxi to znamená, že aj s malým množstvom trénovacích dát môže model úspešne rozpoznať vzory v textoch a správne klasifikovať nenávistnú reč.

* + - 1. Vyváženosť dát (Balanced dataset)

Pri práci s modelmi na detekciu nenávistnej reči je mimoriadne dôležité zabezpečiť vyváženosť dát, teda rovnaký počet príkladov pre každú triedu (napr. nenávistná reč vs. nenávistná reč). Bez vyvážených dát môže model sklony k predpovediam pre jednu triedu (napríklad, že každý text je "neutrálne" označený), čo môže viesť k nízkej presnosti a skresleným výsledkom.

Pre tento účel sme sa rozhodli využiť techniky, ktoré umožňujú vyvážiť dáta. To zahŕňa metódy ako over-sampling (generovanie nových príkladov pre podreprezentovanú triedu), under-sampling (odstránenie príkladov z nadreprezentovanej triedy) a syntetické generovanie dát pomocou techník ako SMOTE (Synthetic Minority Over-sampling Technique). Tieto metódy pomáhajú znížiť riziko, že model bude zaujatý voči jednej triede, a zároveň zaisťujú, že model sa učí rovnako presne rozpoznávať oba typy textov.

* + - 1. Doladenie modelu pomocou LoRA a PEFT

LoRA (Low-Rank Adaptation) a PEFT (Parameter Efficient Fine-Tuning) sú techniky, ktoré umožňujú efektívne doladiť modely s minimálnymi nárokmi na výpočtové prostriedky. Vzhľadom na to, že modely ako mT5 a Slovak-T5 sú veľmi veľké a výpočtovo náročné, LoRA a PEFT poskytujú spôsob, ako prispôsobiť modely bez nutnosti preškoliť všetky parametre, čo výrazne znižuje potrebu pamäte a výpočtového výkonu.

LoRA využíva nízko-rankovú dekompozíciu matíc, čím umožňuje prispôsobiť len časť parametrov modelu, čím sa znižuje náročnosť na pamäť a výpočtový výkon. Táto metóda je zvlášť užitočná pri práci s veľkými modelmi ako mT5-large alebo mT5-base, keďže umožňuje efektívne doladiť model pre konkrétnu úlohu (ako je detekcia nenávistnej reči), bez potreby preškoliť celý model, čo by si vyžadovalo obrovské množstvo výpočtových zdrojov.

PEFT, podobne ako LoRA, umožňuje rýchle doladenie modelu s minimálnymi nárokmi na parametre. PEFT umožňuje prispôsobiť modely s minimálnym počtom nových parametrov, čo vedie k efektívnejšiemu tréningu a menšej spotrebe pamäte, čo je neoceniteľné pri práci s veľkými modelmi na detekciu nenávistnej reči.

* 1. Trénovanie vybraného modelu pomocou LoRA

Pri trénovaní modelov sme použili framework PEFT a knižnicu Hugging Face Transformers. Celý proces zahŕňa niekoľko krokov, ktoré sa detailne popíšu nižšie.

1. Načítanie predtrénovaného modelu

Na začiatku sme použili predtrénované modely, ako mT5-small, mT5-base a Slovak-T5, ktoré boli následne dotrénované na úlohu detekcie nenávistnej reči. Tieto modely sú špecifické pre slovenčinu, čo bolo kľúčové pre zlepšenie výsledkov.

# Načítanie predtrénovaného modelu

model = T5ForSequenceClassification.from\_pretrained("slovak-t5-small")

1. Príprava datasetu

Použili sme dataset hate\_speech\_slovak.jsonl, ktorý obsahuje anotované príklady textov so štítkami indikujúcimi prítomnosť alebo neprítomnosť nenávistnej reči. Tento dataset sme rozdelili na trénovaciu, validačnú a testovaciu množinu, pričom sme dbali na stratifikované rozdelenie podľa štítkov.

# Načítanie datasetu

dataset = load\_dataset('hate\_speech\_slovak.jsonl')

train\_dataset, val\_dataset, test\_dataset = split\_dataset(dataset)

1. Implementácia LoRA adaptácie

Aby sme optimalizovali trénovanie a znížili počet trénovaných parametrov, použili sme LoRA (Low-Rank Adaptation). LoRA umožňuje zamerať sa na nízkohodnotové aproximácie, čo je efektívne pri finálnom doladení predtrénovaných modelov. Implementácia LoRA zahŕňa definovanie konfigurácie pre adaptáciu modelu, ktorá sa aplikuje na každú vrstvu modelu.

# Konfigurácia LoRA

lora\_config = LoraConfig(r=8, lora\_alpha=32, lora\_dropout=0.1)

# Aplikácia LoRA adaptácie na model

model = apply\_lora(model, lora\_config)

1. Trénovanie modelu

Po aplikovaní LoRA na model sme nastavili tréningové parametre, ako sú veľkosť dávky, počet epoch a stratégiu učenia.

