Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Experimenty s moderními modely neuronových sítí pro vícetřídní klasifikaci českého textu

Plzeň 2023 Daniel Cífka

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Daniel CÍFKA**
Osobní číslo:	**A19B0021P**
Studijní program:	**B0613A140015 Informatika a výpočetní technika**
Specializace:	**Informatika**
Téma práce:	**Experimenty s moderními modely neuronových sítí pro vícetřídní klasifikaci českého textu**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prostudujte problematiku a dostupné datové sady pro vícetřídní klasifikaci textu primárně v českém jazyce.
2. Proveďte analýzu již existujících a implementovaných modelů neuronových sítí, které se v této oblasti používají, s důrazem na nejmodernější architektury.
3. Na základě předchozí analýzy vyberte vhodný model a navrhněte množinu experimentů. Experimenty budou probíhat v automatizovaném režimu a poskytovat výsledky v podobě vhodných metrik.
4. Navrženou množinu experimentů realizujte a na základě porovnání výše uvedených metrik zhodnoťte přínosy vybraného modelu.
5. Naměřené metriky zpracujte do formy tabulek, případně grafů a proveďte celkové kritické zhodnocení.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce

Vedoucí bakalářské práce: **Ing. Jiří Martínek, Ph.D.**
Nové technologie pro informační společnost

Datum zadání bakalářské práce: **3. října 2022**
Termín odevzdání bakalářské práce: **4. května 2023**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 25. října 2022

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 3. května 2023

Daniel Cífka

Poděkování

Rád bych tímto poděkoval panu Ing. Jiří Martínek, Ph.D. za cenné a užitečné rady, věcné připomínky a velikou vstřícnost při konzultacích a celkovém vypracování své bakalářské práce.

Abstract

The bachelor's thesis deals with the automatic multi-labels classification of Czech written text with the help of Transformer neural network models, primarily the *small-e-czech* model. The main task of the work is to perform experiments with the aim of comparing the precision of multi-labels text classification on modern models that have knowledge of the Czech language obtained by previous training on a large amount of Czech text. The obtained results are compared to the results reported in available published articles and are objectively evaluated.

Abstrakt

Bakalářská práce se zabývá automatickou vícetřídní klasifikací česky psaného textu za pomoci modelů neuronových sítí typu Transformer, primárně model *small-e-czech*. Hlavním úkolem práce je provést experimenty s cílem porovnání přesnosti vícetřídní klasifikace textu na moderních modelech, které mají znalost českého jazyka získanou předchozím trénováním na velkém množství českého textu. Získané výsledky jsou porovnány s výsledky uváděné v dostupných publikovaných článcích a jsou objektivně zhodnoceny.

Obsah

1 Úvod

Kategorizace či klasifikace textu je jednou ze základních úloh umělé inteligence v oblasti zpracování přirozeného jazyka (angl. Natural Language Processing – NLP). Jedná se nejčastěji o kolekce dokumentů a jiných materiálů (např. novinové články, sportovní a jiné zprávy, ekonomické analýzy apod.).

Základní úloha klasifikace textu je, že se přiřadí textu jedna třída z předem definovaného seznamu (např. kultura, sport nebo ekonomie). Nicméně se často stává, že dokument náleží do více než jedné klasifikační třídy (např. zahraniční/domácí kultura, zahraniční/domácí sport atd.). Vzniká tak potřeba zachytit i další úrovně kategorií, a tak již mluvíme o vícetřídní klasifikaci, kdy se textu přiřadí nikoliv jedna, ale několik tříd najednou.

S rostoucím množstvím textů je nutné použít spolehlivá a automatizovaná řešení, které v maximální možné míře ušetří manuální práci. Mezi tato řešení patří převážně metody strojového učení, které poskytují velké množství klasifikačních algoritmů. V této práci se budu zaměřovat pouze na jednu podoblast strojového učení a tou jsou neuronové sítě, které jsou v poslední době zcela dominantní a bezesporu nejpoužívanější.

Hlavním účelem této práce je vytvořit sadu automatizovaných experimentů s modelem neuronové sítě, vycházející z modelu typu *Transformer*, s cílem provést vícetřídní klasifikaci textu v českém jazyce. Datová sada, na které se provádějí experimenty je datová sada *Czech Text Document Corpus v 2.0* dostupný na URL[1]

V teoretické části práce bude stručně popsán vývoj modelů používaných v NLP. Poté budou uvedeny nejčastější metody předzpracování textu a reprezentace dokumentu. Po nezbytném popisu datové sady následuje praktická část: popis experimentů a prezentace výsledků včetně porovnání s dostupnými publikacemi.

V neposlední řádě bude provedena analýza chyb, které předpokládáme při klasifikaci, zejména pro méně četné třídy, u kterých existuje podezření, že se je model nenaučí.

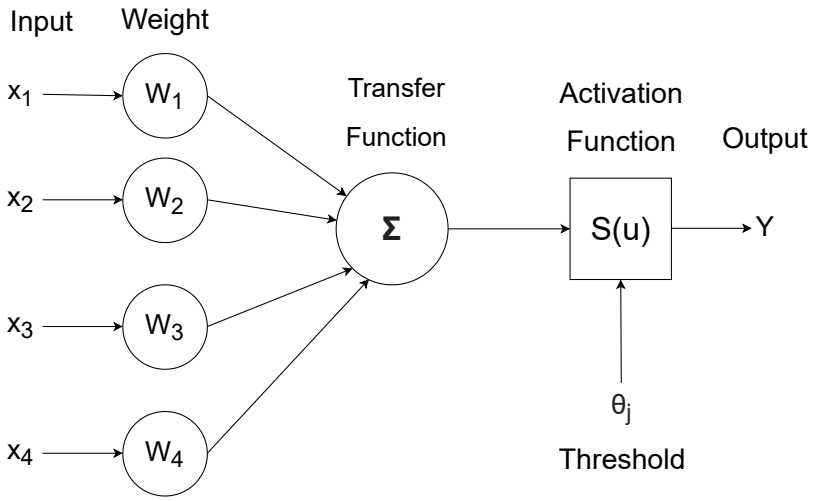
[1] `http://ctdc.kiv.zcu.cz/`

2 Umělá neuronová síť

Umělá neuronová síť (angl. Neural Network – NN) je výpočetní model, který se v poslední letech nejvíce využívá v oblasti umělé inteligence a strojového učení.

Nejčastější forma použití neuronové sítě je tzv. učení s učitelem (angl. Supervised Learning). Je to proces při kterém, jsou k dispozici označená data (vstupní hodnoty společně s očekávaným výstupem). Hlavní využití je při klasifikaci. Příkladem učení bez učitele (angl. Unsupervised Learning) je například shlukování nebo jiná metoda, při kterém nejsou k dispozici označená data. Cílem je pak odhalení skupin podobných vzorků, což může být vhodné např. pro účely vizualizace [8].

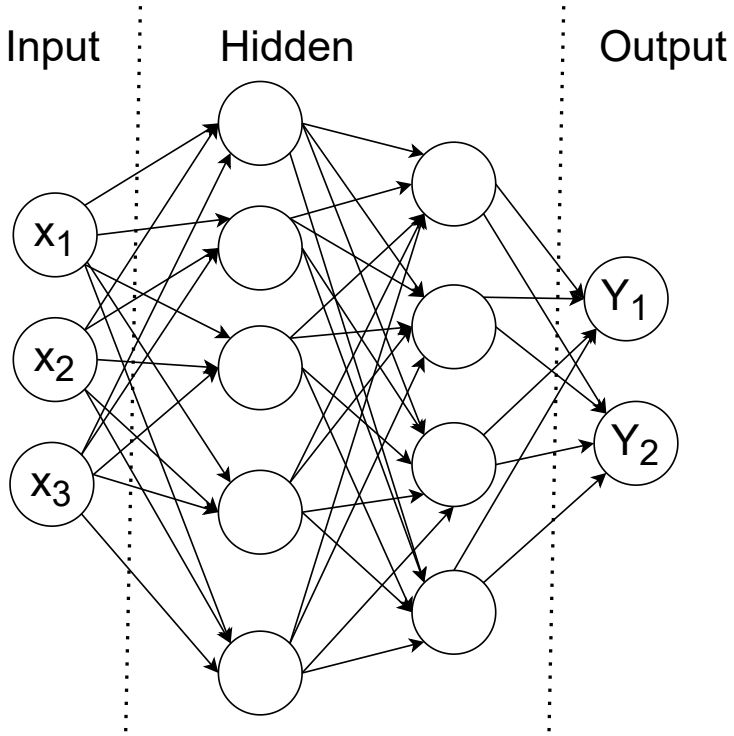
Koncept modelu neuronové sítě je založen na biologickém modelu fungování lidského mozku. Základní jednotkou je umělý neuron. Tyto neurony jsou uspořádané ve vrstvách a vzájemně propojeny spoji ohodnocenými vahami. Takováto propojení a schopnost tyto váhy adaptovat (učit se) na základě trénovacích vzorů je hlavní podstata tohoto algoritmu [1]. Každý neuron může obsahovat více vstupních signálů, ale vždy má pouze jeden výstupní signál (viz McCulloch-Pittsův [26] model na obrázku 4.1).



Obrázek 2.1: Model umělého neuronu

Tento model obsahuje následující parametry:

- x_1, x_2, ..., x_N – vstupy neuronu,
- w_1, w_2, ..., w_N – synaptické váhy,
- Θ – je práh,
- Potenciál neuronu: $u = \sum_{i=1}^{N}(w_i x_i) - \Theta$,
- $S(u)$ – aktivační funkce,
- Y – výstupní hodnota neuronu.



Obrázek 2.2: Příklad modelu Vícevrstvého perceptronu

Podle způsobů propojení jednotlivých neuronů rozdělujeme sítě do různých architektur. Dle počtu vrstev dělíme na jednovrstvé, vícevrstvé, případně rekurentní architektury, které jsou velmi vhodné pro zpracování sekvencí dat.

2.1 Učení neuronové sítě

Klíčovou věcí pro trénování a učení neuronové sítě je tzv. ztrátová funkce E (angl. loss function). Účelem trénování je minimalizace této funkce a dosáhnutí optima na základě trénovacích dat. Funkce vypočítává chybu predikce sítě oproti odpovědím učitele (angl. ground truth). Podoba ztrátové funkce (stejně jako aktivační funkce v poslední vrstvě) závisí na řešené úloze. [8]. Pravděpodobně nejčastěji používaná ztrátová funkce pro úlohu klasifikace do K tříd je tzv. *cross-entropy loss function* (viz následující vzorec, kde p je predikce sítě a y je očekávaný výstup, neboli odpověď učitele).

$$E(p, y) = -\sum_{i=1}^{K} y_i \; log(p_i) \tag{2.1}$$

Během trénování je cílem nastavit váhy sítě tak, aby se minimalizovala chyba na trénovacích datech. Toho je docíleno pomocí algoritmu *backpropagation* resp. jeho tří fází:

1. dopředné šíření (angl. feed-forward pass);
2. zpětná propagace chyby;
3. aktualizace vah (angl. weights update).

Nejčastěji se pro implementaci používá metoda gradientního sestupu, což je algoritmus iterativní optimalizace funkce. Pomocí zpětné propagace chyby jsou vypočteny tzv. gradienty ztrátové funkce vzhledem k vahám sítě ∇E. Vzorec pro aktualizaci vah pak vypadá následovně:

$$w^{(t+1)} = w^{(t)} - \alpha \nabla E w^{(t)} \tag{2.2}$$

kde α je tzv. *learning rate* (míra učení), která určuje míru rychlost konvergence. Obvykle se používá α v roszahu 0.01 – 0.0001, v závislosti na úloze a architektuře sítě. Pro detailnější popis algoritmu *backpropagation* viz [23] nebo [8].

2.2 Hluboké neuronové sítě

Hluboká neuronová síť je typicky mnohovrstevnatá a v současné době nejpoužívanější. Jako příklady lze uvést např. konvoluční neuronové sítě pro obrázky nebo architektury LSTM a Transformer pro zpracování textů (viz níže).

