VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATIZOVANÁ DETEKCE OFENZIVNÍHO JAZYKA A NENÁVISTNÝCH PROJEVŮ V PŘIROZENÉM JAZYCE

AUTOMATED DETECTION OF HATE SPEECH AND OFFENSIVE LANGUAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ALŽBETA ŠTAJEROVÁ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MARTIN FAJČÍK**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Studentka: **Štajerová Alžbeta**
Program: Informační technologie
Název: **Automatizovaná detekce ofenzivního jazyka a nenávistných projevů v přirozeném jazyce**
Automated Detection of Hate Speech and Offensive Language
Kategorie: Zpracování řeči a přirozeného jazyka

Zadání:

1. Popište fenomén ofenzivního jazyka a nenávistných projevů v přirozeném jazyce.
2. Popište současné metody pro řešení problému detekce ofenzivního jazyka a nenávistných projevů v přirozeném jazyce.
3. Popište datové sady vhodné pro trénování modelu.
4. Vyberte vhodnou metodu pro řešení problému.
5. Popište metody vhodné pro vyhodnocení navrženého řešení.
6. Implementuje vybrané metody.
7. Vyhodnoťte implementované řešení.
8. Vytvořte ablační studii nad implementovaným řešením.
9. Porovnejte dosažené výsledky se současnými systémy pro detekci ofenzivního jazyka a nenávistných projevů v přirozeném jazyce.

Literatura:

- T. Davidson, D. Warmsley, M. W. Macy, I. Weber, "Automated hate speech detection and the problem of offensive language", *Proc. 11th Int. Conf. Web Social Media*, pp. 512-515, 2017.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 5 ze zadání

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Fajčík Martin, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 5. listopadu 2018

Abstrakt

Táto práca sa zaoberá fenoménom nenávistných prejavov a ofenzívneho jazyka, ich definíciami a detekciou. Popisuje metódy doterajšieho riešenia detekcie. Zhodnocuje dostupné dátové sady využiteľné pri trénovaní modelov zameraných na detekciu tohto fenoménu. Dáva si za cieľ uviesť ďalšie metódy riešenia detekcie tohto problému a porovnanie ich výsledkov a vyhodnotenie úspešnosti. Zvolený problém bol riešený piatimi modelmi. Dva z nich boli zamerané na extrakciu príznakov a ich následnú klasifikáciu. Ďalšie tri boli riešené pomocou neurónových sietí. Úspešnosť implementovaných modelov som experimentálne vyhodnotila. Výsledky tejto práce umožňujú porovnanie typických prístupov s metódami využívajúcimi najnovšie poznatky z oblasti strojového učenia použitých pre klasifikáciu nenávistného a ofenzívneho jazyka.

Abstract

This thesis discusses hate speech and offensive language phenomenon, their respective definitions and their occurrence in natural language. It describes previously used methods of solving the detection. An evaluation of available data sets suitable for the problem of detection is provided. The thesis aims to provide additional methods of solving the detection of this issue and it compares the results of these methods. Five models were selected in total. Two of them are focused on feature extraction and the remaining three are neural network models. I have experimentally evaluated the success of the implemented models. The results of this thesis allow for comparison of the typical approaches with the methods leveraging the newest findings in terms of machine learning that are used for the classification of hate speech and offensive language.

Kľúčové slová

spracovanie prirodzeného jazyka, ofenzívny jazyk, nenávistný prejav, klasifikácia, strojové učenie, detekcia, spracovanie textu

Keywords

natural language processing, offensive language, hate speech, classification, machine learning, detection, text processing

Citácia

ŠTAJEROVÁ, Alžbeta. *Automatizovaná detekce ofenzivního jazyka a nenávistných projevů v přirozeném jazyce*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Fajčík

Automatizovaná detekce ofenzivního jazyka a nenávistných projevů v přirozeném jazyce

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením inžiniera Martina Fajčíka. Uviedla som všetky literárne zdroje a publikácie, z ktorých som čerpala.

.....
Alžbeta Štajerová
15. mája 2019

Poďakovanie

Ďakujem vedúcemu mojej bakalárskej práce Ing. Martinovi Fajčíkovi za odbornú pomoc a vedenie pri vypracovaní tejto práce.

Obsah

Kapitola 1

Úvod

Internet môže slúžiť ako výborné miesto pre vyjadrovanie svojich názorov a myšlienok. S pribúdajúcim počtom užívateľov sociálnych sietí sa často stáva internet svetom nenávisti. Preto sa rozpoznávanie ofenzívneho jazyka stáva relevantnejšie. Kladie sa čoraz väčší dôraz na dodržiavanie pravidiel spojených s používaním nenávistných prejavov. Sociálne siete neposkytujú dostatočnú ochranu pred týmto fenoménom a zároveň nie sú schopné ho okamžite rozpoznávať. Odhaľovanie prípadov používania nenávistného prejavu, je náročné vyhodnocovať pri súčasnom počte požívateľov internetu ľudskými zdrojmi. Preto sa skúmajú vhodné prístupy, ktoré by tento proces strojovo automatizovali.

Súčasné riešenia nie sú dostačujúce, pretože niektorí užívatelia úmyselne maskujú nenávistný prejav, napríklad vytváraním preklepov. Pre odhalenie používania ofenzívneho jazyka je nutné, aby metóda správne pochopila celý kontext príspevku a tak predišla nesprávnej predikcii. Preto s ohľadom na súčasný stav poznania sú dnešné prístupy založené na strojovom učení.

V mojej práci sa zaoberám detekciou ofenzívneho jazyka a nenávistných prejavov. Popisujem metódy zvolené za účelom detekcie tohto fenoménu. Zhromaždila som a porovnávam dostupné dátové sady využiteľné pri trénovaní týchto modelov. Venujem sa aj ďalším metódam vhodných pre klasifikáciu ofenzívneho jazyka.

Táto práca je členená do siedmych kapitol. V kapitole 2 sa zaoberám definíciou ofenzívneho jazyka a nenávistného prejavu. Opis súčasných metód na detekciu tohto fenoménu sa nachádza v kapitole 3. Zozbierané dátové sady vyhodnocujem v kapitole 4. Popis navrhovaných riešení, ktoré navrhujem je v kapitole 5. V kapitole 6 sa venujem implementácii, popisu použitých technológií, experimentom a vyhodnoteniu jednotlivých modelov. Výsledky mojej práce zhodnocujem v kapitole 6.4.5.

Kapitola 2

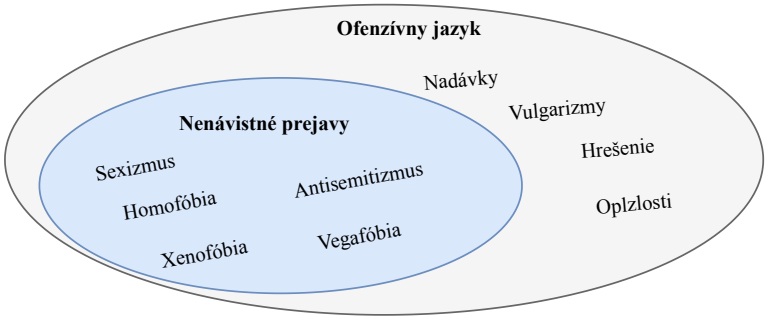
Ofenzívny jazyk a nenávistné prejavy v prirodzenom jazyku

Kapitola sa venuje popisu pojmov *ofenzívny jazyk* (OL) a *nenávistný prejav* (HS). Rieši sa tu problematika jednoznačnej definície a zaradenia.

2.1 Ofenzívny jazyk

Ofenzívny jazyk je reprezentovaný ustálenými slovnými spojeniami využívanými k vyjadreniu emočného postoja (frustrácia, hnev), negatívneho, zlého, vulgárneho používania [27]. Do tejto kategórie spadajú nadávky, neslušné slová, vulgarizmy, rúhanie, hrešenie, oplzlosti, ale aj samotný nenávistný prejav. Používanie ofenzívneho jazyka je považované za nezdvorilé, hrubé a urážlivé. Ofenzívny jazyk sa stále vyvíja a jeho používanie sa rozširuje a stáva sa súčasťou určitých kultúr [16]. Avšak využívanie týchto jazykových prostriedkov nie je vhodné vo všetkých sférach a postupne sa upravuje legislatíva, ktorá obmedzuje prípady ich využívania. V niektorých krajinách je používanie ofenzívneho jazyka trestné. Jedná sa napríklad o Kanadu, kde vyrušovanie spôsobené nadávkami, urážkami a používaním obscénneho jazyka na verejnosti je považované za trestný čin, vo verejných parkoch je nadávanie zakázané [36]. Na Novom Zélande sú oplzlosti a neslušné slová nedovolené vrámci všetkých verejných priestranstiev, v tomto prípade sa ukladá pokuta. North Carolina, štát v Spojených štátoch amerických, musela zákon ohľadom používania nadávok a ďalších neslušných slov, ktoré boli počuté dvomi a viac ľudmi zrušiť z dôvodu jeho protiústavnosti [1].

Mnoho slov a fráz spadá do tejto kategórie, mnohé si zakladajú na sémantike využitia (názvy zvierat, etnické a rasové označenia, nechutné predmety, či slang). Môžu byť len mierne urážlivé, ale aj neprimerane urážajúce. Ďalšími faktormi ovplyvňujúcimi ofenzívny jazyk okrem kontextu sú sociálne prostredie, tón hlasu, či vzťah medzi rečníkom a adresátom, pohlavie a vek [16].



Obr. 2.1: Ofenzívny jazyk v súvislosti s nenávistnými prejavmi

2.2 Nenávistný prejav

Formálne definície pojmu nenávistného prejavu sa nezhodujú. Tento fenomén zahŕňa osobné útoky, nevraživosť, pohŕdanie, ponižovanie, zastrašovanie či agresiu voči špecifickej skupine ľudí alebo menšine [20]. Ďalej tu môžu byť obsiahnuté podnety vyvolávajúce násilie a vyhrážky. Skupinu môžu špecifikovať rôzne vlastnosti ako napríklad etnický pôvod, náboženstvo, pohlavie, sexuálna orientácia, hendikep, vek, farba pleti, životný štýl alebo iná analogická charakteristika. Medzi známejšie nenávistné prejavy sa radí antisemitizmus, sexizmus, xenofóbia, homofóbia a islamofóbia.

Zákony mnohých štátov zakazujú každý prejav nenávisti mierený na menšiny. Porušením môže človek čeliť pokute, v najhorších prípadoch porušenia aj odňatím slobody [7]. Zákony Slovenskej republiky trestá názory hanobujúce národ, rasu alebo etnickú skupinu odňatím slobody. Rovnako to je aj pre podnecovanie nenávisti voči jednotlivcovi alebo skupine osôb. Anglicko , Ukrajina, Švédsko, Španielsko, Rumunsko, Dánsko i ďalšie zakazujú požívanie nenávistného jazyka a postihujú ho pokutou alebo až odňatím slobody na 30 dní až 10 rokov. Naopak legislatíva Spojených štátov amerických je kritizovaná za nedostatočnú ochranu pred nenávistným jazykom. Súdy niekoľko násobne rozhodli, že takéto zákony porušujú slobodu prejavu [30].

Vegafóbia je nenávistný jazyk mierený na vegetariánov a vegánov

Kapitola 3

Základné koncepty strojového učenia

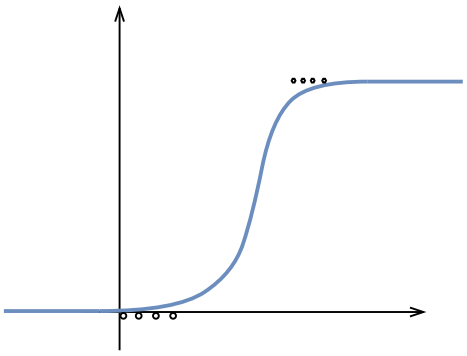
Zaoberá sa metódami, ktoré umožňujú adekvátnu reakciu na rôzne vstupy. Rozhodovanie je zabezpečené bez uvedenia explicitných pokynov. Tieto metódy sú založené na prvkoch matematickej štatistiky a analýzy dát.

Väčšina modelov použitých za posledné roky v oblasti detekcie nenávistného jazyka sa sústreďuje na jeho klasifikáciu. Znamená to, že model rozhodne, ktorá z tried prislúcha danému príkladu. Výstupom tejto úlohy je zvyčajne pravdepodobnostná hodnota prislúchajúceho vstupu do danej triedy. Prevládajú metódy založené na učení s učiteľom nad metódami učenia bez učiteľa.

3.1 Doterajšie metódy riešenia

3.1.1 Logistická regresia

Metóda logistickej regresie (viď obrázok 3.1) označuje problematiku výpočtu pravdepodobnosti $P(Y|X)$ javu Y na základe určitých príznakov. Model vyjadruje pravdepodobnosť výstupu v závislosti na vstupoch, ale nevykonáva klasifikáciu. Úprava na binárny klasifikátor si zakladá na výbere hraničnej hodnoty. Pravdepodobnostná hodnota pod hranicou určuje jednu triedu a hodnota nad hranicou druhú triedu. Prípady kedy logistická regresia môže nadobúdať výstup dvoch typov (dve triedy) označuje termín binárna logistická regresia. Zvyčajne je výsledok reprezentovaný ako “0” a “1”.

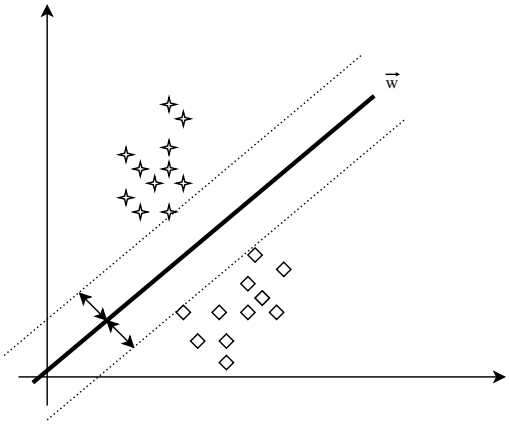


Obr. 3.1: Logistická regresia

V súvislosti s detekciou nenávistného jazyka bola táto metóda využitá s príznakmi, kedy sa vytvorili zo slov v texte n-gramy (bigramy, unigramy a trigramy), ktoré sa ohodnotili TF-IDF. Ďalším príznakom boli testy čitateľnosti, sentiment textu a Part-Of-Speech tagy [2, 7, 4].

3.1.2 Support vector machines

Support vector machines (SVM) je metóda klasifikácie hľadajúca hyper-rovinu, ktorá by rozdeľovala priestor na čo najoptimálnejšie priestory daných tried. Je podobná logistickej regresie. Optimálna nadrovina maximalizuje vzdialenosť medzi triedami. Tento priestor sa označuje ako hraničné pásmo. Obrázok 3.2 ilustruje hraničné pásmo v lineárnom priestore. Existuje rozšírenie SVM, ktoré dokáže rozdelenie tried v prípade lineárne neseparovateľných dát s použitím kernelových funkcií.