# Nastavenie tréningových parametrov

training\_args = TrainingArguments(

output\_dir='./results',

evaluation\_strategy="epoch",

learning\_rate=3e-5,

per\_device\_train\_batch\_size=8,

per\_device\_eval\_batch\_size=16,

num\_train\_epochs=7,

logging\_dir='./logs',

)

1. Vyhodnotenie modelu

Po skončení tréningu sme model hodnotili pomocou Precision, Recall a F1-skóre, ktoré sú štandardné metriky pre binárnu klasifikáciu. Model sme následne porovnali s rôznymi architektúrami, aby sme získali prehľad o jeho výkonnosti.

# Vyhodnotenie modelu

metrics = trainer.evaluate(test\_dataset)

1. Klasifikácia textu

Na záver sme implementovali funkciu pre klasifikáciu textov, ktorá umožňuje predpovedať, či daný text obsahuje nenávistnú reč. Táto funkcia slúži ako finálny krok pri nasadení modelu.

# Funkcia pre klasifikáciu textu

def classify\_text(text, model):

inputs = tokenizer(text, return\_tensors="pt")

with torch.no\_grad():

logits = model(\*\*inputs).logits

prediction = torch.argmax(logits, dim=-1)

return prediction

* 1. Aplikácia few-shot a prompting (príklady promptov)

V rámci experimentu sme sa zamerali na aplikáciu few-shot učenia a prompting techník pre vybrané modely, ktoré sme použili na klasifikáciu nenávistnej reči. Few-shot učenie umožňuje modelu adaptovať sa na nové úlohy s minimálnym množstvom príkladov počas trénovania, čo je veľmi užitočné pri spracovaní špecifických jazykových úloh, ako je detekcia nenávistnej reči v slovenskom jazyku.

Použili sme rôzne modely, medzi ktoré patrili mT5-small, mT5-base, mT5-large, Slovak-T5-base, Slovak-T5-small, a SlovakBERT. Pre overenie výkonnosti modelov sme použili knižnicu lm-eval-harness, ktorá nám umožnila porovnať výsledky v zero-shot a few-shot scenároch.

* + 1. Few-shot učenie

Few-shot učenie sa zameriava na poskytovanie modelu len niekoľkých príkladov, na základe ktorých dokáže generovať predpovede aj pre nové, nevidené texty. Tento prístup je efektívny, keď je množstvo trénovacích dát obmedzené, čo je častý problém pri spracovaní špecifických jazykových úloh, ako je napríklad detekcia nenávistnej reči.

* + 1. Prompting

Prompting je technika, ktorá sa využíva na poskytovanie konkrétnych pokynov modelu vo forme textových otázok alebo príkazov. Na základe promptu model generuje odpoveď, ktorá je vhodná pre danú úlohu. V našom prípade sme vytvorili prompt, ktorý modelu poskytol kontext a požiadal ho o klasifikáciu textu.

Príklady promptov:

1. Príklad 1:

* Vstupný text: "Nenávidím ťa a všetkých ako ty."
* Otázka (Prompt): "Obsahuje text nenávistnú reč?"
* Výstup: "Áno"

2. Príklad 2:

* Vstupný text: "Zamaličko"
* Otázka (Prompt): "Obsahuje text nenávistnú reč?"
* Výstup: "Nie"

V oboch prípadoch model využil minimálne množstvo príkladov a na základe daného promptu dokázal efektívne rozpoznať, či text obsahuje nenávistnú reč alebo nie.

* + 1. Zlepšenie presnosti

Pomocou tejto metódy sa nám podarilo dosiahnuť zlepšenie presnosti modelu, najmä v prípade, keď sme použili špecifické slovenské modely ako Slovak-T5 a SlovakBERT. Tieto modely sú lepšie prispôsobené pre detekciu nenávistnej reči v slovenskom jazyku, pretože boli trénované na dátach v slovenskom jazyku a dokážu lepšie zachytiť nuansy jazyka, ktoré by mohli byť pre iné modely náročné.

* + 1. Význam few-shot učenia a prompting techník

Few-shot učenie a prompting techniky sú veľmi efektívne pri práci s modelmi na špecifické úlohy, kde máme obmedzené množstvo trénovacích dát alebo chceme prispôsobiť model na konkrétne spracovanie úloh, ako je klasifikácia nenávistnej reči. Tieto techniky umožňujú rýchlu adaptáciu modelu na nové úlohy a výrazne zvyšujú presnosť detekcie aj pri malých množstvách trénovacích príkladov.

* 1. Vyhodnotenie pomocou metrik Precision, Recall, F1

Pre hodnotenie výkonu modelov pri úlohe detekcie nenávistnej reči sme použili tri základné metriky: Precision, Recall a F1-score. Tieto metriky nám pomohli objektívne posúdiť, ako efektívne modely vykonávajú klasifikáciu a identifikujú nenávistnú reč.