Tento typ sítě využívá techniky hlubokého učení (Deep learning = DL) [15], kdy ke svému učení je využit obrovský objem dat často v nestruktutované podobě, tzn. bez manuálně navržených příznaků (angl. features). Vstupem do hluboké sítě jsou často přímo obrázky (resp. matice pixelů) či v případě textu sekvence tokenů.

Výhodou a vlastností hlubokých neuronových sítí je víceméně automatická tvorba příznaků, které si síť umí "sama odvodit" [23]. Učení hlubokých sítí probíhá rovněž pomocí algoritmu backpropagation, nicméně u nich byly pozorovány problémy s učením, především díky tzv. vanishing gradient problem [23]. Toto mělo za následek, že zpětné šíření chyby a na to navazující úprava parametrů sítě byly problematické. Nicméně vývoj nových architektur (např. LSTM či Transformer) a masivní používání aktivační funkce ReLU [2] tento problém potlačily.

2.3 Využití neuronových sítí

V dnešní době mají neuronové sítě velké užití a to hlavně v následujících úlohách:

- klasifikace textu,
- analýza sociálních sítí např. polarita sentimentu,
- klasifikace recenzí a reakce spotřebitelů,
- medicínské aplikace, např. lékařská diagnostika,
- rozpoznaní poruch strojů a výpočetních systému,
- různé predikce, samoobslužné systémy a další.

3 Klasifikace textu

Tato bakalářská práce se věnuje experimentům s modely neuronových sítí pro klasifikaci textu v českém jazyce, což je úloha NLP. V této kapitole popíši vývoj základních modelů používané v NLP oblasti. Poté budou následovat části týkající se přímo úlohy klasifikace textu, především předzpracování textu a jak text vlastně reprezentovat.

3.1 Vývoj modelů NLP

V této části popíši stručně vývoj modelů a architektur, které byly a nyní jsou použity v NLP. Tuto část začnu rekurentními neuronovými sítěmi se zaměřením na architektury *Long short-term memory* (LSTM) [14] a *Gated recurrent unit* (GRU) [18]. Následně se přejde k modelům typu *Transformer* [31]. V poslední části textu bude zmínka o modelu BERT [11], který je dnes jeden z nejznámějších nepoužívanějších modelů v NLP.

3.1.1 Rekurentní neuronová síť

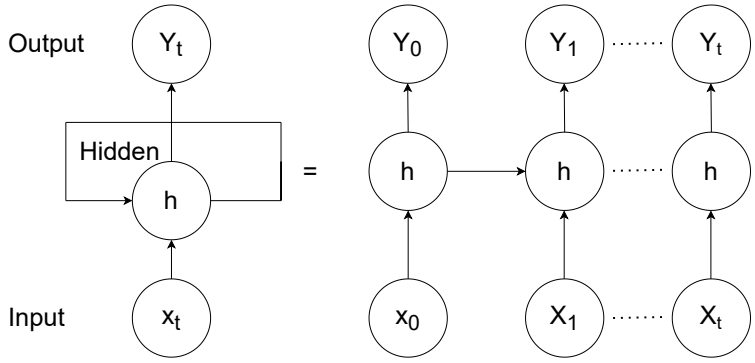
Jedná se o typ neuronové sítě, kde výstup předchozího kroku je vstupem do aktuálně zpracovávaného kroku. V tradičním typu neuronové sítě, jsou všechny vstupy a výstupy na sobě nezávislé, ale v případě predikce textu, je potřeba znát předchozí slovo, aby bylo možné predikovat následující. Z tohoto důvodu vznikla rekurentní neuronová síť (RNN), která tento problém řeší ve své skryté vrstvě. Hlavní a nejdůležitější vlastností je tzv. skrytý stav (angl. hidden state), který si uchovává předchozí informace o sekvenci.

Výpočet aktuálního stavu

$$h_t = f(h_{t-1}, x_t),$$

kde h_t = aktuální stav, h_{t-1} = předchozí stav a x_t = vstupní stav.

Model rekurentní neuronové sítě je uveden na obrázku 3.1.



Obrázek 3.1: Ukázka rekurentní sítě

3.1.2 Long short-term memory (LSTM)

Model LSTM byl navržen autory Hochreiterem a Schmidhuberem [14]. Hlavní výhodou této sítě je učení dlouhodobých závislostí. Standardní rekurentní síť měla tendence "zapomínat" obzvlášť v případech kdy byly zpracovávány dlouhé sekvence. Díky své architektuře a používání tzv. bran (angl. gates) je mnohem méně náchylná na problém mizejícího gradientu [24]. Model je vyobrazen na obrázku 3.2 a je popsán následujícími rovnicemi 3.1 až 3.6.

$$f_t = \sigma(x_t * U_f + h_{t-1} * W_f) \tag{3.1}$$

$$\overline{c}_t = \tanh(x_t * U_c + h_{t-1} * W_C) \tag{3.2}$$

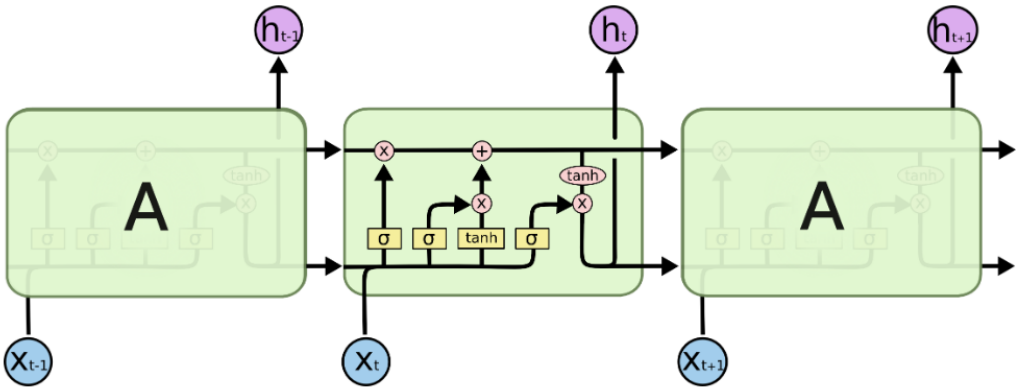
$$i_t = \sigma(x_t * U_i + h_{t-1} * W_i) \tag{3.3}$$

$$o_t = \sigma(x_t * U_o + h_{t-1} * W_o) \tag{3.4}$$

$$c_t = f_t * c_{t-1} + i_t * \overline{c}_t \tag{3.5}$$

$$h_t = O_t * \tanh(c_t) \tag{3.6}$$

f_t –> Forget Gate vector,
$\overline{c}_t$ –> Candidate layer vector,
i_t –> Input Gate vector,
o_t –> Output Gate vector,
h_t –> Hidden state vector,
c_t –> Memory state vector,
W, U –>weight vectors for forget gate,
x_t –> Input vector.



Obrázek 3.2: LSTM architektura [24]

3.1.3 Gated recurrent unit (GRU)

Jedná o jeho zjednodušenou modifikaci modelu LSTM, která má za příčinu mimo jiné rychlejší výpočet [18]. Autorem modelu byl Kyunghyun Cho a kol., který v roce 2014 tento model publikoval. Zjednodušení modelu spočívá v tom, že model GRU využívá pouze dvě brány aktualizační 3.7 a resetovací 3.8. Brány, které jsou reprezentovány vektorem, lze interpretovat jako filtry, které rozhodují o důležitosti vstupu.

Matematicky lze tento model popsat:

$$z_t = \sigma_g(W^{(z)}x_t + U^{(z)}h_{t-1} + b_z) \tag{3.7}$$

$$r_t = \sigma_g(w^{(r)}x_t + U^{(r)}h_{t-1} + b_r) \tag{3.8}$$

$$\hat{h_t} = \tanh(U_h(r_t \odot h_{t-1})W_h x_t + b_h) \tag{3.9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h_t} \tag{3.10}$$

x_t –> Input vector,
$\hat{h}_t$ –> Candidate activation vector 3.9,
h_t –> Output vector 3.10,
z_t –> update gate vector 3.7,
r_t –> reset gate vector 3.8,
W, U, b –>parameter matrices and vector.

3.1.4 Mechanismus Attention a Self-Attention

Vedle úlohy klasifikace textu se rovněž zkoumaly možnosti strojového překladu, tedy úlohy typu sequence-to-sequence (seq2seq). Tyto úlohy nejčastěji fungovaly na architektuře encoder-decoder, kdy kódovací část (encoder) vytvořila latentní reprezentaci (např. vektor či matice o fixní dimenzi), kterou poté četl dekodér a na základě toho vytvořil výstupní sekvenci.

K tomu, aby se vylepšil výkon takovýchto modelů (především pro strojový překlad) byl navržen mechanismus attention [6]. Myšlenka byla využívat nejvíce relevantní části vstupní sekvence a vytvořit váhování takovým způsobem, aby bylo možné zvládat i dlouhé sekvence. Konečným výstupem mechanismu attention je kontextový vektor, který je vstupem do dekodéru. Pro detailnější popis viz [6, 7].

Další fází vylepšení bylo tzv. Self-Attention, která umožňuje interakci mezi jedontlivými vstupy. Díky tomu lze vytvořit lepší reprezentaci vstupu a také není třeba počítat s dekodérem. Toto se stalo základem modelu typu Transformer [31], jehož vlastností je absence jakýchkoliv rekurentních částí.

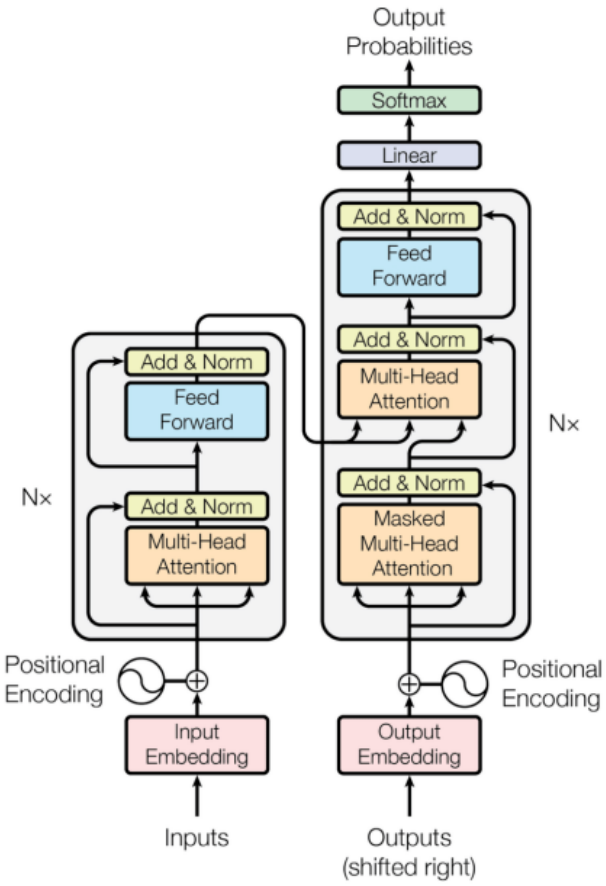
3.1.5 Transformer

Hlavní komponentou tohoto modelu je multi-head self-attention [31]. Tento mechanismus používá "více hlav" k zachycení více kontextů a celkově tak vytvořit lepší reprezentaci vstupu. Vstupní sekvence je transformována na vstupní vektory prostřednictvím tzv. positionval encoding. Toto zajistí dodání informace o relativní či absolutní pozici jednotlivých tokenů, protože zde není žádná rekurentní část zpracovávající sekvenci postupně. Díky tomu lze mimo jiné umožnit paralelizaci a zkrátit tak výpočetní čas pro trénování modelu a zpracovávat velké datové sady.