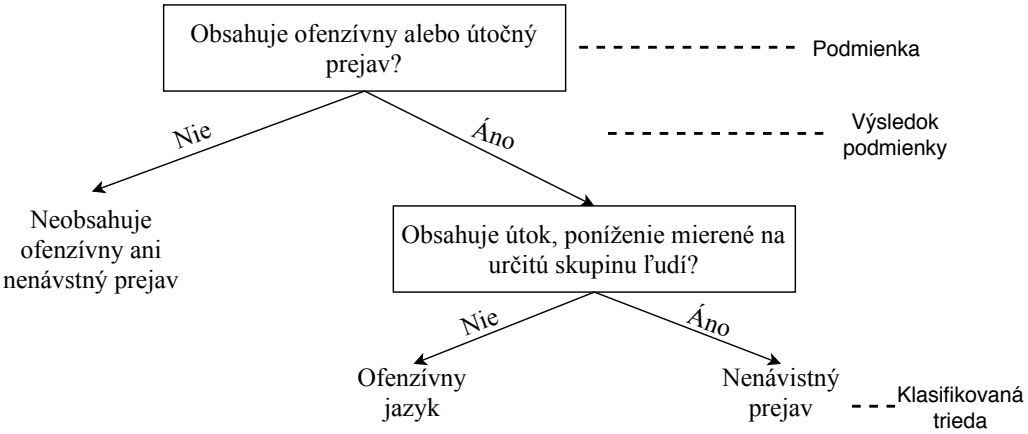


Obr. 3.2: Optimálna rozdeľovacia nadrovina

3.1.3 Rozhodovacie stromy

Táto metóda na základe vstupných hodnôt príznakov, podmienkových atribútov, klasifikuje objekty. Model vytvára stromovú štruktúru rozhodnutí (viď obrázok 3.3). Uzly predstavujú vyhodnotenie vlastností, vetvy výsledok vyhodnotenia a listy stromu reprezentujú predikovanú triedu. Výhodou tejto metódy je ľahká interpretovateľnosť z dôvodu pevných

rozhodnutí (podmienok). Princíp metódy pozostáva v nájdení takého stromu rozhodnutí, ktorý najlepšie predikuje triedy príkladov.



Obr. 3.3: Jednoduchý rozhodovací strom pre klasifikáciu ofenzívneho jazyka a nenávistných prejavov

3.1.4 Náhodný les

Pri tomto postupe je vytváraných niekoľko rozhodovacích stromov miesto spoliehania sa len na jeden strom rozhodnutí. Následná klasifikácia prebieha vyhodnotením stromov a určení najčastejšej predikcie. Oproti rozhodovacím stromom samotným tato metóda lepšie predchádza preučeniu na tréningovej sade.

3.1.5 Naive Bayes

Naive Bayes je pravdepodobnostná metóda vychádzajúca z Bayesovho teorému (viď rovnica 3.1), ktorý vyjadruje pravdepodobnosť udalosti A podmienenú udalosťou B.

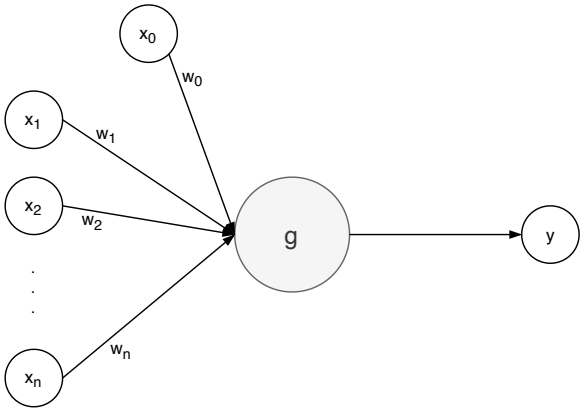
$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \tag{3.1}$$

3.1.6 Neurónové siete

Neurónová sieť, ako výpočtový model, bola zostavená ako abstrakcia ľudského mozgu. Modely sú zložené z umelých neurónov, perceptronov. Ich koncept vychádza z funkcie a tvaru neurónovej bunky. Biologické neuróny dokážu prenášať signál ďalším neurónom. Telo bunky sa nazýva soma, z ktorého vystupujú výbežky. Tieto výbežky tvoria rozhrania medzi bunkami. Dendridy, vstupné výbežky, signál príjmajú a neurity ho z bunky ďalej odvádzajú.

Analogicky fungujú aj umelé neuróny (viď obrázok 3.4), ktorých fungovanie je zjednodušené. Premenné $x_1, \ldots, x_n$ reprezentujú vstupné hodnoty a váhy pre vstupy $w_1, \ldots, w_n$. Špeciálny vstup *bias* predstavuje x_0 a w_0. Táto hodnota ovplyvňuje aktivačnú funkciu neurónu. Potom sa vnútorný stav reprezentuje ako $w_0x_0 + \sum_{i=1}^{n} w_ix_i$. Tento vzťah je daný ako suma váh a vstupov so zarátaním hodnoty bias a jej váhy.

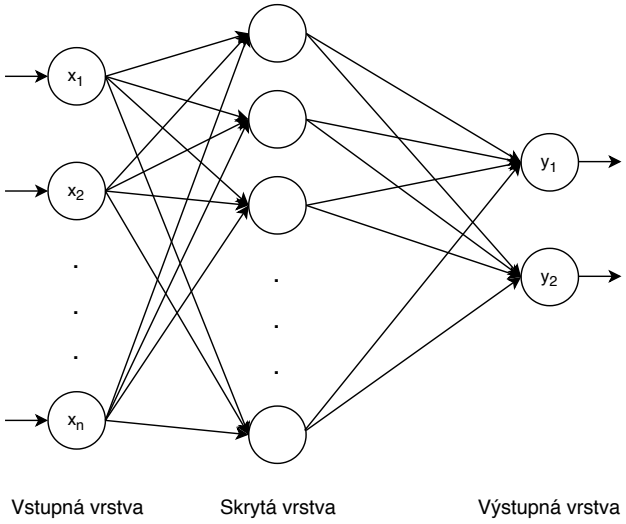
Sieť predstavuje orientovaný graf (viď obrázok 3.5), kde uzly sú neuróny a hrany majú váhy. Jednotlivé neuróny sú usporiadané do vrstiev. Neuróny v danej vrstve sú zvyčajne pre-



Obr. 3.4: Matematický model neurónu

pojené s neurónmi z predchádzajúcej vrstvy. Prvú vrstvu tvoria vstupné hodnoty. Poslednú vrstvu tvoria výstupné hodnoty. Ostatné vrstvy sa označujú ako *skryté vrstvy.*

Architektúr pre neurónové siete je mnoho. Signál sa v nich nemusí šíriť len dopredným smerom. V sieti môžu existovať aj prepojenia na spätné neuróny prípadne na celé vrstvy. Tieto siete sa označujú ako rekurzívne neurónové siete. V prípade, že sa spracováva predošlý výstup, jedná sa o rekurzívne neurónové siete.



Obr. 3.5: Neurónová sieť s jednou skrytou vrstvou

Pri trénovaní neurónovej siete v prípade klasifikácie sa využíva dvojica hodnôt x_i pre vstup a y_i pre očakávaný výstup. Postupným trénovaním sa váhy upravujú tak, aby sa výstup približoval viac výstupu ako predošle.

Aktivačné funkcie

Súčasťou vrstiev v neurónových sieťach bývajú aj aktivačné funkcie. Slúžia na prahovanie hodnôt. V začiatkoch sa využívala ostrá nelinearita, ktorú teraz nahradzujú hladké (okrem ReLU) funkcie.

Ostrá nelinearita Skoková aktivačná funkcia ostrá nelinearita má obor hodnôt $\{0, 1\}$.

$$f(x) = \begin{cases} 1 & \text{pre } x > 0.5 \\ 0 & \text{pre } x \leq 0.5 \end{cases} \tag{3.2}$$

Sigmoida Výstupom *sigmoidy* (viď rovnica 3.3) sú hodnoty v intervale $(0, 1)$.

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

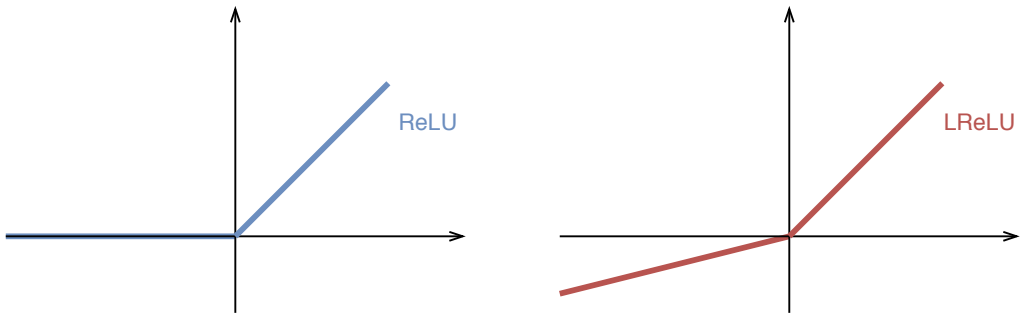
Tanh Tanh, hyperbolický tangens (viď rovnica 3.4) transformuje hodnoty náležiac intervalu $(-1, 1)$.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.4}$$

ReLU ReLU (angl. *rectified linear unit*) aktivačná funkcia nadobúda hodnoty z intervalu $< 0, \infty)$ (viď obrázok 3.6a). ReLu (viď rovnica 3.5) vyjadruje $\max(0, x)$. Obmenami pre túto funkciu je *Leaky ReLUs* (viď rovnica 3.6), ktorá umožňuje malý gradient v neaktivovaných hodnotách (viď obrázok 3.6b).

$$f(x) = \begin{cases} x & \text{pre } x \geq 0 \\ 0 & \text{pre } x < 0 \end{cases} \tag{3.5}$$

$$f(x) = \begin{cases} x & \text{pre } x \geq 0 \\ 0.01x & \text{pre } x < 0 \end{cases} \tag{3.6}$$



(a) ReLU (b) Leaky ReLU

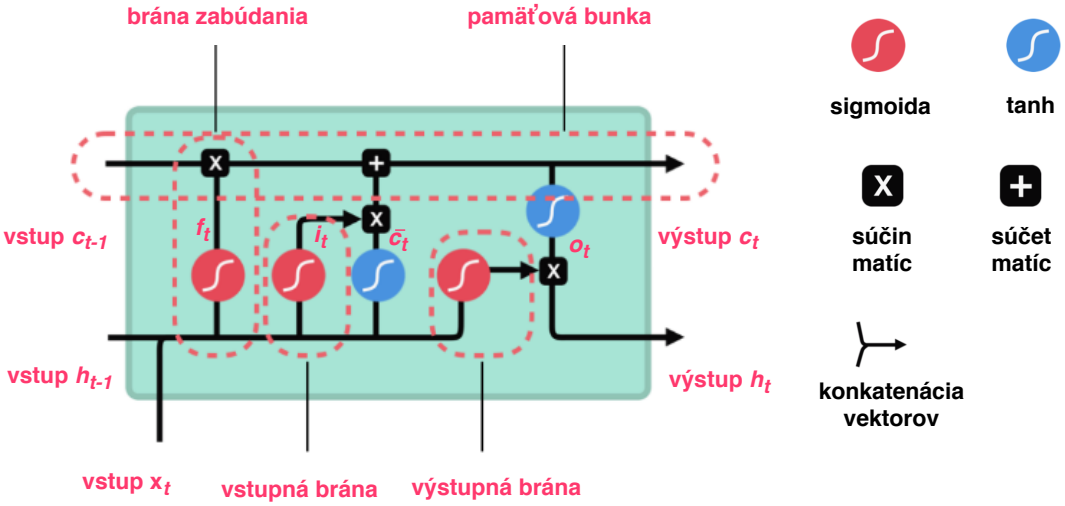
Obr. 3.6: Grafy funkcií ReLU a Leaky ReLU

Softmax Softmax (viď rovnica 3.7), je nelineárna funkcia. Po aplikovaní tejto funkcie hodnoty spadajú do intervalu $(0, 1)$ a po sčítaní všetkých hodnôt výsledok predstavuje hodnotu 1.

$$f(x)_i = \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}} \tag{3.7}$$

3.1.7 LSTM

LSTM (angl. *long short-term memory*) [14] je rekurentnou neurónovou sieťou (vid obrázok 3.7). Využitie LSTM je vhodné pri pochopení dlhších kontextov. Je schopné nájsť závislosti jednotlivými elementami vstupu. Koncept fungovania LSTM pozostáva z pamäťovej bunky (angl. *cell*), brány zabúdania (angl. *forget gate*) a vstupnej brány (angl. *input gate*) a výstupnej brány (angl. *output gate*).



Obr. 3.7: Schéma LSTM [25]

Pamäťová bunka LSTM siete si udržiava povedomie o toku informácii po dlhšiu dobu. Prenáša relatívne informácie. Brány ďalej rozhodujú aké informácie sú dôležité pre udržanie alebo zabudnutie. Brána zabúdania určuje, ktoré dáta z predošlého spracovania h_{t-1} sú relevantné, a ktoré sa zahodia (viď rovnica 3.8). Vstupná brána upravuje pamäťovú bunku (viď rovnica 3.9). Určí, ktoré dáta pamäte sa budú upravovať pre výstup hodnoty c_t. Výstupná brána ďalej určuje, aké dáta sa posunú ďalej pre nasledujúci stav h_t (viď rovnice 3.12). Aktivačné funkcie sú tiež dôležité pre fungovanie LSTM. Sieť sa učí, ktoré hodnoty môžu byť zabudnuté a aké sú dôležité ponechať. Výstupom LSTM je dvojica c_t a h_t.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.8}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.9}$$

$$\bar{c_t} = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3.10}$$

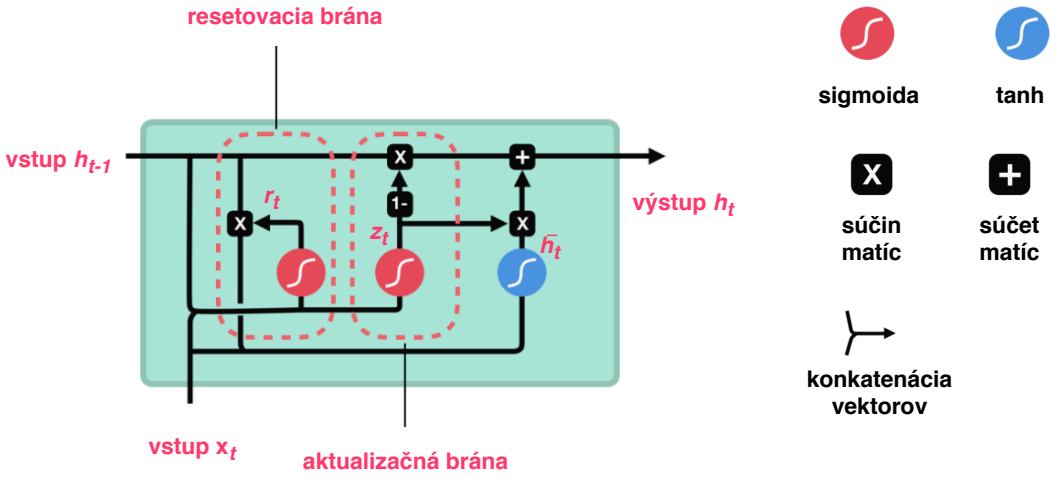
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \bar{c_t} \tag{3.11}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.12a}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{3.12b}$$

3.1.8 GRU

GRU (angl. *gated recurent unit*) [6] je rekurentná neurónová sieť (viď obrázok 3.8). Obsahuje aktualizačnú bránu z_t (angl. *update gate*) a resetovaciu bránu r_t (angl. *reset gate*). Vstupom je podobne ako pri LSTM hodnota z predchádzajúceho spracovania h_{t-1} a nový spracovávaný vstup x_t. Funkcia aktualizačnej brány je podobná bráne zabúdania a vstupnej bráne pri LSTM. Pozostáva v rozhodovaní, ktoré informácie ponechať a aké informácie je potrebné pridať, a to na základe váh W_z (viď rovnica 3.14). Resetovacia brána určuje, ako a ktoré z minulých informácii sa zabudne na základe váh W_r (viď rovnica 3.14). Výstupom je h_t (viď rovnice 3.15).