* + 1. Použitie sklearn.metrics:

sklearn.metrics je knižnica v Python-e, ktorá poskytuje rôzne nástroje na vyhodnotenie výkonnosti klasifikačných modelov. Pre našu úlohu sme využili nasledovné metriky:

* Precision (Presnosť): Precision nám hovorí, aký podiel pozitívnych predpovedí (predikcií nenávistnej reči) bol správny. Inými slovami, ak model označil text ako nenávistný, koľkokrát mal pravdu.
* Recall (Úplnosť): Recall zohľadňuje, koľko skutočných pozitívnych prípadov (skutočnej nenávistnej reči) bolo modelom správne identifikovaných. Inými slovami, ak v texte bola nenávistná reč, recall nám hovorí, ako často ju model správne označil.
* F1-score: F1-score je harmonický priemer medzi Precision a Recall, ktorý je užitočný v prípadoch, kde je dôležité mať vyvážený výkon v oboch metrikách. Používa sa najmä v prípadoch, kde je nevyvážený počet pozitívnych a negatívnych príkladov.

Pre tieto metriky sme využili funkcie z knižnice sklearn.metrics, konkrétne precision\_recall\_fscore\_support, ktorá vracia hodnoty Precision, Recall a F1-score pre danú množinu predikcií.

# Pseudokód pre výpočet Precision, Recall a F1-score pomocou metódy precision\_recall\_fscore\_support

Načítaj knižnicu 'precision\_recall\_fscore\_support' z 'sklearn.metrics'

# Zavolaj funkciu precision\_recall\_fscore\_support s reálnymi hodnotami (y\_true) a predikovanými hodnotami (y\_pred)

# Určte, že chcete hodnotiť pre binárnu klasifikáciu s priemerom 'binary'

precision, recall, f1, \_ = precision\_recall\_fscore\_support(y\_true, y\_pred, average='binary')

# Výsledkom sú hodnoty:

# - precision: Presnosť modelu

# - recall: Citlivosť modelu

# - f1: F1-skóre

Tu y\_true predstavuje skutočné hodnoty (štítky) a y\_pred predikcie modelu. Metriky sa počítajú na základe porovnania týchto hodnôt.

* + 1. Použitie lm\_harness:

lm\_harness je framework určený na vyhodnocovanie výkonu jazykových modelov, najmä v kontexte veľkých jazykových modelov (LLM), ako je GPT alebo T5. Tento nástroj je užitočný pre meranie výkonu modelov v rôznych scenároch, ako sú zero-shot a few-shot učenie.

Zero-shot znamená, že model vykonáva úlohu bez toho, aby bol špecificky trénovaný na daný typ úlohy.

Few-shot znamená, že model dostane niekoľko príkladov (promptov) na základe ktorých sa učí, čo zlepšuje jeho schopnosť vykonávať danú úlohu.

V rámci lm\_harness sme použili nasledovné metriky:

* Precision a Recall boli definované podobne ako v sklearn.metrics, ale v kontexte hodnotenia modelov v scenároch zero-shot a few-shot.
* F1-score je opäť metrika, ktorá kombinuje Precision a Recall, ale je vhodnejšia na hodnotenie modelov v prípadoch, kde máme nevyvážené triedy.

Pre hodnotenie pomocou lm\_harness sme využívali funkcie, ktoré umožnili vyhodnotiť rôzne modely v týchto scenároch. Použitie metriky F1-score bolo obzvlášť dôležité pri porovnávaní modelov, ktoré vykazovali rôzne výsledky v oblasti presnosti a úplnosti.

# Pseudokód pre výpočet Precision, Recall a F1-score pomocou knižnice lm\_eval.metrics

Načítaj knižnicu 'precision', 'recall', 'f1' z 'lm\_eval.metrics'

# Zavolaj funkcie na výpočet metrik

precision\_score = precision(y\_true, y\_pred) # Výpočet presnosti modelu

recall\_score = recall(y\_true, y\_pred) # Výpočet citlivosti modelu

f1\_score = f1(y\_true, y\_pred) # Výpočet F1-skóre

Tento nástroj nám umožnil dôkladne porovnať výkon modelov na detekciu nenávistnej reči v slovenskom jazyku a získať dôležité informácie o tom, ako jednotlivé modely zvládajú túto úlohu.