Modely založené na této architektuře víceméně ovládly současný vývoj a výzkum a vznikají stále větší a větší modely se stovky miliony až miliardami parametrů. Myšlenka je mít co nejobecnější předtrénovaný model (např. z dostupných textů z Wikipedie) a model posléze vyladit na konkrétní typ úlohy.

3.1.6 BERT

Model BERT ("Bidirectional Encoder Representations from Transformers"), založený na architektuře Transformer byl publikován v roce 2018 výzkumníky z Google. Model vylepšil výsledky na 11 NLP úlohách, včetně klasifikace textu. Model je vyobrazen na obrázku 3.4. Model se se před učí na velkém množství textu na úlohách *Masked Language Modeling* a *Next Sentence Prediction*, díky čemuž získá znalost o jazyku. Úloha maskovaného jazykového modelu spočívá v tom, že během tréninku je nějaký počet slov náhodně skryt a model je učen k tomu predikovat původní slova. Druhá úloha je určování jestli dvě věty jdou správně po sobě či nikoliv. Toto je predikováno na základě klasifikační token (tzv. CLS token v obrázku 3.4 označen písmenem C vlevo nahoře).



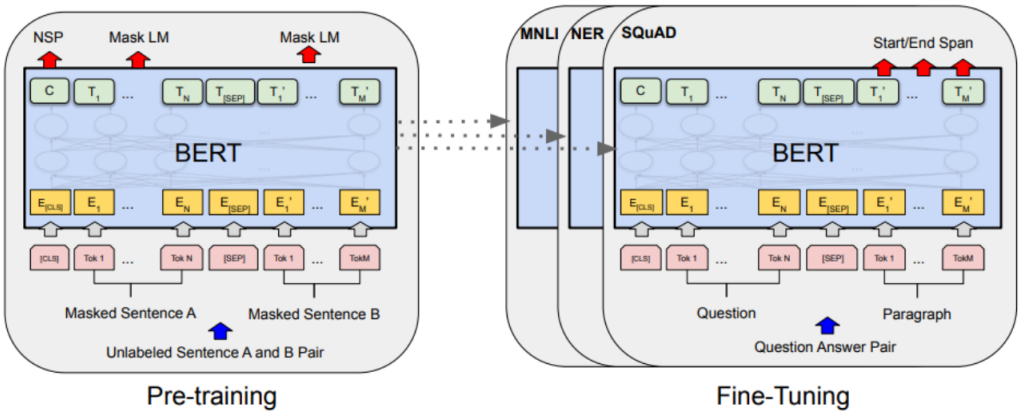
Obrázek 3.3: Transformer model [31]

Takto naučený před trénovaný model lze pak využít v celé řadě úloh, stačí pouze načíst model s před trénovanými váhami a použít, případně vyladit na požadovanou úlohu.

3.1.7 Další modely

Po úspěchu modelu BERT byly vyvinuty další podobné modely např. RoBERTa [21] nebo ELECTRA [10], ale i další, které jsou postaveny na podobném principu.

Zmíním zde model ELECTRA, který je použit pro experimenty v této bakalářské práci. Jelikož model BERT obsahuje velké množství parametrů (ve verzi Large 340 milionů), může být nesnadné natrénovat podobný model od začátku. Proto vznikaly další modely, které měly méně parametrů, ale zároveň se snažily zachovat podobnou úspěšnost úloh. Model ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) z roku 2020, která se před trénovala podobným způsobem jako BERT



Obrázek 3.4: BERT model [11], vlevo model ve fázi předtrénování; vpravo možné aplikace a model ve fázi vyladění

model. Modelu byly předkládány věty, v nichž byly náhodně nahrazovány některé tokeny za jiné. Úlohou bylo rozhodnutí, zda daný token je původní či nikoliv. Takovýmto způsobem trénovaný model poté získal znalost o jazykovém modelu a šlo ho použít pro NLP úlohy. Zejména čas ladění na požadovanou úlohu byl menší než v případě modelu BERT.

Varianta modelu ELECTRA *small-e-czech* je použita pro experimenty v této bakalářské práci. Tento před trénovaný model vydaný firmou seznam.cz [4, 17] v roce 2021 je založený na variantě ELECTRA small a obsahuje 14 milionů parametrů. Model byl před trénován na 250 GB českého textu.

3.2 Předzpracování textu

Předzpracování je první fází celého procesu klasifikace textu. Každý text se před samotnou klasifikací nejdříve předzpracuje a upraví, aby výsledek samotné klasifikace dosahoval lepší klasifikačních hodnot. Metod předzpracování textu je mnoho, já jsem se rozhodl zde zmínit pouze 3 tyto metody a to

- Tokenizace,
- Stemming,
- Lemmatizace.

Kromě výše uvedených se ještě často používá odstraňování interpunkčních znamének a také tzv. stop slov (angl. stop-words). Jedná se o slova, která nenesou žádnou důležitou informaci, např. předložky, spojky, případně

v cizích jazycích určitý či neurčitý člen. Součástí moderních modelů typu *Transformer* bývá tokenizer, který vytvoří tokeny.

3.2.1 Tokenizace

Jedná se o metody předzpracování textu, kdy je vytvořen slovník a dochází k rozdělení textu na malé části. Malé části jsou myšleny slova nebo jejich části. Těmto rozděleným částí se odborně říká tokeny. Každému tokenu je přiřazeno číslo, které slouží zároveň jako index do slovníku, případně do některé z vrstev neuronové sítě (např. embedding layer) [9]. Vstupem do klasifikátoru je pak seznam (nebo přesněji vektor) těchto čísel.

3.2.2 Stemming

Metoda stemmingu odstraňuje koncovky slov. Příkladem může být zjednodušení slov: 'Programy', 'Programu', a 'Programů' na 'Program'. Hlavním problémem těchto metod je, že vzniklá transformace nemusí být validním slovem v daném jazyce [13, 28]. **Příklady stemmerů:**

- Porter stemmer,
- Snowball stemmer,
- Lacaster stemmer,
- Regexp stemmer.

3.2.3 Lemmatizace

Poslední z vybraných metod předzpracování textu je Lemmatizace. Lemmatizace má velmi podobný princip jako stemming s tím rozdílem, že lemmatizace transformuje slova textu do slovníkových tvarů (lemmat). Hlavním účelem lemmatizace je stejný jako o stemming a to eliminovat více tvarů slov[13, 33]. Lemmatizace nemá žádné specializované algoritmy, ale lze tohoto typ metody vytvořit vyhledáváním slov ve slovníku **lemmat**.

3.3 Reprezentace dokumentu

Učící se klasifikátor (např. neuronová síť) je matematický model, který neumí pracovat přímo s textovou reprezentací. Je tedy potřeba převést ho do nějaké číselné reprezentace (např. seznam tokenů), a tak vytvořit tzv. příznakové vektory (feature vector) s fixní dimenzí.

3.3.1 Bag of Words

Mluvíme zde o základním modelu reprezentace textových dokumentů. Základem je tvorba slovníku z dostupného textu. Máme tedy text reprezentovaný příznakovým vektorem, který má fixní dimenzi podle definovaného počtu slov ve slovníku a na pozici slova ve vektoru je obsaženo číslo 1, pokud se slovo v dokumentu vyskytuje nebo 0 pokud se nevyskytuje.

Pokud chceme zohlednit důležitost jednotlivých slov, tak lze do vektoru zakódovat např. četnost slova nebo hodnoty TF-IDF [27].

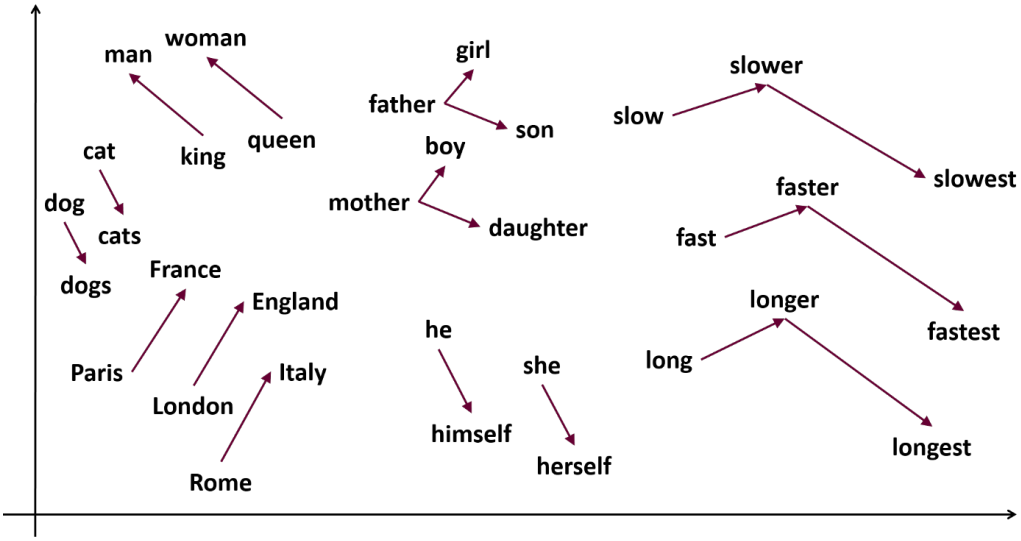
Při použití tohoto typu reprezentace ztrácíme informaci o pozici slova ve vstupním textu. Nelze tedy zjistit jaké slovo předcházelo nebo nasledovalo a ztrácí se tedy kontext textu.

3.3.2 Další metody reprezentace dokumentů

Mezi další metody reprezentace dokumentů můžeme zařadit i tokenizaci (viz výše), kdy je každý dokument reprezentován seznamem číselných tokenů, odpovídající indexu slova ve slovníku. Tento způsob je nejvíce využívaný v modelech typu Transformer. V neposlední řadě můžeme ještě zmínit word2vec model [16].

3.3.3 Word2Vec

Word2Vec je technika NLP, publikovaná v roce 2013 autorem Tomášem Mikolovem [16]. Tato technika využívá jednoduchého modelu neuronové sítě k učení slovních asociaci z velké datové sady textů. Síť se učí predikovat slovo na základě kontextu (okolní slova) nebo naopak. Po natrénovaní lze slovu přiřadit vektor, který má v sobě zakódované informace o kontextu (tj. významu slova), a tyto vektory mohou být použity jako příznaky pro klasifikaci. Dále lze také pomocí např. kosinové vzdálenosti najít významově podobná slova.



Obrázek 3.5: similarity word2vec [16]

3.4 Úloha klasifikace

Klasifikace je úloha, při které máme známý počet výstupních tříd a využívaný klasifikátor k určení konkrétní třídy nebo tříd. Jedná se zpravidla o učení s učitelem na dané trénovací sadě. Samotný klasifikační proces provádí natrénovaný klasifikátor, kde vstupem je vstupní vektor $\vec{x}$, obsahující příznakový popis klasifikovaného objektu a výstupem je identifikátor klasifikované třídy ω.

3.4.1 Binární klasifikace

Jedná se o klasifikaci do dvou tříd, které se navzájem vylučují (značené většinou 0 nebo 1). Využití tohoto typu klasifikátoru je celá řada detekcí: například v emailové klientu, kde rozdělujeme poštu na běžnou poštu a spam či např. v medicíně detekce positivity/negativity výskytu nemoci u pacientů.

3.4.2 Multi–Class klasifikace

Tento druh klasifikátoru vychází z předchozího binárního klasifikátoru, ale je pouze rozšířen o velikost výstupní množiny tříd, která není binární ale jedná se o vektor $\vec{v}$ o dimenzi k. Obdobně jako v binárním klasifikátoru, i zde se klasifikuje objekt pouze do jedné z k klasifikačních tříd.

Tento druh klasifikace je velmi rozšířen, používá se při klasifikaci textu či obrázků a zpravidla výstupní vektor odráží pravděpodobnost příslušnosti k dané třídě.

3.4.3 Multi–Label klasifikace

Poslední skupinou klasifikačních úloh, jsou úlohy typu multi–label, kde se jedná pouze o rozšíření předcházejících klasifikací o možnost zařazení klasifikovaného objektu do více klasifikačních tříd.