Obr. 3.8: Schéma GRU [25]

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{3.13}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3.14}$$

$$\bar{h}_t = \tanh(W \cdot [r_t \cdot h_{t-1}, x_t] + b_c) \tag{3.15a}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \bar{h}_t \tag{3.15b}$$

3.2 Predspracovanie textu

Cieľom predspracovania textu je odstránenie redundantných dát a rušivých informácií. Využíva sa pri tom hlavne odstraňovanie alebo nahradzovanie.

3.2.1 Odstraňovanie stop slov

Stop slová sú slová vyskytujúce sa v jazyku často, ale samy o sebe nemajú významovú informáciu. Spĺňajú syntaktický význam. V slovenčine sa jedná napríklad o spojky (a, aj, i, ...), predložky (pred, za, o, ...), častice a pod.

3.2.2 Lemmatizácia

Lemmatizácia je proces spracovania textu, kedy sa slovo, ktoré nie je vo svojom základe nahradí základom slova (angl. *lemma*). Jedná sa napríklad o zámenu slova *sú* alebo *je* za základ slova *byť*.

3.2.3 Stematizácia

Proces spracovania textu, kedy sa slová nahradzujú časťou slova, ktorá je súčasťou rôznych tvarov tohto slova, je stematizácia. To znamená, že sa odstránia jeho prípony, ale slovo sa nenahradzuje nutne jeho základom. Slovo *nakupujúc* sa nahradí len jeho časťou *nakupuj*. Rovnako by sa nahradili slovo *nakupujúci*.

3.2.4 Tokenizácia

Rozdeľovanie spracovávaného vstupu na definované časti (angl. *tokens*), sa označuje ako tokenizácia. Tokeny môžu predstavovať znaky, slabiky, slová alebo vety.

3.2.5 Rozdelovanie viet

Vety môžu obsahovať interpunkčné znamienka, ktoré nutne nerozdeľujú vetu. Rozdelovanie viet dokumentu je tokenizácia dokumentu na vety. Pri tokenizácii dokumentu *"Dobehol som 1. v poradí."* by mohol byť teda neprávne rozdelený na dve vety vzhľadom na bodku, avšak prvé interpunkčný znak v tomto kontexte vyjadruje radovú číslovku. Správne sa jedná len o dokument s jednou vetou.

3.3 Využívané príznaky

Príznaky sú dôležitou časťou pri rozpoznávaní vzorov a súčasťou strojového učenia ako takého. Vyjadrujú určitú vlastnosť, charakteristiku [3]. Zvolenie tých najinformatívnejších je kľúčovým krokom pre kvalitné výsledky. Miera závislosti medzi skúmaným javom (HS/OF) a pozorovanom jave, ktorý udáva príznak, určuje informatívnosť. Zvyčajne sa jedná o numerické príznaky. Pri klasifikačných úlohách sa spravidla nepoužíva len jeden príznak, ale niekoľko, jedná sa o vektor príznakov (angl. *feature vector*).

Konkrétnymi príznakmi v úlohách spracovania prirodzeného jazyka môžu byť prítomnosť a neprítomnosť špecifického výrazu, počty výskytov termínu. Proces získavania príznakov prebieha v predspracovaní a označuje sa ako extrakcia príznakov.

3.3.1 Bag of words

Bag of words (BoW) je spôsob reprezentácie textu pre metódy strojového učenia z dôvodu neschopnosti spracovať text ako taký. Text sa prekonvertuje ďalej opísaným spôsobom na číselný vektor.

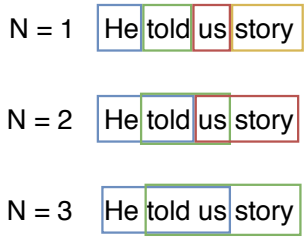
Pre získanie tohto príznaku je potrebné poznať slovník známych slov. Tvorba slovníka je kľúčovým krokom pre tento príznak. V závislosti od množstva dostupných dát a riešeného problému sa rieši predspracovanie textu a to nerozlišovaním medzi veľkými a malými písmenami, vynechaním interpunkčných znamienok a slov s malou informačnou hodnotou (spojky, predložky, zámená, niektoré slovesá) alebo sa slová predspracujú lematizáciou a stematizáciou. Ďalšou úpravou môže byť n-gramový prístup (viď kapitola 3.3.2). Následne

sa pre každé slovo zo slovníka vypočíta skóre na základe obsahu dokumentu. Skóre môže predstavovať frekvenciu tohto slova v dokumente, značku prítomnosti (0 pre neobsahuje, 1 pre obsahuje) alebo TF-IDF hodnoty. Výsledkom je vektor s rozmerom dĺžky ako je veľkosť slovníka. Pozostáva z prevažne nulových hodnôt (riedky vektor). Ak je slovník veľký, jedná sa o veľmi pamäťovo aj výpočtovo náročný príznak.

Avšak takáto reprezentácia textu stráca syntaktickú (poradie) aj sémantickú (kontext) informáciu. Príznak sa zaoberá len obsahom, či sa známe slová zo slovníka nachádzajú v spracovávanom príklade. Z tohto dôvodu je tento typ príznaku nazývaný bag-of-words. Idea vychádza z predpokladu, že podobné dokumenty budú obsahovať podobný obsah.

3.3.2 N-gram

N-gramy vyjadrujú sekvenciu za sebou pokračujúcich n elementov. Týmito elementami nemusia byť len slová, ale aj slabiky, či písmená. N-gramy pozostávajúce z jedného elementu sa nazývajú *unigramy*, z dvoch elementov *bigramy*, z troch *trigramy* [17].



Obr. 3.9: Unigramy, bigramy, trigramy

3.3.3 TF-IDF

Príznaky TF-IDF (viď rovnice 3.16), skratka pre *term frequency vs. inverse document frequency*, sa snaží riešiť problém s menej a viac častými slovami a ich informatívnou hodnotou v porovnaní s tými menej častými, úlohovo špecifickými slovami. Tento prístup je vhodný využiť pri riešeniach problémov, kedy sa využívajú doménovo špecifický slovník slov.

$$TF = \textit{frekvencia termínu v dokumente} \tag{3.16a}$$

$$IDF = \log(\frac{\textit{počet spracovávaných dokumentov}}{\textit{počet dokumentov, ktoré obsahujú termín}}) \tag{3.16b}$$

$$TF\text{-}IDF = TF * IDF \tag{3.16c}$$

V literatúrach sú bežné úpravy pre tento príznak. Môžu predstavovať zmeny v spôsoboch vyhodnocovania hodnôt TF (viď rovnica 3.16a) a IDF (viď rovnica 3.16b). Pre IDF úprava môže znamenať nahradenie pomeru len počtom všetkých dokumentov. TF je možné pozmeniť:

- značka prítomnosti (0 pre neobsahuje, 1 pre obsahuje),
- frekvencia termínu prispôsobená dĺžke dokumentu v pomere,
- logaritmicky upravená hodnota TF $\log(1 + TF)$[21].

3.3.4 Sentiment

Sentiment dokumentu vyjadruje polaritu a teda či je daný dokument pozitívny, neutrálny alebo negatívny.

3.3.5 Slovné druhy (POS)

Slovné druhy pridávajú predstavu o kontexte využitých slov vo vete. Je možné ich využiť aj pri prístupe Bag-of-Words, kedy sa slová nahradia ich náležiacim slovným druhom.

3.3.6 Rozpoznanie pomenovanej entity (NER)

Rozpoznanie pomenovanej entity (angl. *Named-entity recognition*) je proces extrakcie príznakov, kedy sa snaží klasifikovať mená osôb, organizácie, miesta a ďalšie. Pre vetu *"Pracuje ako učiteľ v Dolnom Kubíne."* je takouto entitou Dolný Kubín.

3.3.7 Testy čitateľnosti

Čitateľnosť vyjadruje jednoduchosť, s akou je možné konkrétny dokument pochopiť.

Flesch–Kincaid testy čitateľnosti Mieru zložitosti pochopenia anglického textu vyjadrujú Flesch–Kincaid testy čitateľnosti. Pozostávajú z dvoch testov: *Flesch Reading Ease* (FRES), *Flesch–Kincaid Grade Level*. Výsledné hodnoty sú prispôsobené americkému školskému systému.

Flesch Reading Ease vyhodnocuje dokumenty s vyššími hodnotami ako viac čitateľnejšie ako tie s nižšími [19].

$$FRES = 206.835 - 1.015 * \frac{počet\ slov}{počet\ viet} - 84.6 * \frac{počet\ slabík}{počet\ slov} \tag{3.17}$$

Flesch–Kincaid Grade Level vyjadruje, koľko rokov vzdelania by mal mať človek za sebou, aby pochopil dokument [19]. Využíva sa pri odporúčaní povinného čítania v školách.

$$grade\ level = 0.39 * (\frac{počet\ slov}{počet\ viet}) + 11.8 * (\frac{počet\ slabík}{počet\ slov}) + 15.59 \tag{3.18}$$

3.3.8 Vektorové reprezentácie slov

Word embedding označuje mapovanie slov na vektory reálnych čísel. Vektory predstavujú reprezentáciu slov vo vektorovom priestore. Distribučná hypotéza tvrdí, že slová vyskytujúce sa v rovnakých kontextoch majú tendenciu podobného významu [12]. Slovo je charakterizované spoločnosťou, ktorú si okolo seba drží [9]. Táto reprezentácia umožňuje dobrý sémantický aj kontextový význam. Slová s podobnými vlastnosťami sa v tomto priestore nachádzajú blízko seba. Medzi najvyužívanejšie slovné vektory sa radí GloVe, word2vec, fastText, ELMo.

Word2vec označuje prístupy CBOW (Continuous Bag-Of-Words) alebo Skip-gram. CBOW metóda je vychádza zo snaženia sa predikovať slovo na základe kontextu. Skip-gram metóda má opačný postup. Na základe slova sa pokúša predikovať kontext.

3.4 Vyhodnocovanie výsledkov modelov

Úspešnosť metódy je možné zhodnotiť na základe rôznych metrík. Pre klasifikačné úlohy mnohé z týchto metrík vyhodnocujúce výsledky, vychádzajú z hodnôt uvedených z matice zámen, jej alternatívne názvy sú konfúzna alebo chybová matice.

3.4.1 Presnosť (accuracy)

Počet správnych predpovedí k celkovému počtu vstupov v pomere vyjadruje presnosť (viď rovnica 3.19). Nevýhodným prípadom použitia tejto metriky sa javí vyhodnocovanie pri nevyrovnanej početnosti tried. Táto metrika je menej citlivá na nesprávnu klasifikáciu triedy s menším počtom prvkov [24]. Zmena tejto metriky závisí proporcionálne od počtu dát.

$$Acc = \frac{počet\ správnych\ predikcií}{celkový\ počet\ predikcií} \tag{3.19}$$

3.4.2 Konfúzna matica

Táto matica je používaná pre analýzu výsledkov klasifikácie. Používa sa na vypočítanie ďalších zložitejších metrík opísaných ďalej. Riadky matice reprezentujú skutočné triedy a stĺpce predstavujú predpokladané triedy inštancií, poprípade naopak.

Tabuľka 3.1 znázorňuje konfúznu maticu dvoch tried. Nech negatívne príklady sú tie anotované 0 a pozitívne príklady s hodnotou anotácie 1. **True Negative** (TN), vyjadruje počet negatívnych príkladov, ktoré metóda určila správne. Obdobnou kategóriou pre správne vyhodnotené pozitívne príklady je **True Positive** (TP). Pozitívne príklady, ktoré boli vyhodnotené nesprávne ako negatívne príklady označuje kategória **False Negative** (FN) a počet negatívnych príkladov nesprávne určených ako pozitívne príklady udáva kategória **False Positive** (FP). Hlavná diagonála matice v tomto prípade určuje správne predikcie a vedľajšia diagonála nesprávne.

		Trieda predikcie	
		Positive	**Negative**
Skutočná trieda	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Tabuľka 3.1: Konfúzna matica dvoch tried - *Positive* a *Negative*

Presnosť (precision)

Presnosť (angl. *precision*), inak aj pozitívna prediktívna hodnota, vyplývajúca z konfúznej matice značí pravdepodobnosť správneho vyhodnotenia triedy prvkov (viď rovnica 3.20 vyjadruje tento vzťah pre prvky pozitívnej triedy).

$$precision = \frac{TP}{FP + TP} \tag{3.20}$$

Senzitivita (recall)

Pomer (viď rovnica 3.21) prvkov triedy a prvkov skutočne označených ako daná trieda vyjadruje senzitivitu (angl. *recall*). Určuje mieru schopnosti správne rozlíšiť jednotlivé triedy. Najvyššia hodnota by značila bezchybné rozpoznávanie a teda nedochádza k zlému označeniu tried.

$$recall = \frac{TP}{FN + TP} \tag{3.21}$$

3.4.3 F1 skóre

Harmonický priemer (viď rovnica 3.22) medzi presnosťou a senzitivitou značí F1 skóre. Pre najlepší výsledok dosahuje hodnoty 1, naopak pre najhorší 0 [29]. V prípade viacerých tried sa táto metrika počíta zvlášť a nakoniec sú zpriemerované do $F1_{\text{macro}}$.