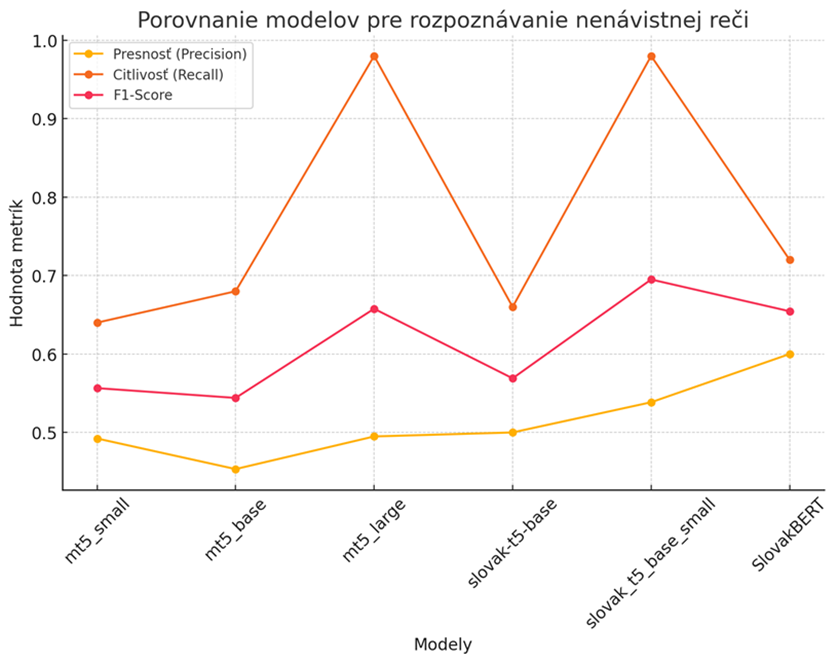
* 1. Prehľad experimentov a výsledkov (tabuľky, grafy)

Tab. 1 Výsledky tréningu modelu pomocou few-shot learning

|  |  |  |  |
| --- | --- | --- | --- |
| Modely | Precision | Recall | F1 Score |
| mT5-small | 0.4923 | 0.6400 | 0.5565 |
| mT5-base | 0.4533 | 0.6800 | 0.5440 |
| mT5-large | 0.4949 | 0.9800 | 0.6577 |
| Slovak-T5-base | 0.5000 | 0.6600 | 0.5690 |
| Slovak-T5-base-small | 0.5385 | 0.9800 | 0.6950 |
| SlovakBERT | 0.6000 | 0.7200 | 0.6545 |

Analýza výsledkov:

* SlovakBERT dosiahol najlepšie výsledky v Precision (0.6000) a Recall (0.7200), čo znamená, že tento model je veľmi presný v identifikácii nenávistnej reči, pričom má aj veľmi dobrú schopnosť zachytiť všetky pozitívne prípady.
* Slovak-T5-base a Slovak-T5-base-small taktiež dosiahli veľmi solídne výsledky, hlavne v Recall s hodnotami 0.6600 a 0.9800, čo ukazuje, že tieto modely sú veľmi citlivé na detekciu nenávistných textov.
* mT5-large sa ukázal ako veľmi silný v Recall (0.9800), avšak jeho Precision (0.4949) naznačuje, že model má tendenciu označovať viac textov ako nenávistné, než je skutočne pravda, čo vedie k menšej presnosti.



Obr. 1 Porovnanie modelov (graf vytvorený pomocou knižnice matplolib)

Analýza výsledkov ukázala, že modely trénované na slovenskom jazyku, ako SlovakBERT a Slovak-T5-base, dosiahli najlepšie výsledky v detekcii nenávistnej reči. Napriek tomu, že mT5-large vykázal vysokú mieru Recall (0.9800), jeho nižšia Precision (0.4949) naznačuje, že model často označuje texty ako nenávistné aj v prípadoch, keď tomu tak nie je.

Na základe týchto výsledkov som sa rozhodla aplikovať metódu LoRA (Low-Rank Adaptation) na model Slovak-T5-base-small, ktorý poskytoval vysokú Recall (0.9800), ale potreboval optimalizáciu v oblasti Precision.

Po aplikácii LoRA na Slovak-T5-base-small sme dosiahli nasledujúce výsledky:

* Precision: 0.5435
* Recall: 1.0000
* F1-score: 0.7042

Tieto výsledky ukazujú, že jemné doladenie modelu pomocou LoRA pomohlo zvýšiť Precision oproti pôvodnému Slovak-T5-base-small, pričom Recall zostal na maximálnej hodnote. To naznačuje, že model je schopný správne identifikovať väčšinu prípadov nenávistnej reči bez výrazného zvýšenia falošných pozitív.

* 1. Vývoj webovej stranky a Telegram bota
     1. Vývoj webovej strankgy

Cieľom webovej aplikácie bolo vytvoriť jednoduché a intuitívne používateľské rozhranie, ktoré umožňuje testovanie vytvoreného modelu na rozpoznávanie nenávistnej reči. Táto časť práce slúži ako doplnok k hlavnému výskumu a poskytuje praktickú ukážku využitia modelu v reálnom čase.

* + - 1. Použité technológie

Na vývoj webovej aplikácie boli použité nasledujúce technológie:

* Frontend: HTML, CSS a React
* Backend: Python s použitím frameworku Flask
* Model: implementovaný pomocou knižníc transformers a PyTorch
* Dáta: Aplikácia nevyužíva databázu, všetky požiadavky sú spracovávané v reálnom čase
  + - 1. Architektúra riešenia

Aplikácia je postavená na klient-server architektúre. Používateľ zadáva vstupný text cez rozhranie vytvorené v Reacte. Tento text je následne odoslaný na server (Flask), kde ho spracuje predtrénovaný model založený na transformer architektúre.