Multi–label klasifikaci lze řešit dvěma možnými postupy. První z postupů je tvorba binárního klasifikátoru pro každou klasifikační třídu. Druhým přístupem, častějších pro neuronové sítě, je prahování výstupní vrstvy. Neurony, jejichž aktivace je větší než daný práh (zpravidla 0.5) reprezentují predikovanou třídu. Tento typ klasifikátoru je využit v této práci, jelikož jeden novinový článek může obsahovat více možných témat (např. sport, sport domácí, hokej).

3.4.4 Vyhodnocení úspěšnosti Multi–label klasifikátoru

Tato práce se zabývá multi-label klasifikací, tudíž nyní popíši, jakým způsobem budeme měřit výkonnost klasifikátoru. Pro vyhodnocování multi-label klasifikace se obvykle používají metriky přesnost (Precision), úplnost (Recall) a F-míra (F-measure). Pro tyto metriky je nutné pro každý testovaný vzorek/vstup spočítat následující:

- TP (True positive) – Klasifikátor správně predikoval třídu, která měla být predikována,
- FP (False positive) – Klasifikátor nesprávně predikoval třídu, která neměla být predikována
- FN (False negative) – Klasifikátor nepredikoval třídu, která měla být predikována.

Pak lze spočítat výše uvedené metriky následujícím způsobem.

$$precision = \frac{TP}{TP + FP}, \tag{3.11}$$

$$recall = \frac{TP}{TP + FN}, \tag{3.12}$$

F–míra (F1) je pak harmonický průměr úplnosti a přesnosti.

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.13}$$

Pro multi-label klasifikace lze ještě použít alternativu k metrice *Accuracy* tzv. *Exact Match*, kdy testovací vzorek je správně pouze tehdy, pokud

neobsahuje ani jedno FP nebo FN. Výsledné skóre je pak poměr správně predikovaných vzorků ku všem.

Pro výpočet výkonu klasifikačních modelů využíváme dvou agregačních metod Macro F1 a Micro F1, pro podrobnější popis lze přečíst zde [3].

Macro F1 je nejjednodušší agregaci skóre F1. Každá třída ve výpočtu obsahuje stejnou váhu jako všechny ostatní třídy, a tedy mluvíme o neváženém průměru všech F1. U Micro F1 agregace má každý vzorek stejnou váhu, bez ohledu na to z jaké je třídy.

Z výše uvedených informací o *Macro* a *Micro* agregacích, lze říci že pro nevyvážený datový set bude vhodnější *Macro* hodnota F–míry, protože při použití *Micro* agregace budou mít více četnější třídy větší vliv na celkovou hodnotu F1 skóre a tedy skrývá výkon menšinových tříd a zesiluje většinu.

3.4.5 Cross–validation

Jedná se o proces vyhodnocování úspěšnosti klasifikátorů, kdy námi vybraná datová sada nemá explicitně vyčleněnou testovací část. Datovou sadu rozdělíme na k podmnožin a $k-1$ podmnožin bude trénovací část a zbývající jedna podmnožina bude pro náš model testovací. Pro natrénovaný klasifikátor provedeme vyhodnocení (evaluaci). Následně obměníme jednu testovací množinu za jednu trénovací a proces opakujeme. Tímto získáme k nezávislých klasifikátorů, kde každý klasifikátor má jiná testovací data. Výslednou evaluaci klasifikátoru dostaneme jako aritmetický průměr měřených metrik všech k klasifikátorů.

Takto provedená evaluace má vyšší vypovídající hodnotu z hlediska statistické významnosti, protože lze porovnat směrodatné odchylky všech k klasifikátorů. Experimenty v této bakalářské práci jsou vyhodnocovány právě tímto způsobem.

4 Datová sada

Tato kapitola poskytuje stručný popis dostupných datových sad v českém jazyce, které se zaměřují na klasifikaci textu (zejména vícetřídní).

4.1 Dostupné datové sady v českém jazyce

ČSFD jedná se o datovou sadu filmových recenzí obsahující 91 tisíc filmových recenzí z československé filmové databáze (ČSFD). Každý z filmů lze klasifikovat do jedné ze tří skupin pozitivní, neutrální a negativní, proto tato datová sada spadá do skupiny multi–class klasifikace, vždy může být pouze v jedné skupině. Podrobnější informace viz. text [12].

MALL je datová sada obsahující 145 tisíc recenzí jednotlivých produktu z eshopu *Mall.cz*. Obdobně jako ve výše zmíněné datové sadě je každá recenze rozdělena do tří skupin pozitivní, neutrální a negativní, proto tato datová sada spadá také do skupiny multi–class klasifikace, vždy může být pouze v jedné skupině. Podrobnější informace viz. text [12].

FCB představuje datovou sadu vybraných příspěvků z několika českých stránek uvedených na stránce *facebook.com*. Velikost datové sady je uváděna okolo deseti tisíc textů, přičemž tyto texty byly ručně přiřazeny do jedné ze tří skupin. I tato datová sada patří do skupiny klasifikace multi–class, každý text patří pouze do jedné z tří skupin. Podrobnější informace viz. text [12].

CN je sada česky psaných novinových článků, získaná ze serveru *ceskenoviny.cz*. Tato datová sada patří k jedné z nejdůkladněji a nejdůsledněji označené datové sadě. Sada obsahuje okolo 250 tisíc článku. Každý článek v této sadě může patřit do jedné či více klasifikačních tříd a na rozdíl od výše zmíněných datových sad je tato stejně jak námi zvolená sada zařazená do skupiny multi–label klasifikace. Bohužel nebylo možné tuto datovou sadu podrobněji prostudovat a použít, jelikož se jedná o soukromou datovou sadu a informace veřejně dostupné lze naleznout zde [12]

4.2 Czech Text Document Corpus

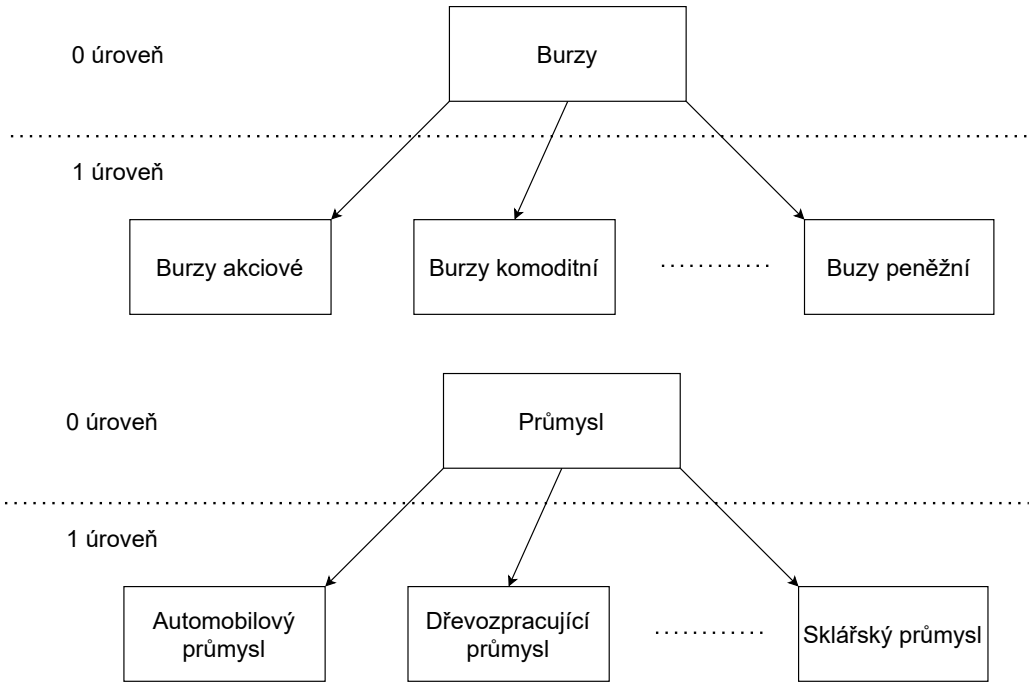
Pro úlohu vícetřídní klasifikace českého textu a s tím související experimenty v této práci bude využita česká datová sada s názvem *Czech Text Document Corpus v 2.0* [19]. Datová sada obsahuje celkem 11 955 česky psaných novinových článků, které jsou ručně klasifikovány do 60 možných

kategorií. V Tabulce 4.1 jsou uvedeny základní statistické informace o příslušné datové sadě; kolik obsahuje dokumentů, celkový počet slov a celkový počet kategorií, které jsou manuálně určeny.

Prvky	**Počet prvků**
Dokumenty:	11 955
Slova:	2 857 375
Kategorie:	60

Tabulka 4.1: Statistika datové sady

Obrázek 4.1 ukazuje část datasetu, kterou lze interpretovat jako hierarchická – tzn. anotované třídy tvoří jednotlivé úrovně (od obecné až po konkrétní). Bohužel takováto struktura neexistuje v datasetu plošně.



Obrázek 4.1: Ukázka úrovní klasifikace

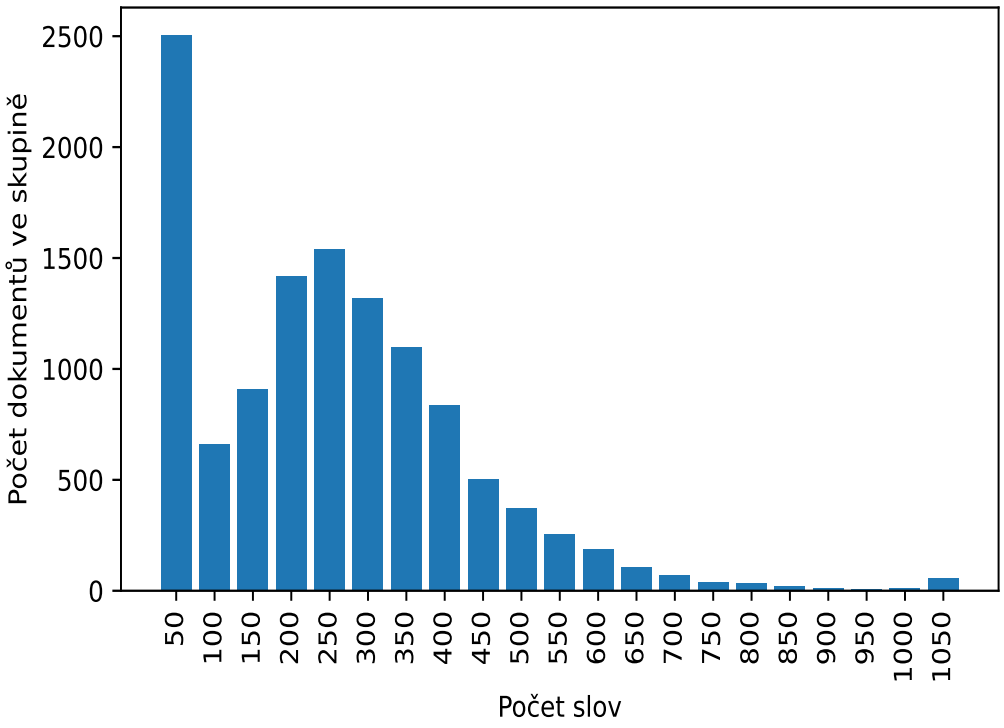
V článku [19] bylo použito pro experimenty podmnožina 37 nejvíce frekventovaných tříd. Aby bylo možné experimenty porovnat, zvolíme stejné nastavení. A proto následující statistiky obsahují grafy v rámci již redukované datové sady.