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{3.22}$$

3.4.4 AUROC

AUROC značí skratku pre anglický názov *Area Under the Receiver Operating Characteristics*. Tiež sa táto metrika zvykne označovať ako AUC-ROC krivka a vyplýva z chybovej matice. Vyjadruje ako správne je metóda schopná rozlišovať medzi danými triedami [11]. Vyššia hodnota blížiaca sa 1 znamená lepšie odlišovanie. Metóda s hodnotou tejto metriky blízka 0 naopak značí najhoršiu mieru schopnosti odlišovania a vyhodnocuje triedy opačne.

Krivka ROC sa vytvorí na základe **True Positive Rate** (viď rovnica 3.23), ďalej TPR, proti **False Positive Rate** (viď rovnica 3.24), ďalej FPR, kde na ose x sa zobrazujú hodnoty FPR a na ose y hodnoty TPR.

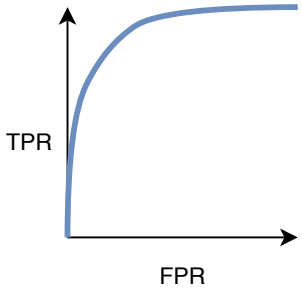
$$TPR = \frac{TP}{TP + FN} \tag{3.23}$$

$$FPR = \frac{TN}{FP + TN} \tag{3.24}$$

3.5 Zhrnutie doterajších využitých postupov

Z dôvodu rôznych dátových sád využitých pri trénovaní a testovaní je nemožné porovnávať presné výsledky vo vyhodnocovacích metrikách. Predošlé prístupy sa zamerali hlavne na extrakciu príznakov ako napríklad syntaktické príznaky. Jednalo sa o použitie slovných druhov a ich využitia pri Bag-Of-Words prístupe.

Postupne sa začínajú zahŕňať aj nejazykové príznaky. Tendencia užívateľov o opakovaný ofenzívny či nenávistný prejav, pohlavie, vek či etnikum autora môže pomôcť pri klasifikácii, ale tie zvyčajne nie sú dostupné alebo sú nespoľahlivé [32]. Okrem charakteristík používateľa



Obr. 3.10: Príklad krivky ROC s hodnotou AUROC 0.75

boli využité aj vlastnosti príspevku, napríklad čas publikácie, počet odpovedí na príspevok, ale aj počet odoberateľov používateľa a počet sledovaní.

Kapitola 4

Dátové sady vhodné pre trénovanie modelu

Zozbieranie ofenzívneho a nenávistného obsahu je náročný proces. Je zložité zoskupiť väčší objem a ešte komplikovanejšie tento obsah klasifikovať. V porovnaní s ostatnými klasifikačnými problémami pre tento fenomén neexistuje mnoho dátových sád. Používané dátové sady vytvorené pre trénovanie klasifikácii nenávistných prejavov sú v angličtine, nemčine, taliančine a holandčine [10]. Označovanie jednotlivých tried prebieha ručným označovaním viacerých anotátorov.

Medzi rozšírenejšie dátové sady používané k trénovaniu modelov sa používajú príspevky (angl. *tweets*) užívateľov Twitteru. Sú to príspevky dlhé maximálne 280 znakov.

4.1 Dátové sady

Najčastejšou využívanou dátovou sadou je sada vytvorená Thomasom Davidsonom a kolektívom. Obsahuje viac ako 24000 tweetov v anglickom jazyku. Každý tweet má označenie triedy jednej z troch kategórii: nenávistný prejav, ofenzívny jazyk bez nenávistného prejavu a prejav bez ofenzívneho a nenávistného jazyka. 16 % tweetov je označených ako neofenzívnych a zvyšných 5 % ako tweety obsahujúci nenávistný jazyk. Väčšina má označenie ofenzívneho jazyka, až 77 %. Okrem označenia triedy sada obsahuje celý text tweetu [7]. Táto dátová sada nemá explicitné rozdelenie na trénovaciu a vyhodnocovaciu sadu. Dátovú sadu vytvorili na základe slovníka slov z Hatebase.org, ktorý obsahoval frázy využívane pri nenávistnom alebo ofenzívnom jazyku. Zozbierali príspevky od 33458 rôznych užívateľov obsahujúcich práve slová zo slovníka. Následne získali všetky príspevky od týchto používateľov (85.4 miliónov príspevkov) z ktorých náhodne vybrali 25000. Túto sadu ručne anotovali 3-6 anotátori z CrownFlower.

Druhá najpočetnejšia dátová sada s okolo 17000 tweetmi v angličtine bola vytvorená pre trénovanie modelu vytvoreným Waseem a kolektívom. Tweety sú označené taktiež jednou z troch kategórii a to tweet obsahuje sexizmus, rasizmus a neobsahujúci sexizmus a rasizmus [32]. Okrem tejto sady vytvorili aj ďalšiu [31]. Celé textové znenie tweetov chýba, je uvádzané len identifikačné číslo tweetu. Tento spôsob uloženia spôsobuje to, že mnohé po viac ako 2 rokoch od zozbierania nie sú dostupné.

https://github.com/t-davidson/hate-speech-and-offensive-language
https://github.com/ZeerakW/hatespeech

Dátová sada	Počet dokumentov	Jazyk	Rok	Triedy dokumentov
T. Davidson, D. Warmsley, M. Macy	24802	angličtina	2016	ofenzívny jazyk (76%), nenávistný prejav (5%), neofenzívny jazyk (16.6%)
Waseem, Zeerak	16899	angličtina	2016	rasizmus (12%), sexizmus (20%), ani jedno z predchádzajúcich (68%)
Wikipédia: osobné útoky	viac ako 100000	angličtina	2016	osobný útok (11.7%), bez útoku (88.3%)
German hatespeech refugees	470	nemčina	2017	nenávistný prejav (32.6%), bez nenávistného prejavu (67.4%)
Hatebase	-	viac ako 70 jazykov	-	slovník obsahujúci ofenzívne a nenávistné frázy
Hades	-	holandčina	-	slovník obsahujúci ofenzívne a nenávistné frázy
Yahoo!	nedostupné	holandčina	2017	dátová sada už nie je dostupná

Tabuľka 4.1: Prehľad dátových sád zameraných na ofenzívny jazyk

Ďalšia dátová sada naväzuje na dátovú sadu vytvorenú Waseem. Okrem identifikačného čísla tweetu, označenia triedy (sexizmus, rasizmus, neobsahujúce nenávistný prejav), sú tu uvedené aj informácie získané z Twitter API - čas vytvorenia, počet rokov od vytvorenia profilu a ďalšie textové vlastnosti tweetu - počet spomenutí iných užívateľov, počet znakov, počet URL, počet hashtagov, pomer veľkých a malých písmen [20].

S cieľom overiť spoľahlivosť anotácií vznikla dátová sada znova zo sociálnej siete Twitter [28]. Okrem označení tried nenávistný prejav, neobsahujúci nenávistný prejav, obsahuje aj mieru ofenzívnosti.

Okrem zozbieraných nenávistných prejavov zo sociálnej siete je dostupná sada z online encyklopédie Wikipédie. Dáta v nej zahrnuté sú označené diskusie v komentároch v anglickom jazyku. Dátová sada je zameraná predovšetkým na osobné útoky a obťažovanie užívateľov [34].

Najrozmanitejším zdrojom je internetová organizácia Hatebase. Tá poskytuje ofenzívne slovné spojenia vo viac ako 80 jazykoch. Okrem miery ofenzívnosti záznamy obsahujú aj cieľ (náboženstvo, etnika).

4.2 Nedostatky dátových sád

Vzhľadom na komplexnosť fenoménu sa pri zbieraní dát vedci snažia o maximalizáciu zachytenia nenávistných prejavov. Využívajú sa preto ustálené ofenzívne pomenovania [20]. Táto metóda môže značne ovplyvniť rozmanitosť dátových sád.

https://github.com/GreenParachute/hate-speech-popularity

Dátové sady vytvárajú užívatelia, nie sú generované automatizovane. Týmto pádom užívatelia ovplyvňujú dostupnosť ich príspevkov mazaním, archiváciou, poprípade deaktivovaním profilu. Ak sú teda v dátovej sade uložené len identifikačné čísla príspevkov a nie ich celé znenie, po určitej dobe nemusia byť dostupné.

Nedostatkom dátových sád môžu byť chýbajúce informácie o autoroch textu. Dostupné dátové sady bývajú zväčša anonymizované. V súčasnej dobe sa rozširuje snaha zahrnúť zázemie užívateľov do vytváraných modelov takisto ako aj ich tendencia k opakovaným ofenzívnym či nenávistným prejavom.

Z pohľadu porovnania výsledkov s inými metódami je možné vnímať chýbajúce rozdelenie dátovej sady na trénovaciu, validačnú a testovaciu za nedostatok.

Ďalším problémom je spoľahlivosť označenia fenoménu dokumentu. Tento nedostatok súvisí s práve subjektívnym chápaním ofenzívneho jazyka, ale predovšetkým nenávistných prejavov v jazyku.

Kvalitu dátových sád ovplyvňuje aj fakt, že triedy príkladov sú početne nevyvážené. V predošle uvádzaných sadách je zastúpenie nenávistného prejavu v ich príkladoch minimálne.

Kapitola 5

Navrhované metódy riešenia detekcie

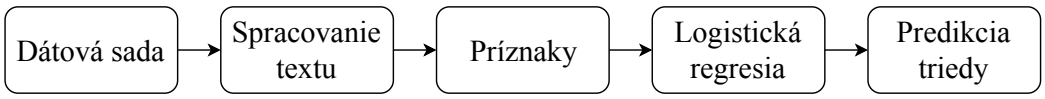
V tejto kapitole sú opísané navrhované metódy riešenia detekcie. Pre doposiaľ krátko skúmaný fenomén sa nevyužili všetky dostupné najmodernejšie metódy a trendy v spracovaní prirodzeného jazyka. Pri detekcii nenávistného jazyka sa zväčša pracovalo s príznakmi získaných spracovaním textu. Využili sa rôzne metódy, ktoré narazili na problémy pri klasifikácii ofenzívneho jazyka.

5.1 Modely vychádzajúce z príznakov

Pre základné porovnávajúce modely som zvolila metódu logistickej regresie. Dátová sada pripravená na trénovanie modelu sa spracuje a získajú sa potrebné príznaky. Tie som zvolila nasledovne:

- vlastnosti tweetu (retweet, počet spomenutí),
- n-gramy označenia slovných druhov TF-IDF (unigramy, bigramy, trigramy),
- TF-IDF n-gramov (unigramy, bigramy),
- testy čitateľnosti (Flesch Reading Ease, Flesch–Kincaid Grade Level),
- vlastnosti textu (počet hashtagov, počet linkov, počet veľkych písmen).

Model tieto príznaky vyhodnocuje na koeficienty pravdepodobnostnej distribúcie.



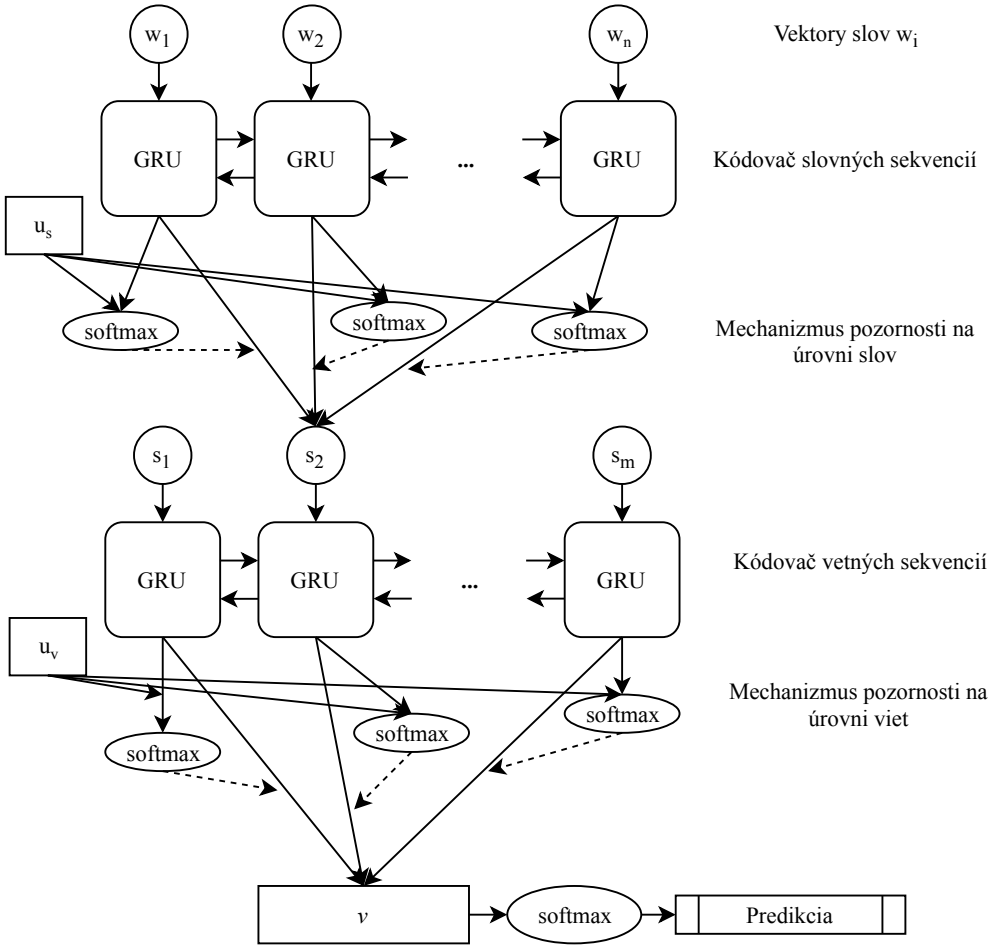
Obr. 5.1: Schéma základného modelu

5.2 Klasifikácia s mechanizmom pozornosti

Metódami bez nutnosti spracovania textu s cieľom získania príznakov sú aj modely využívajúce na vstupe slovné vektory, tzv. *embdedding* vektory. Využitie mechanizmu pozornosti poskytuje väčší prehľad o kontexte. Model sa nezameriava na konkrétne slová ani vety.

5.2.1 Sieť s hierarchickou pozornosťou

Takýmto modelom je *Hierarchical Attention Networks* (HAN) [35]. Architektúra tohto modelu pozostáva z GRU vrstvy tvoriacich enkodér slovných sekvencií, mechanizmom pozornosti na úrovni slov, GRU vrstvou tvoriacou enkodér vetných sekvencií a mechanizmom pozornosti na úrovni viet (viď obrázok 5.2). Kombinácia mechanizmov pozornosti na rôznych úrovniach vychádza z hierarchickej štruktúrovanosti textu. Pre zlepšenie informácií o kontexte model využíva obojsmerný prechod GRU [5] pre slovné aj vetné sekvencie.