# Serverová časť – Flask

@app.route('/predict', methods=['POST'])

def predict():

text = request.json['text'] # Získame text od klienta

inputs = tokenizer(text, return\_tensors='pt') # Tokenizácia textu

outputs = model(\*\*inputs) # Spracovanie cez model

prediction = torch.argmax(outputs.logits, dim=-1).item()

# Klasifikácia na základe výstupov modelu

return jsonify({'result': "Pravdepodobne toxický" if prediction == 1 else "Neutrálny text"}) # Výsledok predikcie

Výsledok klasifikácie (napr. „Nenávistná reč“ alebo „Bezpečný obsah“) sa potom okamžite zobrazí používateľovi.

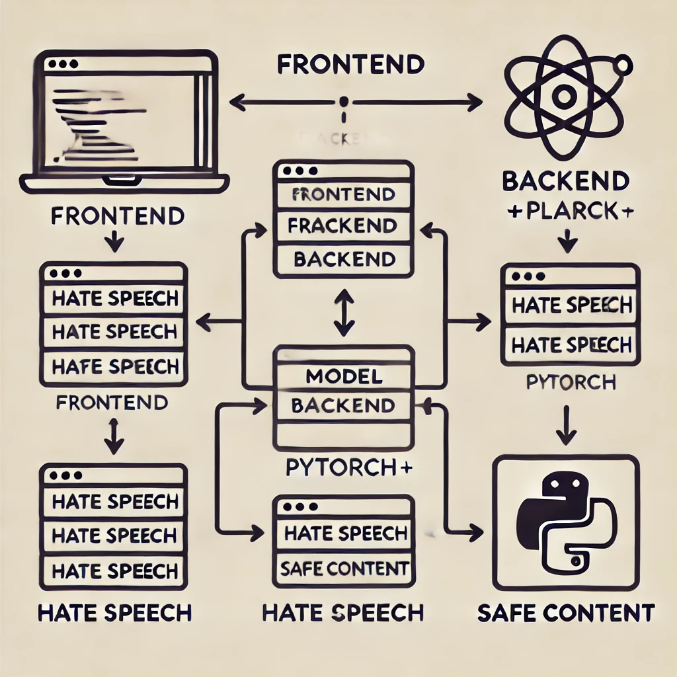
// Klientská časť - React

const handleSubmit = async () => {

const response = await axios.post('http://localhost:5000/predict', { text });

setResult(response.data.result); // Zobrazíme výsledok na obrazovke

};