4.3 Statistiky

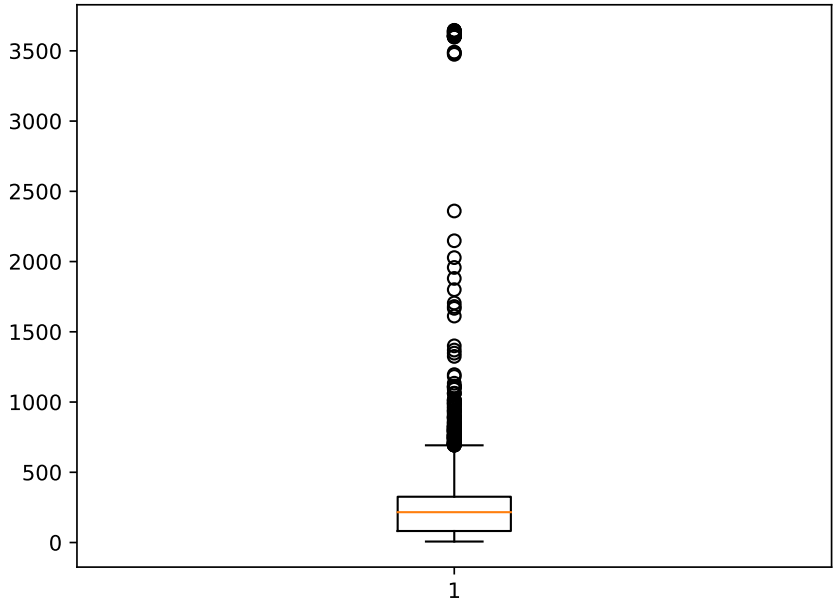
Graf na obrázku 4.2 určuje závislost počtu dokumentů na počtu slov. Na základě počtu slov v dokumentu je přiřazeno určitého rozmezí slov (příklad soubor 00013_dpr_ekl_pol_pod.txt obsahuje 225 slov a byl zařazen do sloupce 250 slov reprezentující rozsah 200 až 250).

Z grafu 4.2 můžeme vidět, že nejvíce článku bylo zařazeno do sloupce s intervalem do 50 slov, kterých bylo přibližně 2500 dokumentů. Drtivá většina dokumentů obsahuje maximálně 700 slov.

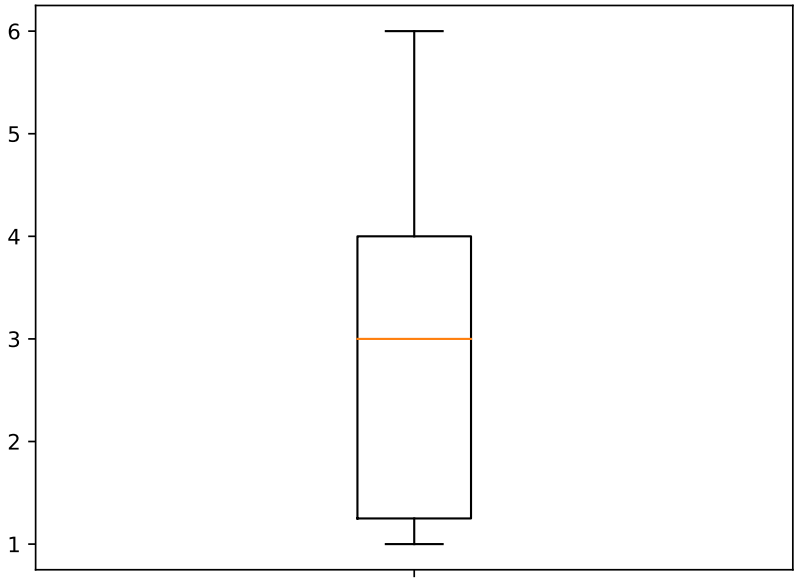
Druhým z grafů 4.3 byl zvolen tzv. box–plot neboli krabicový graf. Kde hranice box–plotu určuje nejvíce se vyskytující hodnoty. Oranžová úsečka uprostřed box–plotu určuje medián. Horní strana box–plotu určuje 1 kvartil a dolní strana určuje 3 kvartil. Dvě úsečky, které nejsou součástí box–plotu určují minima a maxima. Zvláštností tohoto grafu jsou kolečka, které představují tzv. outliery, které představují hodnoty vymezující se z rozsahu dodaných dat v našem případě počet slov v dokumentu.



Obrázek 4.2: Znázornění počtu dokumentů na základě počtu slov, které obsahují

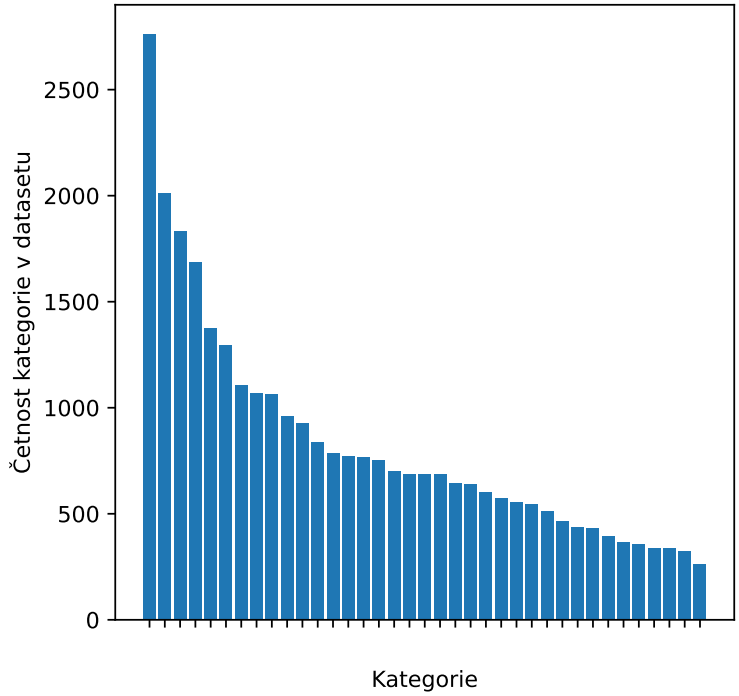


Obrázek 4.3: Znázornění distribuce počtu slov v datové sadě; medián je znázorněn oranžovou čárou



Obrázek 4.4: Znázornění distribuce počtu kategorií v rámci dokumentu (dole); medián je znázorněn oranžovou čárou

Na obrázku 4.4 je znázorněno počet třídu v rámci jednoho dokumentu. Nejmenší počet kategorií, který muže dokument obsahovat je 1 kategorie naopak nejvíce kategorií, které může dokument v datové sadě obsahovat je 6 kategorií. Nejčastější počet kategorií v datové sadě je v rozsahu od 1 do 4 kategorií jak uvádí vyznačený čtvereček grafu. Mediánem počtu těchto kategorií je 3, jak ukazuje oranžová čárka v grafu.



Obrázek 4.5: Znázornění četnosti kategorií v datové sadě s omezením na použitých 37 kategorií

Posledním ze statistických grafu pro datovou sadu je závislost četnosti kategorie v textových dokumentech datové sady, která je ukázána v grafu 4.5. Jak graf napovídá i zde je omezení na 37 nejčetnějších tříd seřazených sestupně. Dále můžeme z tvaru grafu predikovat vyváženost datové sady, ale toto tvrzení můžeme potvrdit teprve po provedení plánovaného experimentu zabývající se vyvážeností datové sady a výkonu modelu pro všechny možné kategorie. Je ale vidět, že všechny třídy mají dostatečné zastoupení.

4.4 Hierarchie a shluky tříd

V rámci této bakalářské práce, jsem se po dohodě s vedoucím práce rozhodl o pokročilejší náhled na klasifikační třídy. Jestli lze z klasifikačních tříd utvořit nějakou hierarchickou strukturu, což by mohlo vést ke shlukům a tím pádem snížení počtu klasifikačních tříd.

Při analýze datové sady bylo zjištěno, že tuto hierarchii by mohlo být možné zrealizovat pomocí hierarchických stromů, kde každá z vrstev stromu by obsahovala různou váhu z rozsahu $< 1, D >$,kde D je hloubka stromu. Následně by tyto váhy byly započteny do výpočtu 3.11, 3.12 a 3.13. Motivací pro toto je pokročilejší evaluace a přístup k chybám klasifikátoru (např. zaměnit sport domácí za sport zahraniční je menší chyba než zaměnit sport za průmysl).

Proběhlo několik pokusů o sestavení těchto stromů, ale žádný z pokusů nebyl zcela úspěšný. Téměř ve všech experimentech jsme vždy došli do situace, kdy chyběla kořenová třída pro konkrétnější třídy. Příkladem je Automobilový průmysl, Chemický a Farmaceutický průmysl a Strojírenství mají jako kořenovou třídu Průmysl, ten v této datové sadě chybí a bylo by potřeba vytvořit fiktivní třídu a tím narušit původní strukturu. Tvorba takovýchto "kořenových tříd" by mohla být zvážena pro teoretickou verzi 3.0 této datové sady.

Další podstatnou věcí neúspěchu bylo omezení na 37 nejčetnějších tříd. Toto omezení není podmínkou, ale poukázalo na to, že některé z tříd lze shlukovat do méně konkrétnějších tříd. Tento jev je žádoucí, pokud četnost kategorie je nízká. Limit nízké četnosti je individuální na každém uživateli a příslušné datové sadě.

4.4.1 Návrh shlukování

Výše bylo uvedeno, že shlukování můžeme provést pro třídy, kde jejich četnost v datové sadě je zanedbatelná. Při dalším zkoumání datové sady

byla vytvořena tzv. co-occurence matrix, která je symetrická a znázorňuje, jak se které dvě třídy spolu vyskytují v datové sadě.

	pol	efm	pod	zak	mag	for	kul	spo
pol	0	168	1834	514	133	1293	119	42
efm	168	0	130	278	42	73	27	262
pod	1834	130	0	331	63	785	91	33
zak	514	278	331	0	92	299	53	100
mag	133	42	63	92	0	57	556	54
for	1293	73	785	299	57	0	21	27
kul	119	27	91	53	556	21	0	6
spo	42	262	33	100	54	27	6	0

Tabulka 4.2: Výřez z co-occurence matrix, pro datovou sadu[19]; celá matice viz Příloha

Zde máme zobrazen pouze krátký výřez matice převedený do tabulkového provedení 4.2. Již z tohoto drobného výřezu lze vyčíst, že některé třídy se vyskytují pospolu velice často. Příkladem jsou třídy 'Politika'(pol), 'Politika ČR'(pod) a 'Parlament a vláda'(for). Tyto třídy obsahují podobný text a tedy i lze v těchto případech provést shluk do obecnější třídy, kde budou tyto specifičtější třídy zahrnuty a tím zvýšit i celkovou úspěšnost modelu.

Tento proces shlukování, kdy vytvoříme obecnou třídu, nelze ale bohužel provést plošně, pouze pro poměrně malou množinu tříd (viz Příloha B), proto byla myšlenka snižování počtu tříd tímto jednoduchým způsobem zavržena.

5 Popis experimentů

V této kapitole budou popsány použité modely, včetně shrnutí základních informací. Jako druhou z částí teto kapitoly bude nastínění základní myšlenky jednotlivých sad experimentů, jednotlivých experimentů a také krátký popis použitých přístupů a konkrétní popis postupu co napsaný program provádí.

5.1 Použité modely

small–e–czech je nová neuronová síť, která byla vytvořená a přetrénovaná firmou *Seznam.cz*. Síť je schopna řešit úlohy spjaté s porozuměním českého jazyka. Díky ní se povedlo firmě *Seznam.cz* zkvalitnit výsledky vyhledávaní a odstranění překlepů ve vyhledávaném textu.

small–e–czech vychází ze sítě typu Electra [10] z roku 2020. Nejprve byl před učen na velkém množstvím českých textů, aby získal povědomi o českém jazyce, významu slov a následně je doučen na menším datové sadě na konkrétní úlohu, kterou od sítě požadujeme. Před učení sítě bylo prováděno po dobu 20 dní na velikosti 250GB. Podrobnější informace o neuronové síti *small–e–czech* lze naleznou zde [4, 17]

FERNET–C5 je neuronová síť, vytvořená výzkumníky z katedry kybernetiky fakulty Aplikovaných věd na Západočeské univerzitě v Plzni. Síť je schopna řešit úlohy spjaty s porozuměním českého jazyka, jelikož pro tento účel byla předtrénovaná.