Obr. 5.2: Model *Hierarchical Attention Networks* [35]

Zo vstupnej sekvencie slov sa vytvorí sekvencia vektorovýcz reprezentácii slov $w_m1, \ldots, w_mn$. Výstupom obojsmerného GRU je pre n-té slovo z vety m $h_{mn} = [\overrightarrow{h_mn}, \overleftarrow{h_mn}]$. Zosumarizovaná informácia sa predáva mechanizmu pozornosti na úrovni slov u_n (viď rovnice 5.1) pridelí pozornosť na základe váh W_s a kontextového vektora u_s.

$$u_mn = \tanh(W_s \cdot h_mn + b_s) \tag{5.1a}$$

$$\alpha_mn = \frac{\exp({u_{mn}}^T u_s)}{\sum_n \exp({u_{mn}}^T u_s)} \tag{5.1b}$$

$$s_m = \sum_m \alpha_{mn} h_{mn} \tag{5.1c}$$

Následné spracovanie vektorov viet je analogické ako pre vektorové reprezentácie slov (viď rovnice 5.2). Vektor vety s_m spracúva ďalej vetný enkodér pozostávajúci z ďalšej obojsmernej GRU vrstve. Výstupom je $h_m = [\overrightarrow{h_m}, \overleftarrow{h_m}]$, ktorému mechanizmus pozornosti na úrovni viet prináleží pozornosť na základe váh W_s a kontextového vektoru vety u_v. Celkovým výstupom je vektor v, ktorý reprezentuje celý dokument. Tento vektor sa využíva pre klasifikáciu tohto dokumentu.

$$u_m = \tanh(W_v \cdot h_m + b_v) \tag{5.2a}$$

$$\alpha_m = \frac{\exp({u_m}^T u_v)}{\sum_m \exp({u_m}^T u_v)} \tag{5.2b}$$

$$v = \sum_m \alpha_m h_m \tag{5.2c}$$

5.2.2 Model so self-attentive mechanizmom pozornosti

Ďalším modelom využívajúcim mechanizmus pozornosti je *structured self-attentive sentence embedding*. Architektúra modelu pozostáva z obojsmerného LSTM, mechanizmu pozornosti a klasifikátora.

Na vstupe spracováva maticu S konkatenovaných vektorových reprezentácii n slov danej sekvencie $(w_1, w_2, \ldots w_n)$. Vektor w_n je vektor s d dimenziami a teda rozmer vstupnej matice je $n \times d$. Následne maticu S spracuje obojsmerná vrstva LSTM (viď rovnice 5.3). Výstup H s rozmerom $n \times 2 \cdot u$ ďalej spracúva mechanizmus pozornosti (viď rovnica 5.4), schopný spracovať variabilnú dĺžku vstupu n. W_{s1} je matica s rozmerom $d_a \times 2 \cdot u$. Rozmer d_a je možné zvoliť ľubovoľne. W_{s2} je matica s rozmerom $r \times d_a$, kde r je počet extrahovaných pozorností.

$$\overrightarrow{h_n} = \overrightarrow{LSTM}(w_n, \overrightarrow{h_{n-1}}) \tag{5.3a}$$

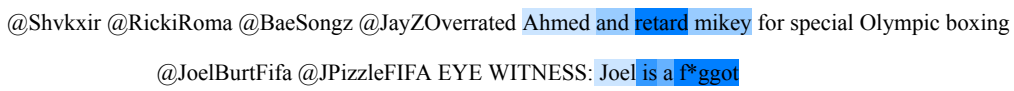
$$\overleftarrow{h_n} = \overleftarrow{LSTM}(w_n, \overleftarrow{h_{n-1}}) \tag{5.3b}$$

$$h_n = [\overrightarrow{h_n}, \overleftarrow{h_n}] \tag{5.3c}$$

$$H = (h_1, h_2, \ldots, h_n) \tag{5.3d}$$

$$A = softmax(W_{s2} \cdot \tanh(W_{s1} \cdot H^T)) \tag{5.4}$$

Výhodou využitia metód založených na mechanizmu pozornosti je vizualizácia, na ktoré časti textu sa model zameral. Náhľad s teplotnou mapou (viď obrázok 5.3) uľahčuje analýzu vlastností nenávistných prejavov a celkovo ofenzívneho jazyka.



Obr. 5.3: Vizualizácia vyhodnotenia modelu s mechanizmom pozornosti

5.3 Konvolučné neurónové siete

Konvolučné neurónové siete (angl. *convolutional neural network*, skr. CNN, ConvNet) sa najčastejšie využívajú pri analýze obrazových vstupných dát pri počítačovom videní. Spracovaním textového vstupu do určitej abstrakcie signálu je možné ich zúžitkovať aj pre klasifikáciu úlohy textu. Hlavnými časťami konvolučných sietí sú konvolučné vrstvy, tzv. *pooling* vrstvy, aktivačné vrstvy a *dropout* vrstvy.

Konvolučná vrstva Konvolučná vrstva, tiež skrytá vrstva, pozostáva z učiacich sa filtrov (často aj jadier, kernelov). Tie majú obmedzené pole videnia, na základe ktorých vstup transformujú pre ďalšie spracovanie. Toto pole videnia môžme chápať ako maticu, pre ktorú sa rozhodne jej veľkosť. Typicky je tento rozmer v rádoch jednotiek a jeho šírka a výška je zvyčajne rovnaká. S touto maticou prebieha konvolúcia, kedy sa matica priloží na každú možnú časť vstupu. Výsledkom každého posunutia je jedna výstupná hodnota. Faktorom ovplyvňujúcim spracovanie je veľkosť kroku (angl. *stride*), ktorá určuje o koľko sa má filter posúvať. Ďalšou úpravou je veľkosť a hodnota výplne (angl. *padding*) pre vstupnú maticu. Okraje vstupu sa obalia zvyčajne nulovými hodnotami (angl. *valid padding*).

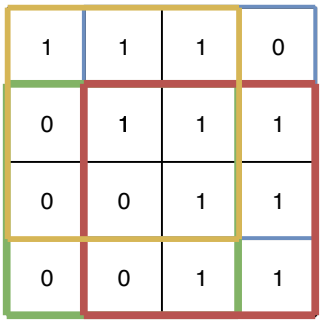
$$y(n_1, \ldots, n_M) = \sum_{k_1=-\infty}^{\infty} \cdots \sum_{k_M=-\infty}^{\infty} h(k_1, \ldots, k_M) \cdot x(n_1 - k_1, \ldots, n_M - k_M) \tag{5.5}$$

Poolingová vrstva Cieľom poolingovej vrstvy je znížiť veľkosť spracovávaných dát. Znižuje časovú a pamäťovú náročnosť následných operácií. Táto vrstva funguje podobne ako konvolučná. Hyperparametrami upravujúcimi správanie vrstvy sú veľkosť okna a krok posuvu okna.

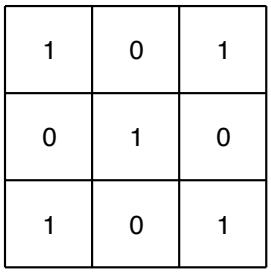
Vstup je postupne prechádzaný oknom, z ktorého sa vypočíta hodnota, ktorá je následne jednou z hodnôt výstupu. Najčastejšie je využívaný *max pooling*, zo spracovávaného okna sa vyberie maximálna hodnota, *average pooling*, výstupom je priemer všetkých hodnôt v danom okne.

Aktivačná vrstva Aktivačné vrstvy pozostávajú z aktivačných funkcií (viď kapitola 3.1.6). V prípade konvolučných sietí prevláda využívanie ReLU aktivačnej funkcie. Táto vrstva nemá bližšie špecifikujúce hyperparametre.

Dropout vrstva Táto vrstva pomáha predchádzať neurónovým sieťam preučeniu (angl. *overfitting*) a podporuje lepšiu schopnosť schopnosti generalizácie. Využívajú sa výlučne pri trénovaní. Vrstva upravuje vstup tak, aby bol výstup zakaždým iný a teda sieť by sa nemohla naučiť presnej postupnosti. Pre každú hodnotu vstupu sa na základe pravdepodobnosti vynulovania hodnoty rozhodne o jej vynulovaní. Hyperparametrom tejto vrstvy je práve pravdepodobnosť (angl. *drop-out rate*), hodnota z intervalu $\langle 0, 1 \rangle$.



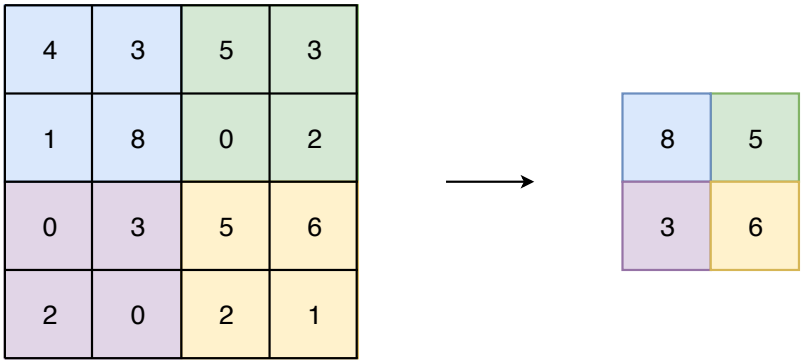
(a) Matica vstupu s naznačenými aplikáciami filtra 5.4b a krokom 1



(b) Konvolučný filter s rozmerom 3x3

(c) Výstup matice po aplikovaní filtra 5.4b

Obr. 5.4: Konvolúcia v 2D priestore s jedným filtrom



Obr. 5.5: Vizualizácia poolingovej vrstvy s využitím max poolingu veľkosťou okna 2 a veľkosťou kroku 2

5.3.1 Konvolučná neurónová sieť so vstupom na úrovni znakov

Prevažná väčšina klasifikačných metód je založená na extrakcii príznakov na úrovni slov (n-gramy, TF-IDF, Bag of Words). Avšak tieto metódy zlyhávajú hlavne pri prípadoch, kedy užívatelia zámerne urobia typografickú chybu. Jedná sa napríklad o zámenu písmena za číslicu, kedy slovo *girl* nahradia slovom *g1rl* [26]. Hypotézou je, že tento problém by mohla vyriešiť práve metóda pracujúca na úrovni charakterov, ktorá sa nezameriava na slová, ale písmená. Významnou nevýhodou použitia konvolučných sietí je nutnosť obrovskej dátovej sady. Takýto model bol navrhnutý Xiang Zhang, Junbo Zhao a Yann LeCun [37]. Je to obdobná metóda ako konvolučná neurónová sieť na úrovni slov [18].

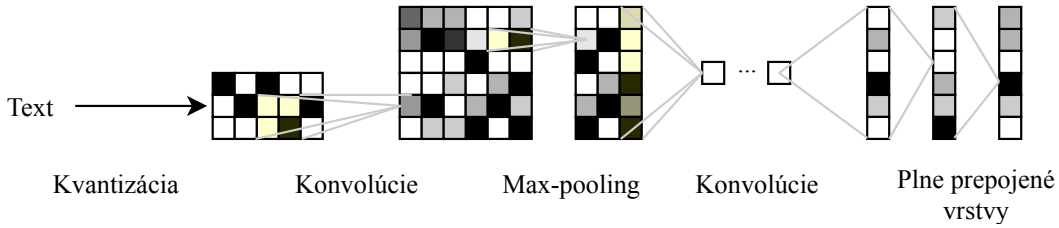
V tejto metóde sa spracováva text ako určitý signál bez ďalších syntaktických a sémantických príznakov. Text sa v prvom kroku upraví z postupnosti slov na množinu vektorov, ktoré majú podobný tvar kódu 1 z n, kde hodnotu "1" má práve jeden príznak. Číslo n predstavuje veľkosť prijateľnej abecedy. Konkrétnu ukážku kódovania je možné vidieť v ta-

buľke 5.1. Počet vektorov je daný fixnou veľkosťou l_0. Hodnota l_0 vyjadruje počet znakov, ktoré by stačili na zachytenie väčšiny významu. To znamená, že všetky znaky presahujúce túto hranicu nebudú zahrnuté. Zároveň všetky znaky neobsiahnuté v abecede budú reprezentované vektorom s nulovými hodnotami. Autori tejto metódy odporúčajú znaky v abecede ako sú písmená, číslice, interpunkčné znamienka a biele znaky (medzera, nový riadok). Rozlišovanie veľkých a malých písmen môže byť taktiež zohľadnené vo vstupnej abecede.

	A	L	L	.
A	**1**	0	0	0
B	0	0	0	0
C	0	0	0	0
⋮	⋮	⋮	⋮	⋮
L	0	**1**	**1**	0
⋮	⋮	⋮	⋮	⋮
.	0	0	0	**0**
⋮	⋮	⋮	⋮	⋮
{	0	0	0	0
}	0	0	0	0

Tabuľka 5.1: Ukážka spracovania textu *"All."* na kód 1 z n

Ďalej model (viď obrázok 5.6) pozostáva z niekoľkých konvolučných vrstiev a z plne prepojených vrstiev. Na vstupe modelu je 2D matica s rozmermi veľkosti abecedy a hodnoty l_0. Model využíva aj ďalšie *dropout* vrstvy [13].



Obr. 5.6: Model konvolučnej siete na úrovni charakterov [37]

Vstupnú maticu spracuje 1D konvolúcia (viď rovnica 5.6), kde $f(x)$ predstavuje kernelovú funkciu, k veľkosť kernelu, $g(x)$ vstup, d veľkosť kroku a $c = k - d + 1$ ofset. Výstup sa spracuje max-poolingom (viď rovnica 5.7).

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c) \tag{5.6}$$

$$h(y) = \max_{x=1}^{k} g(y \cdot d - x + c) \tag{5.7}$$

5.4 Predtrénované jazykové modely

Jazykové modely počítajú pravdepodobnosť danej postupnosti slov (viď rovnica 5.8). Tieto modely dosahujú výborné výsledky v rôznych úlohách. Sami o sebe majú kontext o jednotlivých frázach, ktoré sa môžu zdať podobné. Rozšíreným je N-gramový jazykový modelom berúci ohľad na $N-1$ predchádzajúcich slov.je Trénovanie týchto modelov je dátovo náročné. Vyžadujú obrovské generické dátové sady. Po predtrénovaní sú dolaďované (angl. *fine-tuning*) dátovou sadou špecifickou pre úlohu.