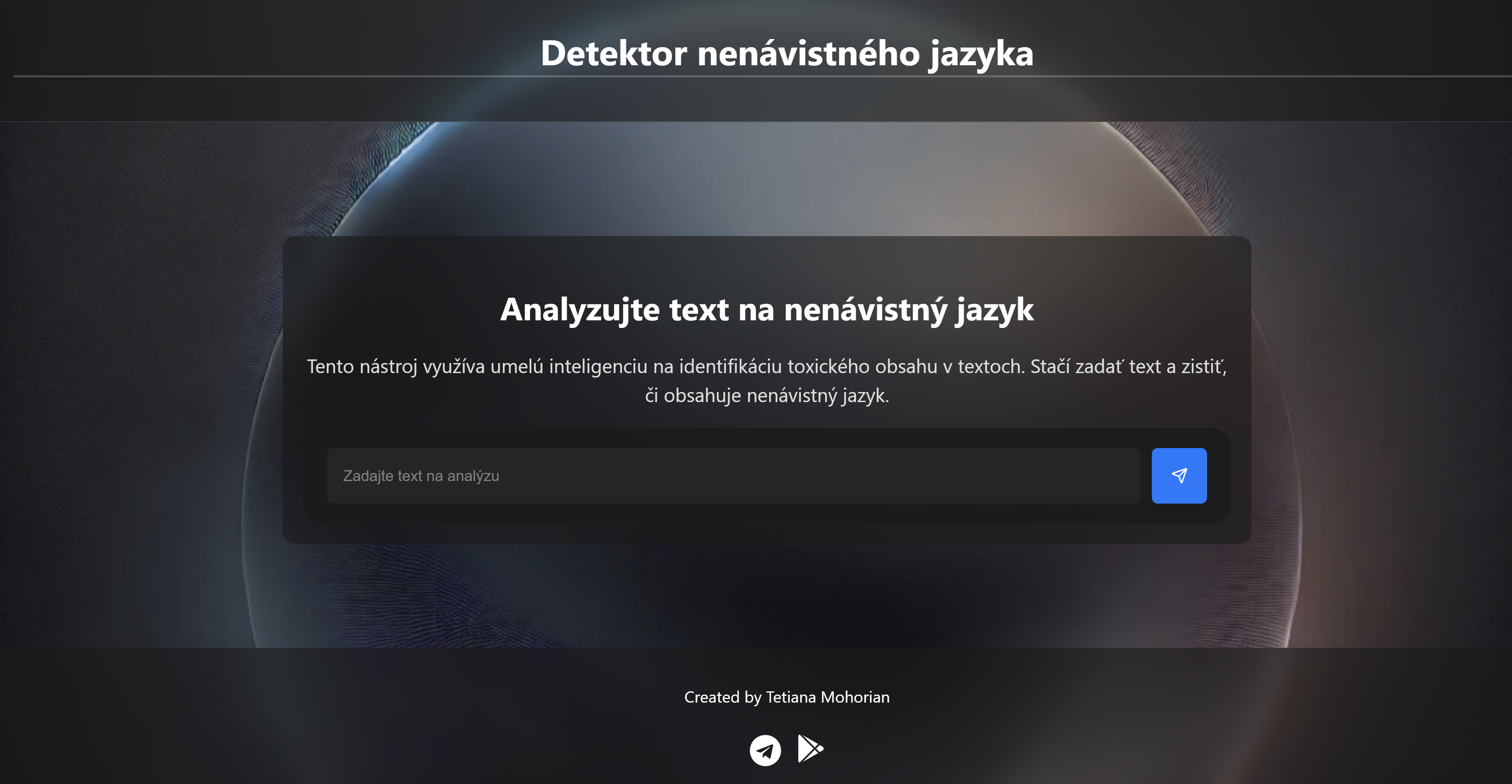


Obr. 2 Architektúra riešenia

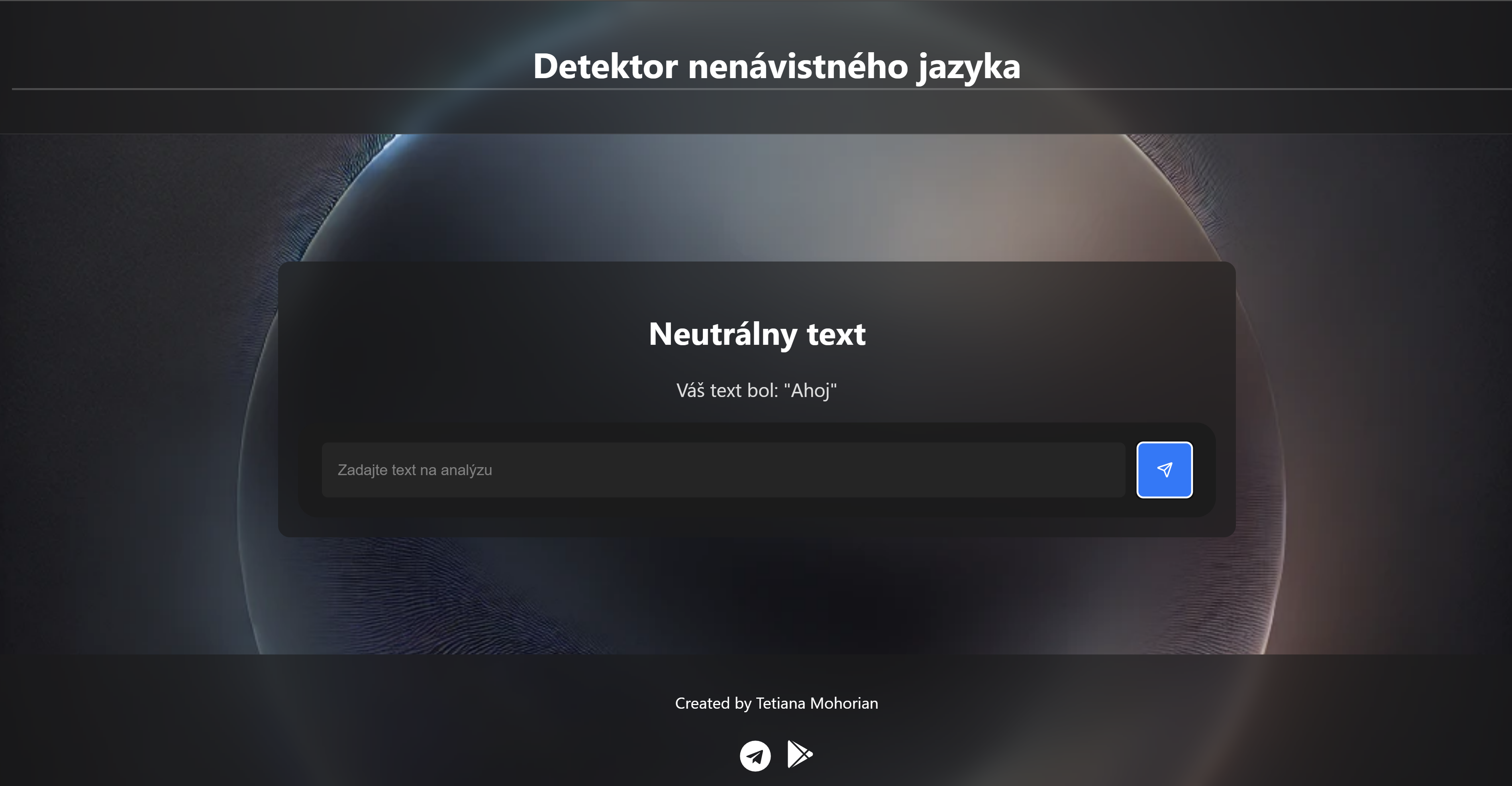
* + - 1. Používateľské rozhranie

Používateľské rozhranie je navrhnuté minimalisticky s dôrazom na jednoduchosť a prehľadnosť. Obsahuje:

* textové pole na zadanie vstupného textu,
* tlačidlo na odoslanie textu na analýzu,
* výstupný blok so zobrazením výsledku klasifikácie.