Model *FERNET–C5* je typ sítě vycházející z modelu BERT-base [22] z roku 2018 od firmy *Google*. Jedná se o model z rodiny jazykových modelů. Model *FERNET–C5* byl nejprve naučen na datové sade Common Crawl dataset (C5), která je českou modifikací anglické datové sady C4 [25], který byl dodán jako obyčejná textová data. Před učení proběhlo na datové sadě C5 s velikostí 93GB filtrovaných český textů. Podrobnější informace o modelu lze získat zde [20]

FERNET–News je další z neuronové sítě, od výzkumníku z katedry kybernetiky fakulty Aplikovaných věd na Západočeské univerzitě v Plzni. Tento model je obdobně jako model *FERNET–C5* před trénován pro úlohy spojeny s českým jazykem.

Model *FERNET–News* je založen na jazykovém modelu RoBERTa-base [21], která vychází z klasického modelu Bert, který je robustnější a více optimalizovaný než výchozí model. V případě *FERNET–News* byla použita

datová sada Czech news corpus (News Corpus), kde data byla získaná pomocí vytvořeného frameworku, získávající informace z webových článku, televizních pořadu a radiových stanic. Datová sada byl očištěn o html značky a převeden do normalizované podoby z výslednou velikostí 20,5GB. Více o datové sadě lze naleznout zde [32].

RobeCzech je česko jazyčný kontextový model založený na architektuře transformer a trénován výhradně na česky psaných textech. Jak bylo řečeno jedná se jedno jazyčný model vycházejí z výchozího modelu RoBERTa, který představuje robustně optimalizovaný předtrenovovaný model BERT. *RobeCzech* byl vytvořen na matematicko fyzikální fakultě univerzity Karlovy.

Tento model byl před trénován na čistě psaných českých texte, které jsou **SYN v4** , velký korpus současné psané češtiny, 4 188M tokenů, **Czes** , sbírka českých novinových a časopiseckých článků, 432 mil. žetonů, okumenty s minimálně 400 tokeny z české části webového korpusu **W2C** a prosté texty extrahované z české **Wikipedie**. Podrobnější popis o příslušném modelu naleznete zde [30].

5.2 Nastavení experimentů

Program bakalářské práce obsahuje celkově čtyři sady experimentů. Dvě sady experimentů jsou prováděny na modelu *small-e-czech* s rozdílným počtem výstupních tříd. První sada testuje model pří počtu 37 výstupních tříd a druhá sada testuje pro 60 výstupních tříd. Další dvě sady experimentů jsou prováděny pro počet 37 výstupních tříd, ale vždy na jiném z modelů *FERNET–News* a *RobeCzech*.

Každá sada experimentů obsahuje 3 konfigurace, kde každá z nich bere v úvahu různý počet vstupních tokenů. Počet vstupních tokenů je 50, 128 a 512. Každý experiment využívá evaluaci pomocí cross-validace, kde jsme rozdělili datovou sadu na pět stejně velikých částí vždy $\frac{4}{5}$ použijeme jako trénovací data a $\frac{1}{5}$ jako testovací data.

Obecný princip jednotlivých experimentů a testů je takový, že zvolený model je načtený a probíhá ladění (angl. fine-tuning) na úloze vícetřídní klasifikace. Tento model je po absolvování požadovaného množství epoch (v experimentech použito většinou 15 – 30 epoch) uložen do souboru, aby bylo možné je poté pouze načíst a vyhodnotit. Parametr *learning rate* (míra učení) je nastavena na 0.0001.

6 Prezentace výsledků a diskuze

Hlavním modelem pro testování experimentů byl zvolen model *small-e-czech*, který byl vytvořen teprve nedávno a zatím není mnoho textů, které by jej testovali na klasifikačních úlohách. Z tohoto důvodu byl model otestován a porovnán s ostatními modely, které obsahují velké množství parametrů, jako je například **BERT**. V každé níže uvedené tabulce je vždy tučně vyznačena nejvyšší hodnota, kterou model získal pro příslušnou metriku.

6.1 Model small-e-czech

Input tokens	50	128	512
Micro Precission	86.06%±0.54	**88.88%±0.76**	86.99%±1.28
Micro Recall	82.87%±0.72	82.16%±0.88	**85.11%±0.51**
Micro F1	84.43%±0.30	85.38%±0.37	**86.03%±0.45**
Macro Precission	86.88%±0.82	**89.96%±0.47**	87.87%±0.63
Macro Recall	82.00%±0.77	80.54%±0.49	**84.49%±0.40**
Macro F1	84.36%±0.26	84.99%±0.42	**86.14%±0.32**
Exact match	60.17%	60.86%	**61.47%**

Tabulka 6.1: Výsledky modelu small-e-czech_epochs=20_lr=0.0001 pro 37 klasifikačních tříd, v závislosti na počtu vstupních tokenů

První ze sady experimentů, bylo testování modelu *small-e-czech* pro 37 klasifikačních tříd v závislosti na počtu vstupních tokenů, kde počet vstupních tokenů byl 50, 128 a 512. Pro každý počet vstupních tokenů byl proveden experiment, čili trénování a evaluace pomocí metrik Precission, Recall a výsledná F–míra, pro agregaci Micro a Macro.

Z výsledku uvedené v tabulce 6.1 je patrné, že i přestože se jedná o menší model oproti ostatním, tak celková F–míra je pouze o několik málo procent nižší ve srovnání s ostatními většími modely jako je například *FERNET–News (Bert)* nebo *RobeCzech (RoBerta)* – viz srovnávací tabulka 6.7.

Dále můžeme vidět, že po redukci na 37 tříd je datová sada vyvážená, jelikož hodnota F–míry pro macro i micro je velice podobná. Lze díky tomu

potvrdit pozorování z grafu 4.5 o dostatečném zastoupení všech 37 tříd v datové sadě.

Velmi pozoruhodnou informací je, že i pro relativně nízký počet vstupních tokenů, tj. 50, lze získat slušné výsledky. Tento fakt naznačuje, že nejpodstatnější informace jsou uvedeny na začátku článků. Výhodou použití nižšího počtu tokenů je zvýšená rychlost trénování, aniž bychom ztratili podstatným způsobem úspěšnost modelu.

Nejvyšší úspěšnosti z hlediska většiny metrik model *small-e-czech* získal s maximální velikostí tokenů 512 (téměř 61.5% exact match).

Druhá sada experimentů, testovala model *small-e-czech* za stejných podmínek jako v případě první sady experimentů. Nebyla ovšem provedena redukce na 37 nejčastějších tříd, ale bylo použito všech 60 definovaných tříd. V tomto případě je očekávána nižší macro hodnota, protože třídy s nízkým zastoupením se model pravděpodobně nenaučí.

Input tokens	50	128	512
Micro Precission	87.51%±1.10	**88.82%±1.31**	87.12%±0.25
Micro Recall	77.79%±0.46	79.82%±0.69	**84.99%±0.87**
Micro F1	82.36%±0.50	84.06%±0.29	**86.04%±0.34**
Macro Precission	79.47%±0.80	80.60%±1.03	**87.55%±1.03**
Macro Recall	57.59%±0.34	59.46%±1.52	**77.05%±1.92**
Macro F1	66.78%±0.22	68.42%±0.94	**81.95%±1.14**
Exact match	53.46%	55.14%	**59.86%**

Tabulka 6.2: Výsledky modelu small-e-czech_epochs=20_lr=0.0001 pro 60 klasifikačních tříd, v závislosti na počtu vstupních tokenů

Z výsledku v tabulce 6.2 vidíme, že rozdíl hodnot F–míry pro Micro a Macro agregaci už není zanedbatelný tak jako v případě první sady experimentů. Rozdíl mezi macro a micro hodnotou F–míry je pro input tokens 50 přibližně 16%.

Další zajímavý fakt vyplývající z tabulky 6.2 je, že v této konfiguraci silně pomáhá větší počet vstupních tokenů. Při použití 512 vstupních tokenů jsou výsledky jasně nejlepší (viz řádek Macro F1 a hodnota v posledním sloupečku). Hodnota exact match rovněž je skoro 60%.

6.2 Modely FERNET–News a RobeCzech

V následujících dvou sadách experimentů jsem vyzkoušel provést měření s většími modely, konkrétně *FERNET–News* a *RobeCzech*. Tyto modely byly výrazně pomalejší z hlediska času trénování a také náročnější na hledání hyperparametrů. Ačkoliv jsou experimenty na této datové sadě a výsledky z těchto modelů již publikované ([20] a [30]), i tak jsem se rozhodl po dohodě s vedoucím experimenty provést. Důvodem bylo porovnání rychlosti a celkové chování modelů. Bohužel z důvodu nedostatku času byly provedeny a změřeny pouze některé experimenty.

Input tokens	50
Micro Precission	87.64%±1.17
Micro Recall	87.67%±1.24
Micro F1	87.64%±0.42
Macro Precission	88.20%±1.01
Macro Recall	87.28%±0.89
Macro F1	87.73%±0.31
Exact match	65.53%

Tabulka 6.3: Výsledky modelu FERNET–News_epochs=20_lr=0.0001 pro 37 klasifikačních tříd pro 50 vstupních tokenů

Input tokens	50
Micro Precission	88.3%±0.60
Micro Recall	85.16%±1.82
Micro F1	86.69%±1.12
Macro Precission	88.26%±1.19
Macro Recall	78.66%±2.86
Macro F1	81.5%±2.44
Exact match	61.87%

Tabulka 6.4: Výsledky modelu FERNET–News_epochs=16_lr=0.0001 pro 60 klasifikačních tříd pro 50 vstupních tokenů

U obou těchto modelů docházelo také občas ke stagnaci při trénování, kdy se model nebyl schopen naučit (zejména v případech, kdy se použil delší vstup – 128 nebo 512 tokenů viz obrázky 6.3 nebo 6.4).

Input tokens	50
Micro Precission	87.54%±0.73
Micro Recall	87.67%±0.82
Micro F1	87.54%±0.18
Macro Precission	88.44%±0.50
Macro Recall	87.06%±0.21
Macro F1	87.74%±0.75
Exact match	65.27%

Tabulka 6.5: Výsledky modelu RobeCzech_epochs=20_lr=0.0001 pro 37 klasifikačních tříd pro 50 vstupních tokenů

Input tokens	50
Micro Precission	88.04%±2.07
Micro Recall	84.12%±1.41
Micro F1	86.01%±0.73
Macro Precission	83.95%±1.22
Macro Recall	68.19%±1.91
Macro F1	75.25%±1.54
Exact match	59.3%

Tabulka 6.6: Výsledky modelu RobeCzech_epochs=20_lr=0.0001 pro 60 klasifikačních tříd pro 50 vstupních tokenů

6.3 Celkové porovnání výsledků

V tabulce 6.7 lze vidět poslední ze statistik a to je porovnání výkonosti vybraných modelů neuronových sítí podle metriky Micro F–míry.