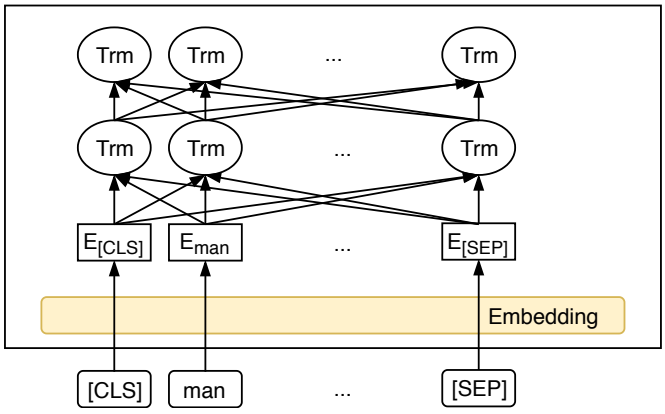
$$p(w_1, \ldots, w_d) = p(w_1) \prod_{i=2}^{d} p(w_i | w_1, \ldots, w_{i-1}) \tag{5.8}$$

5.4.1 BERT jazykový model

BERT (angl. *Bidirectional Encoder Representations from Transformers*) je jazykový model, ktorý využíva aplikáciu obojsmerne učeného transformer (mechanizmus pozornosti) [8]. Nová metóda prináša kontext textovej sekvencie zovšadiaľ všade.

Pri predtrénovaní sa využíva technika *masked language model* (MLM), kedy sa 15% tokenov zo vstupu náhodne zamaskuje. Následne sa model snaží zamaskovaný token klasifikovať len na základe kontextu. Identifikácie jednotlivých tokenov sa nahradia vektorovými reprezentáciami WordPiece týchto tokenov. Tieto tokeny tvoria najčastejšie časti slova Je nutné aby sekvencia tokenov začínala *[CLS]* tokenom a bola ukončená špeciálnym tokenom *[SEP]*. Maskovaný token sa zvyčajne nahradí tokenom *[MASK]*. V prípadoch kedy sa token maskuje, sa slovo nahradí iným v 10% a v 10% zostáva nezmenené.

Pre klasifikáciu sa následne využívajú výstupy z poslednej vrstvy enkodéra. Aj keď je proces predtrénovania modelu náročný, je nutné ho spraviť len raz a predtrénovaný model je dostupný.



Obr. 5.7: Architektúra modelu BERT, enkodér využívajúci transformery

5.4.2 Univerzálny jazykový model

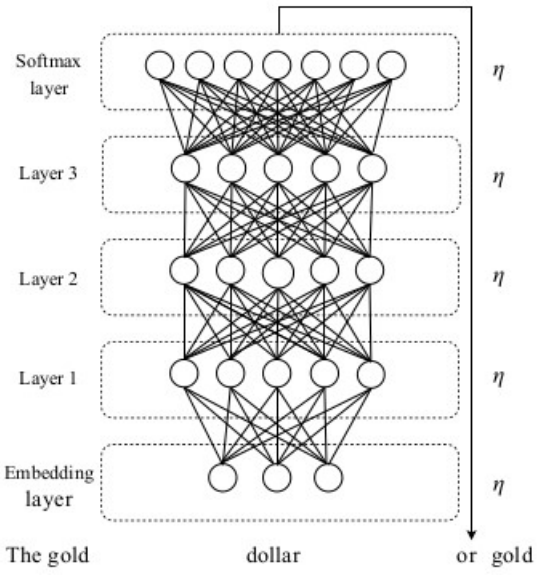
Model *Universal Language Model Fine-tuning* [15] (ULMFiT) vychádza z *AWD-LSTM* modelu [23]. Architektúra tohto modelu pozostáva z 3 LSTM vrstiev. Rieši problém v prípadoch kedy nie sú dostupné obrovské trénovacie dátové sady. Metóda pozostáva z 3 fáz (viď obrázky 5.8). V prvom kroku sa model vytrénuje obrovskej všeobecnej dátovej sade. Takáto dátová sada by mala zohľadňovať všeobecné znaky jazyka. Vhodnou je napríklad

korpus Wikitekt-103. Obsahuje články z Wikipédie s celkovým počtom slov 103227021, slovník tvorí viac ako 267000 slov [22].

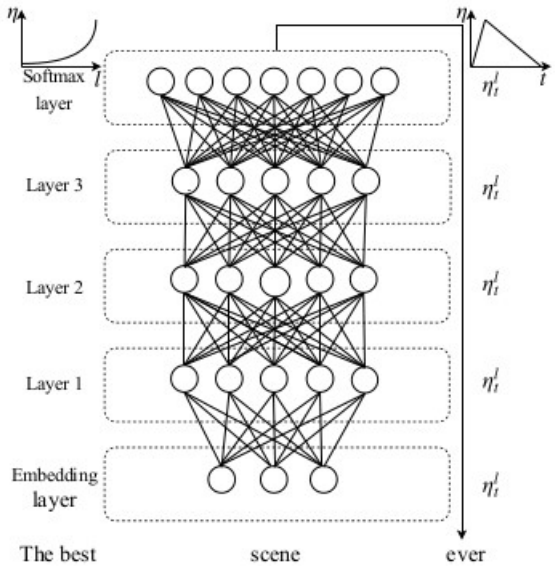
Ďalším krokom je fine-tuning modelu pre konkrétny klasifikačný problém s trénovacou sadou určenou pre danú úlohu. Z predpokladu autorov, že každá vrstva modelu zachycuje inú informáciu, je využívané diskriminatívne dolaďovanie (angl. *discriminative fine-tuning*). Namiesto rovnakého koeficientu rýchlosti učenia je pre každú vrstvu rôzna. Druhou úpravou je aj úprava koeficientu rýchlosti učenia. Dôvodom je cieľ rýchlej konvergencie do výhodnej hodnoty a následné ustálenie. Docielené je to pomocou postupného zvyšovania a následne postupného znižovania rýchlosti učenia s ohľadom na počet iterácií. Pre výsledný graf veľkosti tohto koeficentu sa ako trojuholníkové učenie (angl. *triangular learning rates*)

Pre trénovanie klasifikátora je schéma modelu poupravená pridaním lineárnych vrstiev. Vo fáze trénovania je jednotlivým vrstvám postupne umožňované ladenie.

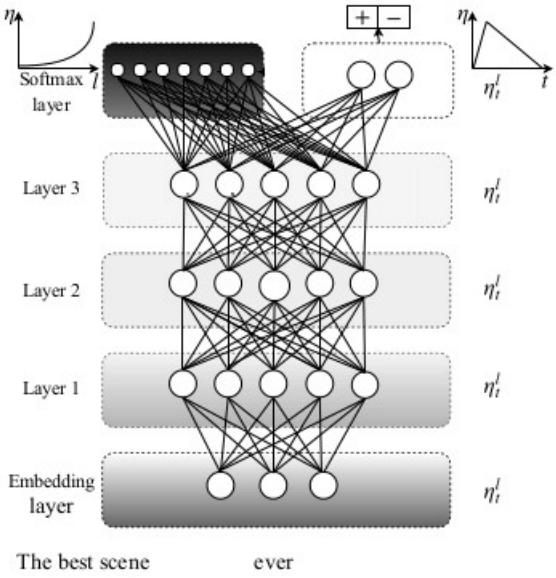
https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/



(a) Jazykový model pre predtrénovanie



(b) Fine-tuning jazykového modelu



(c)

Obr. 5.8: ULMFiT architektúra modelu pre jednotlivé fázy trénovania: a) predtrénovanie jazykového modelu na všeobecnej dátovej sade, b) diskriminačný fine-tuning modelu na dátovej sade z riešenej domény, c) trénovanie klasifikátoru, prechod bielej farby do čiernej znázorňuje postupné umožnenie ladenia [15]

Kapitola 6

Implementácia, experimenty a vyhodnotenie

Cieľom bolo implementovať a porovnať modely, ktoré vedia klasifikovať ofenzívny a nenávistný jazyk na základe trénovacích dát. V tejto kapitole sú popísané implementačné detaily a vykonané experimenty.

Na implementáciu navrhovaných modelov bolo použité prostredie Python 3.6 za pomoci frameworku PyTorch 1.2.1. Skripty boli spúšťané pod operačným systémom Ubuntu 18.04.1. Na trénovanie daných modelov bola využitá grafická karta GeForce RTX 2080 Ti.

6.1 PyTorch

PyTorch je open-source nástroj, ktorý poskytuje knižnicu pre jazyk Python vhodnú na strojové učenie. Umožňuje vykonávať výpočty na GPU aj CPU. Obsahuje prostriedky na vytvorenie jednotlivých vrstiev modelov, optimalizačné algoritmy a stratové funkcie.

Výpočty sa prevádzajú nad tenzormi, sú to zovšeobecnené objekty znázorňujúce vektory či matice. Tieto objekty tvoria n-rozmerné matice jedného dátového typu. Tvar tenzoru určuje počet dimenzií a veľkosť dimenzií.

6.2 Tréning

6.2.1 Hodnota koeficientu učenia

Gradient a jeho veľkosť určujú ako sa budú aktualizovať váhy. Pri optimalizácii sa postupuje v smere negatívneho gradientu,

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E(w^{(\tau)}) \tag{6.1}$$

kde η predstavuje veľkosť kroku (angl. *learning rate*).

6.2.2 Epochy

Jedna epocha znamená spracovania celej dátovej sady cez model. Zväčša je dátová sada veľká a preto je nutné ju rozdeliť na časti.

[1] https://pytorch.org

6.2.3 Veľkosť skupiny

Z dôvodu veľkého množstva príkladov v dátovej sade sa dátova sada rozdeľuje na časti (angl. *mini-batches*).

6.2.4 Iterácie

Iterácie predstavujú počet skupín, na ktoré bola dátová sada rozdelená.

6.2.5 Chybová funkcia

Pre trénovanie klasifikačného problému s C triedami, bola zvolená *cross-entropy chybová funkcia* vychádzajúca z negatívneho logaritmu pravdepodobností [3].

Pre rovnicu 6.2 predstavuje y_n pravdepodobnosť správneho označenia pre distribúciu $\widehat{y}_n$ pre daný model.

$$J(w) = -\log p(y_n|\widehat{y}_n) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log \widehat{y}_n + (1-y_n)\log(1-\widehat{y}_n)] \tag{6.2}$$

6.2.6 Optimalizačná metóda

Pri trénovaní bola využitá optimalizačná metóda Adam (odhad adaptivního momentu, angl. *adaptive moment estimation*). V práci bolo experimentované aj s optimalizačnou metódou SGD (stochastický gradientný zostup, angl. *stochastic gradient descent*), avšak metóda Adam sa preukázala byť ako efektívnejšia a viedla k rýchlejšiemu učeniu modelu.

6.2.7 Regularizácia

Regularizácia sa využíva k predchádzaniu over-fittingu. Takou metódou je L1 regularizácia (angl. *lasso regression*) alebo L2 regularizácia (angl. *ridge regression*)

6.3 Rozdelenie dátovej sady

Z princípu fungovania strojového učenia, predikcia neznámych dát, je potrebné rozdeliť využívanú dátovú sadu na trénovaciu a evalvačnú, resp. testovaciu. Takže trénovacia sada je tá, na základe ktorej sa model učí a aktualizuje váhy. S testovacou sadou sa vyhodnocujú výsledky správnosti fungovania modelu, ako je schopný rozlišovať jednotlivé triedy. Oddelenie ďalšej časti, validačnej, od trénovacej dátovej sady. Táto skupina príkladov poskytne nezávislé hodnotenie modelu počas trénovania. Je určená na vyhodnocovanie modelu, hlavne pri hľadaní hyperparametrov. Odporúčaný pomer veľkostí trénovacej, validačnej a testovacej časti sa odporúča na 7:1:2.

Miesto oddelenia validačnej časti príkladov je možné využiť aj krížovú validáciu (angl. *cross-validation*). Trénovacia časť sa ale rozdelí na viacero častí. Následne sa zvolí ako jedna časť ako validačná a ostatné sa použijú k natrénovaniu modelu klasifikátoru. Tento postup sa zopakuje práve toľko krát, na koľko častí sa rozdelila trénovacia sada.

	Nenávistný prejav	Ofenzívny jazyk	Neutrálny jazyk	Celkovo
trénovacia	1013	13365	2970	17348
v %	5.8	77	17.2	
validačná	141	1948	389	2478
v %	5.7	78.6	15.7	
testovacia	276	3877	804	4957
v %	5.6	78.2	16.2	

Tabuľka 6.1: Rozdelenie príkladov medzi triedami v trénovacej, testovacej a validačnej sady z dátovej sady [7]

6.4 Implementácia jednotlivých modelov

Každý model je implementovaný vo vlastnom Python moduly. Modul obsahuje konfikuračný súbor `config.json`. V tomto súbore sa špecifikujú parametre pre daný model (cesta k dátovým sadám, koeficient učenia, ...). Moduly obsahujú súbor `main_{model}.py`, ktorý po spustení načíta konfiguračný súbor a podľa neho načíta dátovú sadu a inicializuje PyTorch modul (výber typu hardwéru na trénovanie - CPU/GPU), optimalizačnú metódu a chybovú funkciu. Následne sa model trénuje a testuje.

6.4.1 Model logistickej regresie s niekoľkými príznakmi

Po načítaní dát sa vrámci predspracovania extrahujú príznaky do feature vektora. Syntaktické príznaky sú extrahované regulárnym výrazom, tokenizácia viet je implementovaná s `PunktSentenceTokenizer` z balíčka `nltk`. Na tokenizáciu slabik je využitý framework `pyphen`. Na tokenizáciu slov sa využíva balík `spaCy`, ktorý poskytuje priraďovanie slovných druhov k daným slovám. TF-IDF príznaky sa získajú s pomocou `TfidfVectorizer`. Vektor príznakov sa normalizuje a predáva sa ďalej na trénovanie alebo testovanie. Výsledky testovania nad dátovou sadou je možné vidieť v tabuľke 6.2.