Obr. 3 Pred odoslaním používateľskej správy



Obr. 4 S výsledkom rozpoznavania.

* + 1. Vývoj Telegram bota

Súčasťou projektu je aj vytvorenie Telegram bota, ktorý v reálnom čase analyzuje správy v skupinovom chate a upozorňuje na výskyt nenávistnej reči. Tento bot slúži ako praktická aplikácia modelu a zároveň dopĺňa webové rozhranie o automatizovaný dohľad nad komunikáciou.

* + - 1. Funkcionalita bota

Bot je implementovaný v jazyku Python s využitím knižnice python-telegram-bot. Je schopný:

* prijímať správy od používateľov v skupinovom chate,
* analyzovať text pomocou trénovaného klasifikačného modelu (transformer),
* rozpoznať, či ide o nenávistnú reč,
* automaticky odstrániť nevhodnú správu,
* upozorniť používateľa, že porušil pravidlá komunikácie.
  + - 1. Integrácia modelu

Model bol trénovaný pomocou transformers a PyTorch a načítava sa pri štarte aplikácie. Pre každý prijatý text bot vykoná tokenizáciu, klasifikáciu a v závislosti od výsledku podnikne potrebné kroky (odstránenie správy, upozornenie atď.).

* + - 1. Práca s databázou

Bot je prepojený s databázou MySQL, kde ukladá údaje o porušovateľoch:

* používateľské meno (username),
* samotnú správu,
* čas odoslania.

Táto funkcionalita umožňuje sledovanie výskytu nevhodných správ a prípadnú analýzu dát. V databáze sa nachádza tabuľka violators, ktorá bola vytvorená nasledujúcim spôsobom:

-- Vytvorenie tabuľky pre ukladanie porušovateľov pravidiel komunikácie

CREATE TABLE violators (

id INT AUTO\_INCREMENT PRIMARY KEY, -- Unikátny identifikátor každého záznamu

username VARCHAR(255), -- Používateľské meno autora správy

message TEXT, -- Obsah správy, ktorá bola klasifikovaná ako nenávistná

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP -- Čas a dátum, kedy bola správa zaznamenaná);

Údaje sa do tabuľky ukladajú pomocou funkcie save\_violator(), ktorá sa vykoná vždy pri detekcii nenávistnej správy.

* + - 1. Bezpečnosť a logovanie

Bot používa .env súbor na bezpečné uloženie tokenu API. Okrem toho zaznamenáva varovania a chyby do log súboru log.txt, čo uľahčuje ladenie a sledovanie prevádzky.

Záver

V tejto bakalárskej práci som sa zaoberala detekciou nenávistnej reči pomocou veľkých jazykových modelov (LLM), pričom som sa sústredila na využitie modelov ako mt5-small, mt5-base, mt5-large, slovak-t5-base, slovak-t5-small, SlovakBERT v few-shot scenároch. Cieľom bolo nielen implementovať tieto modely na našej konkrétnej dátovej množine, ale aj vyhodnotiť ich úspešnosť pomocou štandardných metrík ako Precision, Recall a F1-score. Primárne výsledky ukazujú, že nenávistné prejavy sú hlavným problémom, ktorý má negatívny dopad na spoločnosť, čo vedie k podnecovaniu nenávisti a násilia. Dôležitym problemom je potreba vytvoriť efektívne modely na identifikáciu nenávistných prejavov. Na tento účel sa využívajú rôzne techniky, ako je hlboké učenie a technológie spracovania prirodzeného jazyka. Modely založené na transformátoroch preukázali pôsobivé výsledky v tejto oblasti vďaka presnej detekcii nenávistných prejavov. Okrem toho je pri skúmaní nenávistných prejavov rozhodujúce zvážiť kontext, pretože význam slov sa môže líšiť v závislosti od situácie. To si vyžaduje neustále zlepšovanie algoritmov od výskumníkov a vývojárov systémov, aby sa zaručila presná interpretácia kontextu a znížilo sa množstvo falošných poplachov. Riešenie nenávistných prejavov si preto vyžaduje komplexný prístup, ktorý kombinuje technológiu, medziodborové štúdie a morálne úvahy, aby sa výrazne zlepšil spoločenský blahobyt a podporili sa bezpečnejšie komunikačné prostredia.