Modely	Micro F1
small-e-czech	86.03%±0.45
FERNET–C5 [20]	**91.25%±0.38**
FERNET–News [20]	90.85%±0.47
RobeCzech [30]	90.47%±0.53
Czert [29]	90.23%±0.49
Slavic-BERT [5]	89.59%±0.48
MLP (Lenc and Kral, 2017) [19]	83.90%
CNN (Lenc and Kral, 2017) [19]	84.70%

Tabulka 6.7: Celkové srovnání vybraných modelů, které byly použity při klasifikaci česky psaného textu; pro výsledky [19] není uvedena směrodatná odchylka

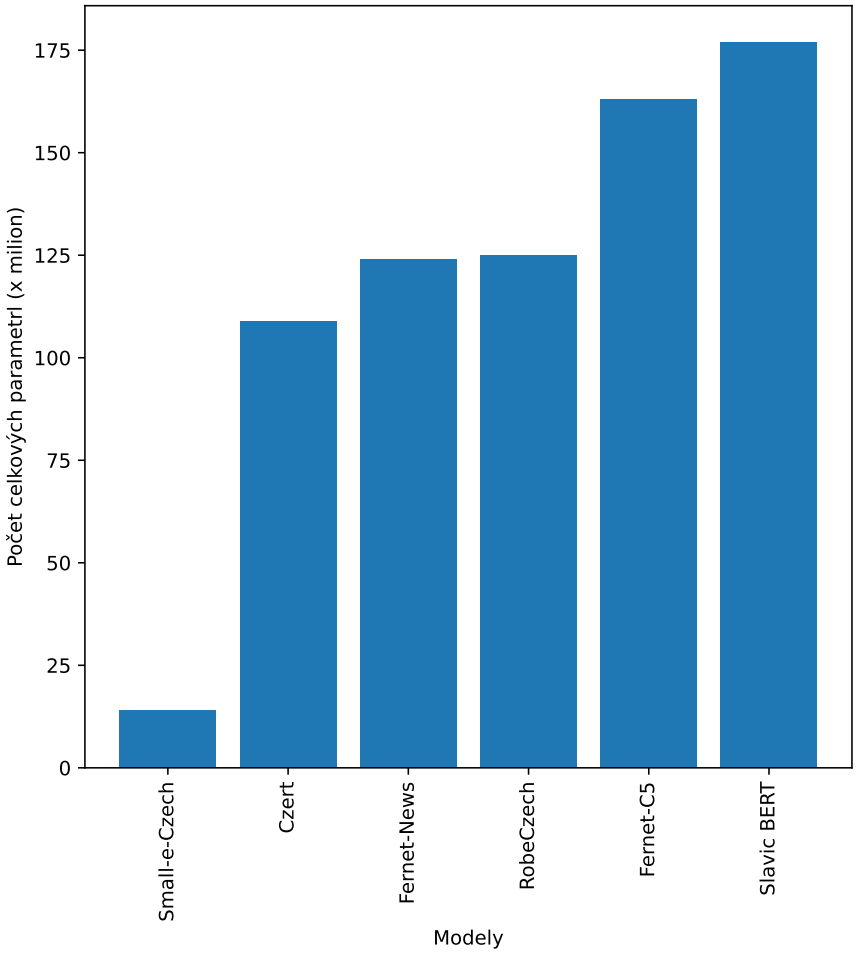
Z výsledků můžeme vyčíst, že velké modely typu BERT nebo RoBERTa dosahují vynikajících výsledku přes 90%. Bohužel není k dispozici srovnání se všemi 60 třídy. Do přehledu výsledků jsem přidal i výsledky z článku z roku 2017 [19], kde modely jako je vícevrstvý perceptron (MLP) nebo konvoluční neuronová síť (CNN) dosáhly hodnot okolo 84% micro F–míry.

Z mého pohledu má model *small-e-czech* optimální poměr mezi délkou učení sítě a dosažené úspěšnosti. Do výčtu jsou ještě přidány výsledky modelu Czert a Slavic-BERT, který je vícejazyčný.

6.4 Časy běhů a počet parametrů modelů

Pro každý z modelů byla provedena základní statistika celkového počtu parametrů a rychlost výpočtu jedné epochy. Tyto statistiky byly zjištěny na PC s operačním systémem Linux Debian a pro výpočet neuronové sítě byla použita grafická karta Nvidia–GeForce RTX 3060 s 12 GB GPU memory.

- *FERNET–News*, varianta RoBERTa base 123 milionů parametrů,
- *FERNET–C5*, varianta Bert base 163 milionů parametrů,
- *RobeCzech*, Varianta RoBERTa base 125 milionů parametrů,
- *small-e-czech*, varianta Electra small 14 milionů parametrů.



Obrázek 6.1: Znázornění celkového počtu parametrů pro jednotlivé modely

První ze statistických údajů je celkový počet parametrů pro jednotlivé modely. Je vidět že modely založeny na typu BERT jsou skoro deseti násobně vetší, než model *small-e-czech*, který obsahuje 14 milionů parametrů.

V grafu 6.1 je znázorněno porovnání počtu parametrů. Nejvíce parametrů obsahuje model *Slavic–Bert*[5]. Tento poslední model je vícejazyčný model, který byl natrénován pro slovanské jazyky (Čeština, Polština, Bulharština a Ruština). Tento model obsahuje 180 milionů parametrů, což je cca třináctinásobek modelu *small-e-czech*.

Porovnání z hlediska času trénování ukazuje tabulka 6.8. Parametr, který výrazně ovlivňuje rychlost výpočtu jedné epochy je velikost dávky (batch size). Batch size byla volena pro každý experiment tak, aby byla GPU maximálně vytížena z hlediska GPU memory. Obecně ale platí, že čím větší model, tím jsme nuceni snižovat batch size a tím pádem se čas trénování prodlužuje.

Model	Čas jedné epochy [min]	Batch size
small-e-czech	0.16	128 vzorků
RobeCzech	0.60	128 vzorků
FERNET–News	0.60	128 vzorků

Tabulka 6.8: Časy trénování pro experimenty s 50 vstupními tokeny; zprůměrováno a zaokrouhleno na celé minuty

Model	Čas jedné epochy [min]	Batch size
small-e-czech	0.30	128 vzorků
RobeCzech	1.56	48 vzorků
FERNET–News	1.33	48 vzorků

Tabulka 6.9: Časy trénování pro experimenty s 128 vstupními tokeny; zprůměrováno a zaokrouhleno na celé minuty

Model	Čas jedné epochy [min]	Batch size
small-e-czech	1.75	32 vzorků
RobeCzech	8	8 vzorků
FERNET–News	8	8 vzorků

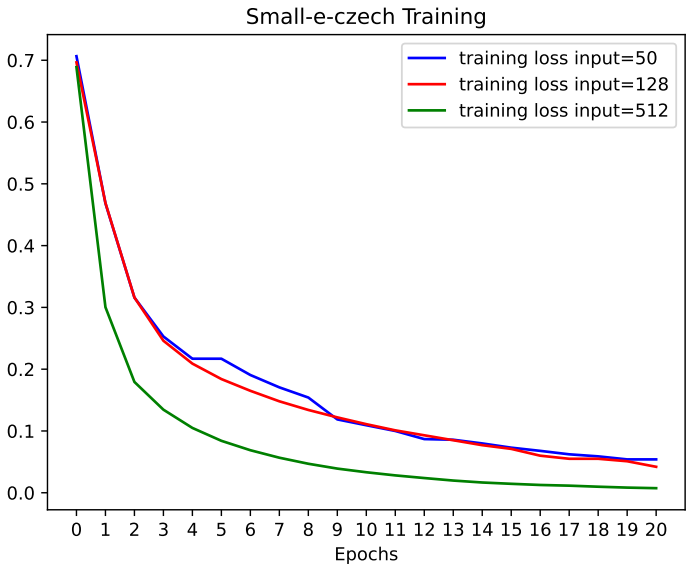
Tabulka 6.10: Časy trénování pro experimenty s 512 vstupními tokeny; zprůměrováno a zaokrouhleno na celé minuty

6.5 Průběhy loss function

Tato sekce obsahuje sadu obrázků s průběhem trénování (tj. závislost loss function na trénovacích epochách).

6.5.1 Small-e-czech

Experimenty s tímto modelem byly nejstabilnější ze všech prováděných. Model vykazoval setrvalý pokles ztrátové funkce, a to i s různým nastavení learning rate (od 0.0001 až 0.005). Pokud se použila maximální možná délka vstupu (512 tokenů) konvergence modelu byla rychlejší a celková úspěšnost rovněž lepší než pro ostatní konfigurace.

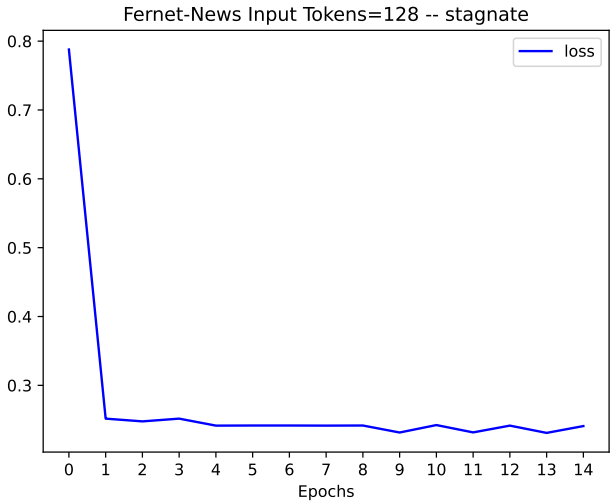


Obrázek 6.2: Průběh trénování modelu small-e-czech pro input tokens=50, 128 a 512

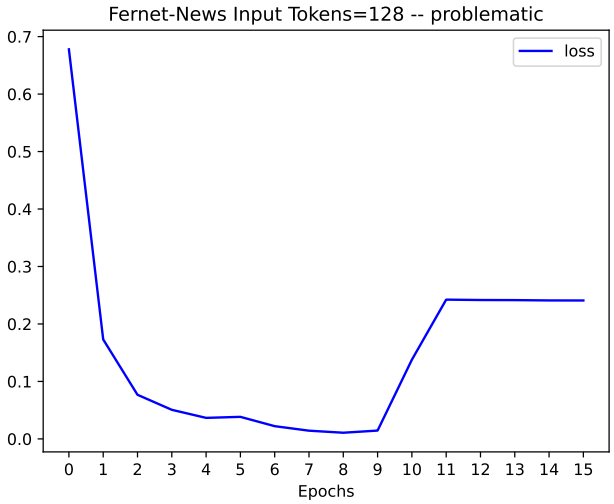
I když maximální možná délka vstupu může být omezující (např. v případech klasifikace dokumentů s velmi dlouhým textem) model se chová konzistentně i pro vyšší počet tříd (např. 60). Zajímavým experimentem by bylo vyzkoušení modelu na datové sadě, která obsahuje několik stovek či tisíce tříd a pozorovat jeho chování.

6.5.2 FERNET–News

Tento model vykazoval (podobně jako *RobeCzech*) jisté problémy při trénování a ne vždy docházelo ke konvergenci. Následující dva obrázky ukazují problematické situace (obr. 6.3 stagnace a obr. 6.4 ukazuje případ, kdy v prvních 9 epochách docházelo k poklesu, ale pak loss funkce začala růst).



Obrázek 6.3: Problematický průběh (stagnace) trénování modelu FERNET–News input tokens=128



Obrázek 6.4: Problematický průběh trénování modelu FERNET–News input tokens=128

V obou těchto případech došlo k znehodnocení výsledku výkonu modelu. V první případě zobrazeném na obrázku 6.3 došlo ihned od první epochy ke stagnaci hodnoty loss function. V druhém případě na obrázku 6.4 byl průběh učení z počátku velmi perspektivní až do deváté epochy, kdy nastal prudký nárůst loss function a následná stagnace. Oba tyto případy měly za následek neúplné využití potenciálu sítě a tedy jejich výsledný výkon neodpovídal předpokládanému výkonu.

Při tomto modelu (podobně i pro *RobeCzech*) by bylo potřeba provést několik další experimentů, které by měli za úkol nalezení optimálních hyperparametrických hodnot pro vybraný model .

6.6 Analýza Chyb

Experimentálně bylo zjištěno, že v některých případech model *small-e-checz* predikuje špatně klasifikační třídy, které jsou si podobné. Např. u prvního dokumentu bylo zjištěno, že model predikoval úspěšně kategorie *politika* a *politika ČR*, ale následně predikoval kategorii *Parlament a vláda* na místo správné predikce *Kriminalita a práva*. Tato špatná predikce je v tomto případě způsobené podobností obsahu mezi těmito dvěma kategorii. Takových případů se v datové sadě nachází více, ale ve většině se bude jednat o podobnost obsahů témat, která jsou klasifikována. Pro představu, které z témat mohou být zaměňovány lze predikovat z coocurence matrix uvedené v Příloze B.