	Počet príznakov	#Θ	Acc	$\mathbf{F1_{macro}}$
LR	153K	459K	0.69	0.541

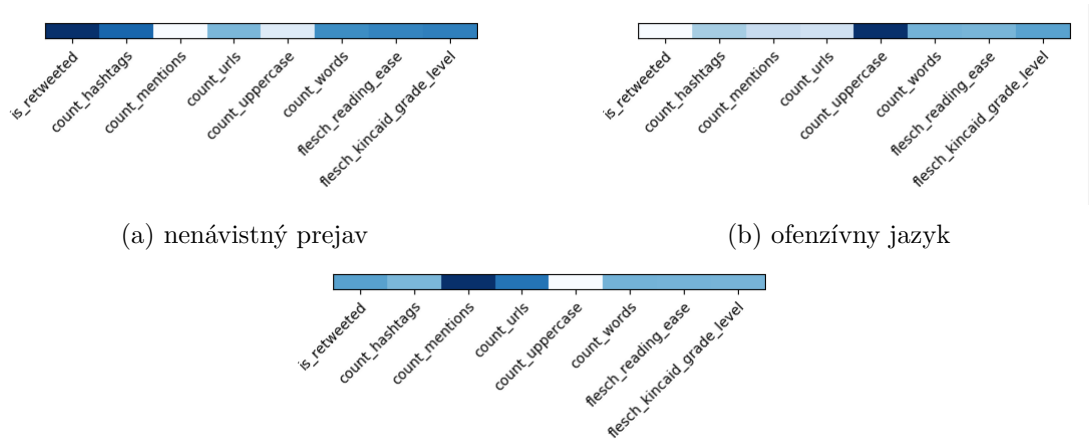
Tabuľka 6.2: Dosiahnuté výsledky na testovacej dátovej sade pomocou logistickej regresie

Vplyv príznakov

Vhodný výber skúmaných vlastností príkladov patrí k strojovému učeniu. Na tomto modely sa preukázalo, že príspevky obsahujúce nenávistný prejav boli vytvorené za účelom reakcie na iný príspevok tzv. *retweet*. Pre ofenzívny jazyk bol najpríznačnejším príznakom prítomnosť veľkých písmen v dokumente. Porovnanie príznakov a ich váh na vyhodnotenie je možné vidieť na obrázkoch 6.1. Oba testy čitateľnosti sa preukázali ako vhodné pre všetky klasifikované triedy.

6.4.2 Model s TF-IDF bag-of-words prístupom

Model využíva len bag-of-words príznaky, ktoré sú získané funkciou `TfidfVectorizer` z balíčka `sklearn`. Výsledky testovanie je možné vidieť v tabuľke 6.3.



(a) nenávistný prejav

(b) ofenzívny jazyk

(c) neutrálny jazyk

Obr. 6.1: Váha príznakov pri rozpoznávaní tried, príznaky s tmavším anotovaním majú pri rozhodovaní triedy väčší vplyv; príznak *is_retweeted* značí či bol príspevok retweetnutý, príznak *count_hashtags* - počet hashtagov, príznak *count_mentions* - počet označení iných užívateľov, príznak *count_urls* vyjadruje počet URL v dokumente, príznak *count_uppercase* - počet veľkých písmen v dokumente, príznak *count_words* - počet slov v dokumente, príznaky *flesch_reading_ease* a *flesch_kincaid_grade_level* - testy čitateľnosti

	Veľkosť slovníka	#Θ	**Acc**	$\mathbf{F1_{macro}}$	$\mathbf{F1_{HS}}$	$\mathbf{F1_{OL}}$	$\mathbf{F1_{N}}$
BoW	32K	111K	0.863	0.712	0.397	0.916	0.825

Tabuľka 6.3: Dosiahnuté výsledky na testovacej dátovej sade

6.4.3 Konvolučná neurónová sieť na úrovni znakov

Vstupné dáta sú tokenizované na znaky, ktoré sa v prípade potreby (nezahrnutie veľkých písmen do vstupnej abecedy) zmenia na ich variantu malého písmena. Pri implementácii modelu som vypustila spätnú kvantizáciu (posledné písmená sa spracujú ako prvé a prvé ako posledné), pretože výsledky neovplyvňovala.

	#Θ	**Acc**	$\mathbf{F1_{macro}}$	$\mathbf{F1_{HS}}$	$\mathbf{F1_{OL}}$	$\mathbf{F1_{N}}$
$\text{CharCNN}_{\text{case\_differ}}$	96M	0.823	0.671	0.3451	0.895	0.775
$\text{CharCNN}_{\text{lowercase}}$	95M	0.846	0.670	0.3163	0.907	0.787

Tabuľka 6.4: Dosiahnuté výsledky na testovacej sade, #Θ vyjadruje počet trénovaných parametrov, model $\text{CharCNN}_{\text{case\_differ}}$ mal vo vstupnej abecede obsiahnuté aj malé aj veľké písmená, model $\text{CharCNN}_{\text{lowercase}}$ obsahoval abecedu len s malými písmenami

Vplyv abecedy na výsledky

Z poznatku o príznačnosti príznaku veľkých písmen z modelu pracujúceho s príznakom počtu veľkých písmen (viď kapitola 6.4.1) by bolo očakávané, že rozlišovanie veľkých a malých písmen pomôže zvýšiť úspešnosť detekcie. Experimentálne však vyplynulo, že táto úprava nemá na výslednú úspešnosť vplyv.

Maskovanie fráz

Princíp tohto modelu si zakladá na reprezentácii textu ako postupnosti znakov. To znamená, že model nepotrebuje mať informáciu o slove, ale ku vstupu sa správa ako ku určitému signálu. Z tohto poznatku by mohol model, byť schopný riešiť maskovanie (typografická chyba, zámerná zámena znakov za iné). Model bol otestovaný na pozmenenej dátovej sade, z ktorej sa náhodne ofenzívne frázy zmenili pridaním znaku (medzera) alebo zmenením znaku (číslica alebo znak ∗ miesto písmena).

6.4.4 Model so self-attentive mechanizmom pozornosti

Model so self-attentive mechanizom (viď kapitola 5.2.2) bol vyskúšaný s rôznymi vektorovými reprezentáciami (GloVe, ELMo, BERT).

	$\boldsymbol{p}$	$\#\Theta$	**Acc**	$\mathbf{F1_{macro}}$	$\mathbf{F1_{HS}}$	$\mathbf{F1_{OL}}$	$\mathbf{F1_{N}}$
$\text{Attention}_{\text{ELMO}}$	0.7	57M	0.896	0.761	0.470	0.937	0.874
$\text{Attention}_{\text{BERT}}$	0.0	166M	0.846	0.681	0.342	0.908	0.793

Tabuľka 6.5: Dosiahnuté výsledkyna testovacej dátovej sade, $\boldsymbol{p}$ vyjadruje penalizačný koeficient

Vizualizácia pozornosti

Interpretáciu jednotlivých pozorností je možné vizualizovať a to buď pre každú hlavu pozornosti samostatne alebo zjednotením pozornostní do jednej. Pre zjednotenie jednotlivých pozornostní je nutné spočítať riadkové vektory do jedného a normalizovať hodnoty tak, aby sa súčet hodnôt novo vzniknutého vektora rovnal hodnote 1. Vytvorí sa vektor, ktorý pripomína pravdepodobnostné rozloženie. Následné je jednoduché určiť, ktorá hodnota nadobúda maxima a teda na akú časť sa sústreďuje pozornosť najviac.

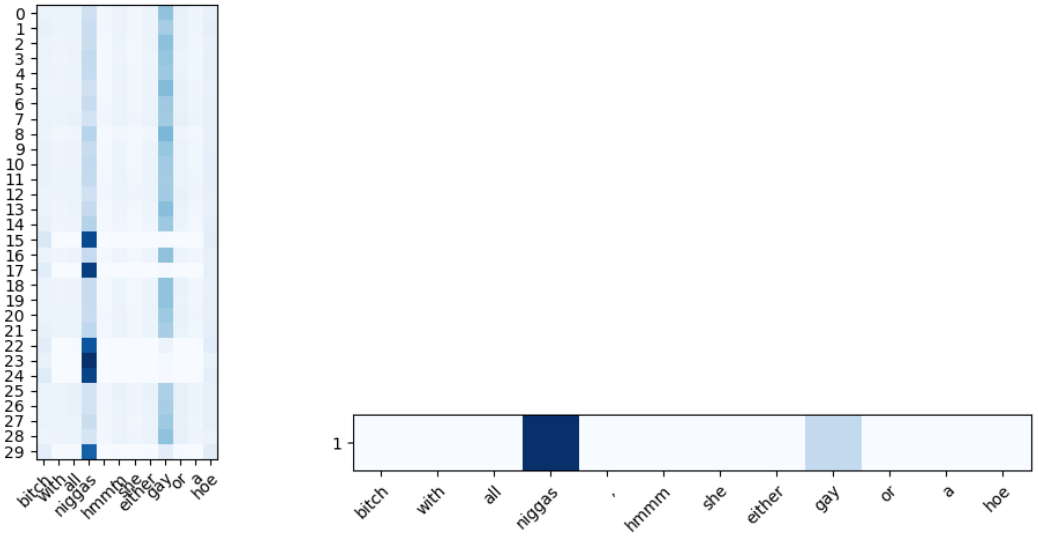
Z tejto vizualizácie je následne jednoduché určiť, ktoré slová najviac ovplyvňujú klasifikáciu. Pozornosť sa zväčša sústreďovala pri nadávkach a neslušných slovách (*b*tch*, *f*g*, *f*ggot*, *n*ger*, , *c*nt*, *h*oe*, *retard*, *sh*t*, *queer*, *f*ck* a ďalšie tvary tohto slova, ...) alebo aj pri slovách so sexuálnym významom (*p*ssy*, *d*ck*, *nudes*, *titties*).

Penalizácia

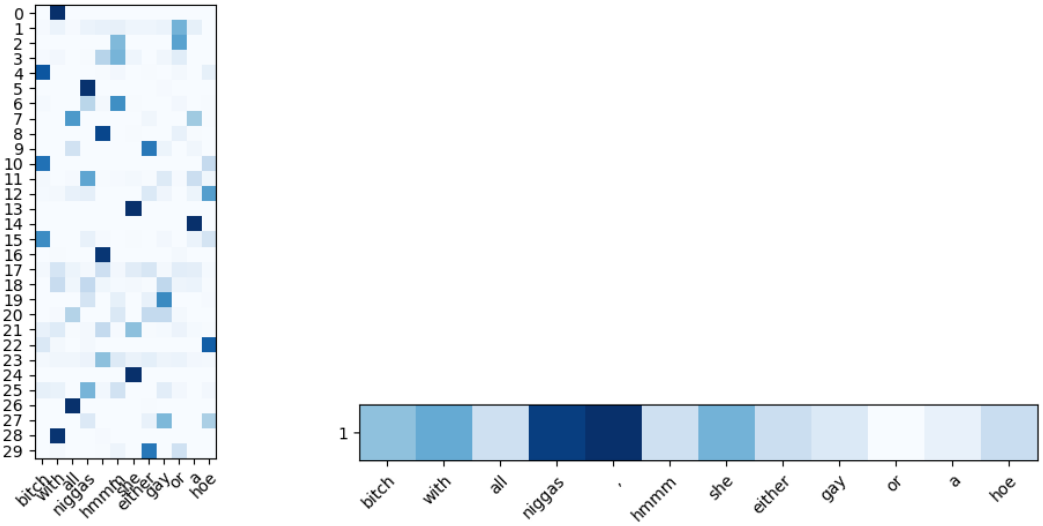
Z dôvodu redundantných údajov sa rôzne hlavy mechanizmu pozornosti zameriavali na rovnaké dáta. Odporúčaným riešením je využitie Frobeniusovej normy matice (viď rovnica 6.3) s rozmerom $r \times c$.

$$\|M\|_F^2 = \sqrt{\sum_{r=1}^{r}\sum_{c=1}^{c}|m_{rc}|^2}^2 \tag{6.3}$$

$$P = \|AA^T - I\|_F^2 \tag{6.4}$$



Obr. 6.2: Vizualizácia pozornosti bez aplikovania penalizácie



Obr. 6.3: Vizualizácia pozornosti s aplikovaním penalizácie

6.4.5 BERT klasifikátor

Model (viď obrázok 5.7) využíva predtrénovaný BERT model implementovaný Hugging Face. Tokenizér prevedie vstup na malé písmená a rozdelí na základe bielych znakov. Následne je na tokeny aplikovaný WordPiece [33]. Ten ich rozdelí na najčastejšie znakové n-gramy. Dosiahne sa dobrá reprezentácia slov s primeranou veľkosťou slovníka 30000.

BERT transformer prijíma na vstupe vektorovú reprezentáciu tokenov E_t, označenie segmentu E_s, ktoré označuje, ktoré slová patria do jednotlivých dokumentov (pri klasifikačných úlohách sa používa len jeden dokument) a označenie masky E_p, ktoré označuje tokeny vstupu od výplňových tokenov. Trojicu trasnformátor spracuje a klasifikácia prebieha na výstupe posledných hláv pozornosti.

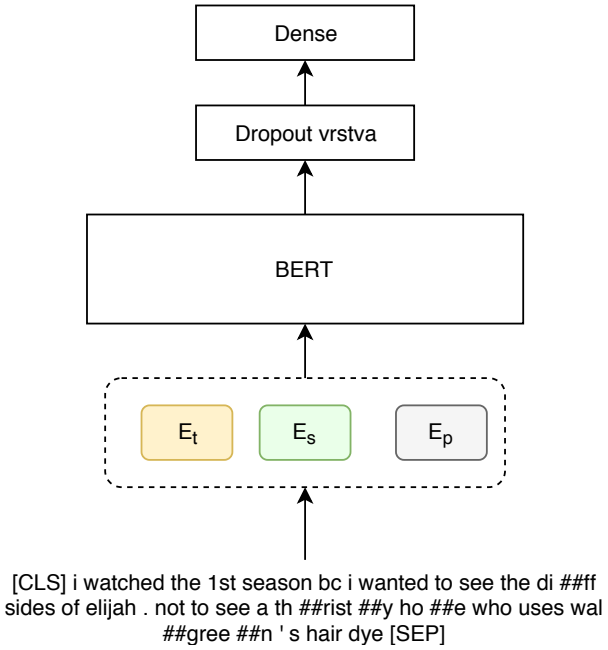
Ako transformer bol zvolený *"bert-base-uncased"* s konfiguračnými parametrami:

- 12 vrstiev pozornosti,
- veľkosť jednotky skrytého stavu je 768,
- 12 hláv pozornosti,
- a veľkosťou trénovacej skupiny 32 príkladov.

Okamžitý fine-tuning BERT modelu nedosiahol dobré F1 skóre. Preto bolo učenie BERT modelu pre prvých 10 epoch pozastavené, učila sa klasifikačná vrstva.

	#Θ	**Acc**	$\mathbf{F1_{macro}}$	$\mathbf{F1_{HS}}$	$\mathbf{F1_{OL}}$	$\mathbf{F1_{N}}$
$\text{BERT}_{\text{BERT}}$	109M	0.873	0.761	0.347	0.930	0.847

Tabuľka 6.6: Dosiahnuté výsledky na testovacej dátovej sade



Obr. 6.4: Architektúra BERT klasifikátoru

Záver

V mojej práci som sa venovala doterajším riešeniam detekcie nenávistného a ofenzívneho jazyka. Porovnávala som dostupné dátové sady vhodné pre trénovanie modelov. Načrtla som problematiku definície a anotácie ofenzívneho jazyka a nenávistných prejavov a vybrala vhodné prístupy na riešenie ich detekcie. Tieto metódy som implementovala a ich výsledky som vyhodnotila.