Zoznam použitej literatúry

1. Zhang, Aston; Lipton, Zack; Li, Mu; Smola, Alex. Dive into Deep Learning. Kniha. [Online] Amazon Science, 03. 03 2020. [Dátum: 10. 03 2020.] Dostupné na: <https://d2l.ai/index.html>.
2. Phillips, Mark. Dive Deep into Python 3. [Online] Independent Publishing, 2023. [Dátum: 10. 05. 2023.] Dostupné na: <https://diveintopython3.problemsolving.io/>.
3. DOI: 10.1080/08839514.2023.2166719. Swanson, Anthony; Vojtech, Jan. Automated Hate Speech Detection: A Review of Recent Advances. Journal of Interpersonal Violence, 20. 06 2023. [Dátum: 20. 06 2023.] Dostupné na: <https://www.tandfonline.com/doi/full/10.1080/08839514.2023.2166719#d1e579>.
4. DOI: 10.4208/cmes.49631. Li, Wei; Wang, Xiaoming; Zhang, Jing; Yang, Lin. Research on a New Approach to Hate Speech Detection Using Transformer Models. Computer Modeling in Engineering & Sciences, 15. 08 2024. [Dátum: 15. 08 2024.] Dostupné na: <https://cdn.techscience.cn/files/CMES/2024/TSP_CMES-140-3/TSP_CMES_49631/TSP_CMES_49631.pdf>.
5. DOI: 10.1109/ACCESS.2022.9691704. Akbar, Ali; Mahmood, Muhammad; Ali, Mohsin; Shah, Asif. Hate Speech Detection in Multilingual Contexts Using Machine Learning Techniques. IEEE Access, vol. 10, 20. 11 2022. [Dátum: 20. 11 2022.] Dostupné na: <https://ieeexplore.ieee.org/abstract/document/9691704>.
6. DOI: 10.3390/cosmetics5040037. Ameen, Muhammad; Ullah, Ashraf; Qayyum, Anwar; Anjum, Ayesha; Ali, Sheraz. A Study on the Detection of Hate Speech in Social Media Using Deep Learning Techniques. Cosmetics, vol. 5, no. 4, 15. 12 2023. [Dátum: 15. 12 2023.] Dostupné na: <https://www.mdpi.com/2413-4155/5/4/37>.
7. DOI: 10.48255/ceurws.3159. Valdovinos, Armando; Pineda, Eduardo; Castañeda, Enrique; Rodríguez, David. A Comprehensive Review of Hate Speech Detection Approaches in Social Media. CEUR Workshop Proceedings, vol. 3159, 25. 08 2023. [Dátum: 25. 08 2023.] Dostupné na: <https://ceur-ws.org/Vol-3159/T1-39.pdf>.
8. DOI: 10.48550/arXiv.2110.08402. Raffel, Colin; Shazeer, Noam; Roberts, Adam et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5). arXiv, 18. 10 2021. [Dátum: 20. 10 2021.] Dostupné na: <https://arxiv.org/abs/2110.08402>.
9. Hu, Edward J.; Shen, Yelong; Wallis, Shawn et al. LoRA: Low-Rank Adaptation of Large Language Models. arXiv, 24. 02 2022. [Dátum: 26. 02 2022.] Dostupné na: <https://arxiv.org/abs/2203.02245>.
10. lm\_eval\_harness Documentation. [Online] EleutherAI, 2023. [Dátum: 15. 10 2023.] Dostupné na: <https://github.com/EleutherAI/lm-evaluation-harness>.
11. TUKE-KEMT Hate Speech Slovak Dataset. [Online] Hugging Face, 2024. [Dátum: 10. 04 2024.] Dostupné na: <https://huggingface.co/datasets/TUKE-KEMT/hate_speech_slovak>.
12. Hugging Face Transformers Library. [Online] Hugging Face, 2024. [Dátum: 10. 04 2024.] Dostupné na: <https://huggingface.co/docs/transformers/index>.

Prílohy

Príloha A: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe. CD je spravidla grafické s logom univerzity a fakulty. Pozri . Tieto CD robia v Univerzitnej knižnici TUKE.

Príloha B: Používateľská príručka

Príloha C: Systémová príručka

Táto časť diplomovej práce je povinná a obsahuje zoznam všetkých príloh vrátané elektronických nosičov. Názvy príloh v zozname musia byt’ zhodné s názvami uvedenými na príslušných prílohách. Tlačené prílohy majú na prvej strane identifikačné údaje – informácie zhodné s titulnou stranou diplomovej práce doplnené o názov príslušnej prílohy (Systémová príručka, Používateľská príručka). Identifikačné údaje sú aj na priložených diskoch alebo disketách. Ak je médií viac, sú označené aj číselne v tvare I/N, kde I je poradové číslo a N je celkový počet daných médií.

Každá príloha začína na novej strane a je označená samostatným písmenom (Príloha A, Príloha B, ...). Číslovanie strán príloh nadväzuje na číslovanie strán v hlavnom texte.



Obr. 5 Obrázok grafického CD média