Každá zo skúmaných metód má nedostatky v určitých prípadoch. Síce sa mi nepodarilo prekonať najlepšie dosiahnuté výsledky, ale napriek tomu je moja práca vhodná k celkovému porovnaniu súčasných metód. Jednotlivé modely som implementovala na základe predchádzajúcich prác z oblasti klasifikácie a spracovania textu. Experimentálne som ich úspešnosť vyhodnotila nad dátovou sadou. Výsledky metód založených na hlbokom učení sa nepreukázali lepšími ako jednoduchšie metódy založené na štatistickej analýze. V budúcnosti sa môžu výsledky zmeniť, pretože užívatelia internetu nachádzajú nové spôsoby ako maskovať nenávistný prejav.

Myslím si, že problematickosť tohto fenoménu a jeho detekcie každým dňom rastie a nutnosť okamžitého riešenia čoraz akútnejšia. Sociálne siete sa už teraz snažia o automatizovanú detekciu, ktorá ale častokrát vyžaduje kontrolu ľudskými zdrojmi. Preto by som sa chcela v budúcnosti tomuto fenoménu naďalej venovať a detekciu zlepšiť.

Implementované modely sú prístupné na adrese `https://github.com/betsst/hate-speech-detection-BP` pod licenciou MIT.

Literatúra

[1] STATE'S ANTI-PROFANITY LAW UNCONSTITUTIONAL RULES SUPERIOR COURT JUDGE. [Online; navštíveno 11.05.2019].
URL https://www.aclu.org/news/states-anti-profanity-law-unconstitutional-rules-superior-court-judge

[2] Badjatiya, P.; Gupta, S.; Gupta, M.; aj.: Deep Learning for Hate Speech Detection in Tweets. *CoRR*, ročník abs/1706.00188, 2017, 1706.00188.
URL http://arxiv.org/abs/1706.00188

[3] Bishop, C. M.: *Pattern recognition and machine learning*. Information science and statistics, New York: Springer Science+Business Media, 2006, ISBN 0-387-31073-8.

[4] Burnap, P.; Williams, M. L.: Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making. *Policy & Internet*, ročník 7, č. 2, 2015: s. 223–242, doi:10.1002/poi3.85, https://onlinelibrary.wiley.com/doi/pdf/10.1002/poi3.85.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/poi3.85

[5] Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; aj.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR*, ročník abs/1406.1078, 2014, 1406.1078.
URL http://arxiv.org/abs/1406.1078

[6] Cho, K.; Van Merriënboer, B.; Gulcehre, C.; aj.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Davidson, T.; Warmsley, D.; Macy, M.; aj.: Automated Hate Speech Detection and the Problem of Offensive Language. *arXiv.org*, 2017.
URL http://search.proquest.com/docview/2074118430/

[8] Devlin, J.; Chang, M.; Lee, K.; aj.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, ročník abs/1810.04805, 2018, 1810.04805.
URL http://arxiv.org/abs/1810.04805

[9] Firth, J. R.: A synopsis of linguistic theory 1930-55. ročník 1952-59, 1957: s. 1–32.

[10] Fortuna, P.; Nunes, S.: A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, ročník 51, č. 4, 2018: s. 1–30, ISSN 1557-7341.

[11] Hanley, J. A.; McNeil, B. J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, ročník 143, č. 1, 1982: s. 29–36.

[12] Harris, Z. S.: Distributional structure. *Word*, ročník 10, č. 2-3, 1954: s. 146–162.

[13] Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; aj.: Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, ročník abs/1207.0580, 2012, 1207.0580.
URL http://arxiv.org/abs/1207.0580

[14] Hochreiter, S.; Schmidhuber, J.: Long short-term memory. *Neural computation*, ročník 9, č. 8, 1997: s. 1735–1780.

[15] Howard, J.; Ruder, S.: Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ročník 1, 2018, s. 328–339.

[16] Jay, T.: The Utility and Ubiquity of Taboo Words. *Perspectives on Psychological Science*, ročník 4, č. 2, 2009: s. 153–161, doi:10.1111/j.1745-6924.2009.01115.x, pMID: 26158942,
https://www.psychologicalscience.org/journals/pps/4_2_inpress/Jay.pdf.
URL https://doi.org/10.1111/j.1745-6924.2009.01115.x

[17] Jurafsky, D.; Martin, J. H.: *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009, ISBN 0131873210.

[18] Kim, Y.: Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2014, s. 1746–1751, doi:10.3115/v1/D14-1181.
URL http://aclweb.org/anthology/D14-1181

[19] Kincaid, J. P.; Fishburne Jr, R. P.; Rogers, R. L.; aj.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.

[20] Klubička, F.; Fernández, R.: Examining a hate speech corpus for hate speech detection and popularity prediction. *arXiv.org*, 2018.
URL http://search.proquest.com/docview/2073288106/

[21] Manning, C. D.; Raghavan, P.; Schütze, H.: Scoring, term weighting and the vector space model. *Introduction to information retrieval*, ročník 100, 2008: s. 2–4.

[22] Merity, S.: The WikiText Long Term Dependency Language Modeling Dataset. [Online; navštíveno 27.01.2019].
URL https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/

[23] Merity, S.; Keskar, N. S.; Socher, R.: Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*, 2017.

[24] Mishra, A.: Metrics to Evaluate your Machine Learning Algorithm. [Online; navštívené 28.12.2018].
URL https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

[25] Nguyen, M.: *Illustrated Guide to LSTM's and GRU's: A step by step explanation.* [Online; navštíveno 12.03.2019].
URL https://towardsdatascience.com/illustrated-guide-to-lstms-and-grus-a-step-by-step-explanation-44e9eb85bf21

[26] Nobata, C.; Tetreault, J.; Thomas, A.; aj.: Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, International World Wide Web Conferences Steering Committee, 2016, s. 145–153.

[27] Razavi, A. H.; Inkpen, D.; Uritsky, S.; aj.: Offensive Language Detection Using Multi-level Classification. In *Advances in Artificial Intelligence*, editace A. Farzindar; V. Kešelj, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-13059-5, s. 16–27.

[28] Ross, B.; Rist, M.; Carbonell, G.; aj.: Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. In *Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication*, *Bochumer Linguistische Arbeitsberichte*, ročník 17, editace M. Beißwenger; M. Wojatzki; T. Zesch, Bochum, sep 2016, s. 6–9.

[29] Sasaki, Y.; aj.: The truth of the F-measure. *Teach Tutor mater*, ročník 1, č. 5, 2007: s. 1–5.

[30] Volokh, E.: STATE'S ANTI-PROFANITY LAW UNCONSTITUTIONAL RULES SUPERIOR COURT JUDGE. [Online; navštíveno 11.05.2019].
URL
https://www.washingtonpost.com/news/volokh-conspiracy/wp/2015/05/07/no-theres-no-hate-speech-exception-to-the-first-amendment/?noredirect=on&utm_term=.ce0080ba7f1e

[31] Waseem, Z.: Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, Austin, Texas: Association for Computational Linguistics, November 2016, s. 138–142.
URL http://aclweb.org/anthology/W16-5618

[32] Waseem, Z.; Hovy, D.: Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, San Diego, California: Association for Computational Linguistics, June 2016, s. 88–93.
URL http://www.aclweb.org/anthology/N16-2013

[33] Wu, Y.; Schuster, M.; Chen, Z.; aj.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, ročník abs/1609.08144, 2016, 1609.08144.
URL http://arxiv.org/abs/1609.08144

[34] Wulczyn, E.; Thain, N.; Dixon, L.: Ex Machina: Personal Attacks Seen at Scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, ISBN 978-1-4503-4913-0, s. 1391–1399, doi:10.1145/3038912.3052591.
URL https://doi.org/10.1145/3038912.3052591

[35] Yang, Z.; Yang, D.; Dyer, C.; aj.: Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, s. 1480–1489.

[36] Yosowich, M.: Canada's weirdest laws: it's illegal to swear in a Toronto park. [Online; navštíveno 11.05.2019].
URL https://legalblogs.findlaw.ca/uncommon-law/canadas-weirdest-laws-its-illegal-to-swear-in-a-toronto-park-1004/

[37] Zhang, X.; Zhao, J. J.; LeCun, Y.: Character-level Convolutional Networks for Text Classification. *CoRR*, ročník abs/1509.01626, 2015, 1509.01626.
URL http://arxiv.org/abs/1509.01626

Príloha A

Obsah priloženého pamäťového média

Na priloženom CD sa nachádza:

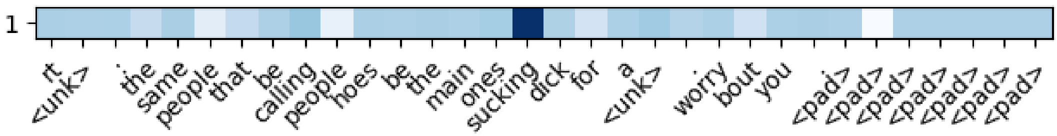
- zdrojový kód textu bakalárskej práce napísany v jazyku LaTeX,
- práca vo formáte *.pdf*,
- zdrojové kódy implementovaných modelov

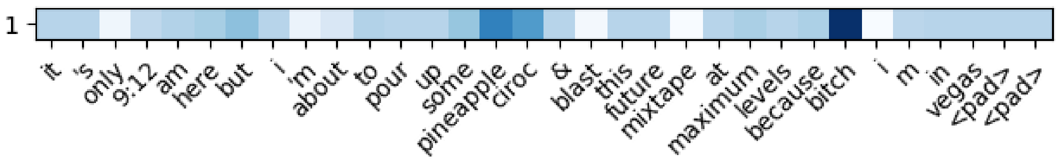
Príloha B

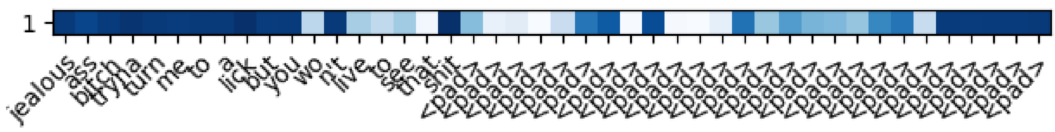
Vybrané vizualizácie pozornosti modelu

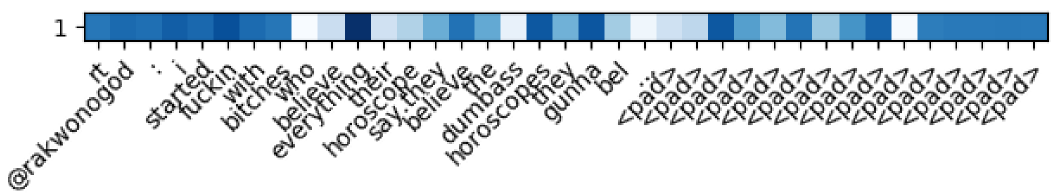
Príloha obsahuje vybrané vizualizácie pozornosti na príkladoch z testovacej dátovej sady.

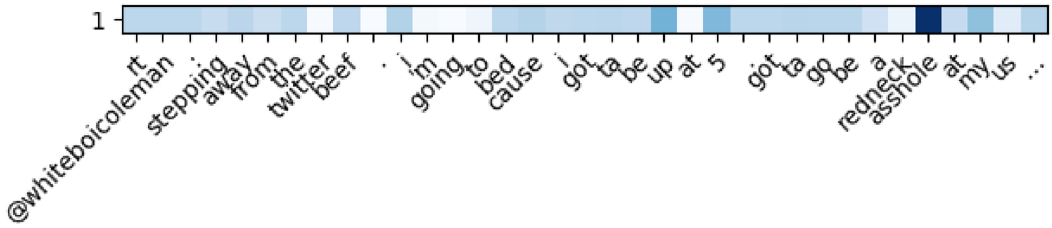
Vizualizovaná pozornosť s využitou penalizáciou:

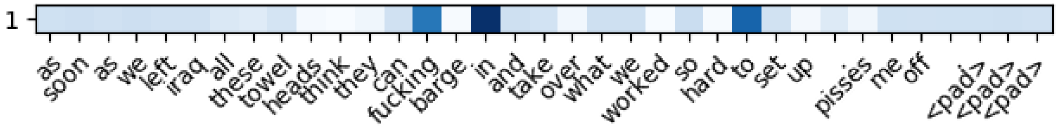


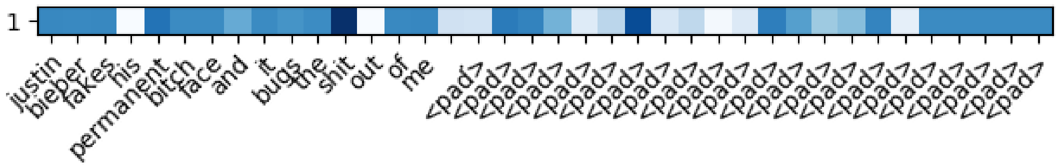


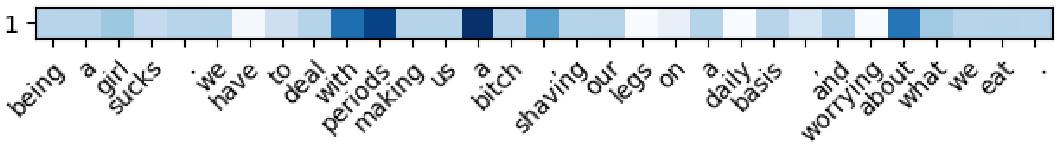




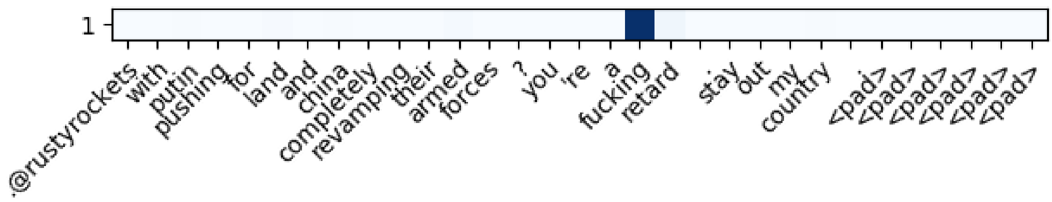


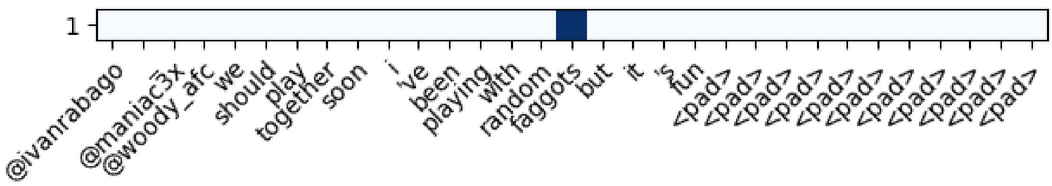






Vizualizovaná pozornosť bez využitej penalizácie:






1
only
things
that
calm
me
down
pussy
and
some
cali
tree
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>
<pad>