Další analýzou bylo zkoumání metrik Precision (P) a Recall (R) pro pět nejčetnějších a pět nejméně četných kategorií z výběru 37 kategorii. Při pohledu na Tabulku 6.11 zjistíme, že rozdíl mezi P a R je poměrně malý. Pro tyto třídy je pak poměrně velká i F1. Pro nejméně četné třídy (Tabulka 6.12) je situace jiná a rozdíl mezi P a R je velký. Vysoká přesnost (P) značí, že pokud model *small-e-czech* určí třídy *Pragensie*, *Media a reklama*, *Potravinářství* nebo *Suroviny*, tak si je velmi jistý, že tomu tak je. Nižší úplnost (R) ale značí, že model neoznačí zdaleka všechny, co by označit měl. Kvůli malé četnosti těchto téma se model nezvládl naučit podstatné informace pro příslušné téma a proto pouze některá z témat jsou specifická sami o sobě, například *Pragensie* a *Media a reklama*. Tabulka B.2 v Příloze B ukazuje hodnoty P a R pro všech 37 tříd.

	Precision	Recall
Politika (pol)	81.01%	79.36%
Firmy (efm)	89.60%	83.09%
Politika ČR (pod)	81.49%	76,61%
Kriminalita a právo (zak)	90.80%	70.65%
Magazínový výběr (mag)	85.93%	73.09%

Tabulka 6.11: Metriky Precision a Recall pro 5 nejčetnějších tříd

	Precision	Recall
Suroviny (sur)	97.31%	68.47%
Potravinářství (ptr)	95.68%	43.95%
Media a reklama (med)	96.67%	17.43%
Pragensie (prg)	95.56%	7.23%
Chemický a Farmaceutický průmysl (che)	77.51%	32.28%

Tabulka 6.12: Metriky Precision a Recall pro 5 nejméně nejčetnějších tříd

7 Závěr

Tato bakalářská práce obsahuje experimenty pro úlohu vícetřídní klasifikace textu v českém jazyce. Zaměřili jsme se na datovou sadu *Czech Text Document Corpus v. 2.0* (CTDC), který obsahuje novinové články a jejich anotace. Ačkoliv datová sada obsahuje 60 klasifikačních tříd, autoři prvotních experimentů v článku [19] provedli redukci datové sady a vybrali pouze 37 nejčetnější.

Experimenty byly provedeny s moderními modely neuronových sítí typu *Transformer*. Pro implementaci byl zvolen jazyk Python s knihovnou Torch. Hlavní testovaný model byl český *small-e-czech*, po dohodě s vedoucím byly přidány experimenty s *FERNET–News* a *RobeCzech*. Díky tomu byla porovnána výkonnost relativně malého modelu s modely, které jsou více než 10-ti násobně velké z hlediska počtu parametrů. Dosažené výsledky byly porovnány s dostupnými publikovanými články.

Byly provedeny i některé experimetny s počtem tříd 60. Dle očekávání klesla výrazně hodnota Macro F1, kvůli méně četným třídám. I když nebyly provedeny všechny experimenty s modely *FERNET–News* a *RobeCzech*, zdá se že, v případě délky vstupu 512 by model *small-e-czech* spolehlivě překonaly. Bohužel nebylo možné se porovnat s publikovanými výsledky při použití všech 60 tříd, protože všechny citované publikace obsahují výsledky na redukované datové sadě.

Hlavní výhodou modelu *small-e-czech* je z mého pohledu poměr rychlosti učení a úspěšnosti klasifikace. *Small-e-czech* umožňuje rychlé učení a ladění na cílovou úlohu. Další výhodou je i relativní stabilita na výběr hyperparametrů (learning rate, počty epoch apod.).

Zajímavé je rovněž porovnání vícejazyčného modelu *Slavic-Bert* s ryze českými modely. Modelem *small-e-czech* se jej nepodařilo porazit. Je nutné si ale uvědomit, že model *Slavic-Bert* ze všech sledovaných modelů obsahuje největší počet parametrů (180 milionů oproti 14 milionům modelu *small-e-czech*).

V neposlední řadě byla provedena analýza chyb a hierarchie klasifikačních tříd. Bylo provedeno několik pokusů o vytvoření shluků tříd, které buďto byly málo četné (a tak dávalo smysl je spojit s jinou třídou) a nebo se dvě a více objevovaly ve většině případů pospolu (velká část společného obsahu). Takovéto shlukování se ale nakonec nepodařilo uskutečnit. Nicméně by to bylo námětem pro další aktualizaci datové sady (např. ve verzi 3.0).

Literatura

[1] Tomáš Jurczyk, M. U. Neuronové sítě a jejich využití. *IT Systems*. 2014, 3633, 2, s. 2–3.

[2] Agarap, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*. 2018.

[3] Allwright, S. *Micro vs Macro F1 score, what's the difference?* [online]. 2022. [cit. 2023/04/13]. Dostupné z: `https://stephenallwright.com/micro-vs-macro-f1-score/`.

[4] arahusky. *seznam/small-e-czech* [online]. 2022. [cit. 2023/04/13]. Dostupné z: `https://github.com/seznam/small-e-czech`.

[5] Arkhipov, M. et al. Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, s. 89–93, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3712. Dostupné z: `https://aclanthology.org/W19-3712`.

[6] Bahdanau, D. – Cho, K. – Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. 2014.

[7] Balachandar, K. *ML – Attention mechanism* [online]. 2022. [cit. 2023/01/24]. Dostupné z: `https://www.geeksforgeeks.org/ml-attention-mechanism/?ref=rp`.

[8] Bishop, C. M. – Nasrabadi, N. M. *Pattern recognition and machine learning*. Springer, 2006.

[9] Chakravarthy, S. *Tokenization for Natural Language Processing* [online]. 2020. [cit. 2022/10/06]. Dostupné z: `https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4`.

[10] Clark, K. et al. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, 2020.

[11] Devlin, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, s.

4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. Dostupné z: https://aclanthology.org/N19-1423.

[12] Habernal, I. – Ptáček, T. – Steinberger, J. Sentiment Analysis in Czech Social Media Using Supervised Machine Learning. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, s. 65–74, Atlanta, Georgia, June 2013. Association for Computational Linguistics. Dostupné z: http://www.aclweb.org/anthology/W13-1609.

[13] Heidenreich, H. *Stemming? Lemmatization? What?* [online]. 2018. [cit. 2022/10/06]. Dostupné z: https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8.

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

[15] IBM. *What is deep learning?* [online]. [cit. 2023/02/16]. Dostupné z: https://www.ibm.com/topics/deep-learning.

[16] Karani, D. *Introduction to Word Embedding and Word2Vec* [online]. 2018. [cit. 2022/10/06]. Dostupné z: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa.

[17] Kocián, M. et al. Siamese BERT-based Model for Web Search Relevance Ranking Evaluated on a New Czech Dataset. In *Siamese BERT-based Model for Web Search Relevance Ranking Evaluated on a New Czech Dataset*. arXiv, 2021. doi: 10.48550/ARXIV.2112.01810. Dostupné z: https://arxiv.org/abs/2112.01810.

[18] Kostadinov, S. *Understanding GRU Networks* [online]. 2017. [cit. 2022/09/20]. Dostupné z: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be.

[19] Kral, P. – Lenc, L. Czech Text Document Corpus v 2.0. In chair), N. C. C. et al. (Ed.) *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, May 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

[20] Lehečka, J. – Švec, J. Comparison of Czech Transformers on Text Classification Tasks. In *Statistical Language and Speech Processing*. : Springer International Publishing, 2021. s. 27–37. doi: 10.1007/978-3-030-89579-2_3. Dostupné z: https://doi.org/10.1007%2F978-3-030-89579-2_3.

[21] Liu, Y. et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.

[22] Lutkevich, B. *BERT language model* [online]. 2020. [cit. 2022/10/06]. Dostupné z: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model.

[23] Nielsen, M. A. *Neural networks and deep learning*. 25. Determination press San Francisco, CA, USA, 2015.

[24] Olah, C. *Understanding LSTM Networks* [online]. 2015. [cit. 2022/09/20]. Dostupné z: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[25] Raffel, C. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, 2020.

[26] Rosenblatt, F. The perceptron - A perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York, January 1957.

[27] Scott, W. *TF-IDF from scratch in python on a real-world dataset* [online]. 2019. [cit. 2023/01/30]. Dostupné z: https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089.

[28] Sharma, P. *An Introduction to Stemming in Natural Language Processing* [online]. 2021. [cit. 2022/10/06]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/11/an-introduction-to-stemming-in-natural-language-processing/.

[29] Sido, J. et al. Czert – Czech BERT-like Model for Language Representation, 2021.

[30] Straka, M. et al. RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. In *Text, Speech, and Dialogue*. : Springer International Publishing, 2021. s. 197–209. doi: 10.1007/978-3-030-83527-9_17. Dostupné z: https://doi.org/10.1007%2F978-3-030-83527-9_17.

[31] Vaswani, A. et al. Attention is All you Need. In Guyon, I. et al. (Ed.) *Advances in Neural Information Processing Systems*, 30. Curran Associates, Inc., 2017. Dostupné z: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[32] Jan et al. General framework for mining, processing and storing large amounts of electronic texts for language modeling purposes. *Language Resources and Evaluation*. 2014, s. 227–248. ISSN 1574-020X. doi: 10.1007/s10579-013-9246-z. Dostupné z: http://www.kky.zcu.cz/en/publications/SvecJan_2014_Generalframeworkfor.

[33] Wikipedia. *Lemmatisation* [online]. 2022. [cit. 2022/10/06]. Dostupné z: https://en.wikipedia.org/wiki/Lemmatisation#Algorithms.

A Uživatelská příručka

Program testující experimenty, které určují výkonost zkoumaných modelů pracuje ve třech módech **Statistický**, **Experimentální** a **Testovací**. Je potřeba aby pro všechny módy vždy existoval textový soubor s klasifikačními třídami uložený ve složce se spouštěcím modulem.

A.1 Statistický mód

V tomto módu program zjistí základní statistické údaje o datové sadě (Počet dokumentů všech před zpracování, Počet dokumentů textový, počet slov a 37 nejčetnějších kategorií seřazených od nejčetnější) dále vytvořit dva pdf soubory s grafy, kde jeden bude box plot s počtem slov v dokument a druhý bude klasický sloupcový graf kde je uveden počet dokumentů spadající do jedné z N kategorií. Následně při výpisu kategorií provede tvorbu souboru, kde bude uvedeno příslušných 37 nejčetnějších klasifikačních tříd, jelikož tento soubor nebo soubor se všemi klasifikačními třídami je použit v dalších módech. Tento mód požaduje jako jediný parametr cestu k příslušné datové sadě. Mód je plně automatizovaný a není potřeba žádná interakce ze strany uživatele ale i přesto výpočet statistik nějakou dobu trvá (přibližně minutu) Statistický mód spustíme příkazem ve formátu:

```
python3 Main.py statistic <dataset>
```

```
$ python Main.py statistic
/mnt/data/ctk_czech_text_corpus/czech_text_document_corpus_v20
Load all name of files
Load all categories
Count of file in dataset for all preprocesing is 59775
Count of text file in dataset is 11955
Count of all words in dataset is 2857375
1.categorie: Politika, count: 2762
2.categorie: Firmy, count: 2011
3.categorie: Politika_ČR, count: 1834
4.categorie: Kriminalita_a_právo, count: 1687
5.categorie: Magazínový_výběr, count: 1375
6.categorie: Parlamenty_a_vláda, count: 1293
7.categorie: Kultura, count: 1105
8.categorie: Sportovní_zpravodajství, count: 1068
9.categorie: Doprava, count: 1061
10.categorie: Zdravotnictví, count: 961
11.categorie: Služby, count: 927
12.categorie: Makroekonomie, count: 838
13.categorie: Stavebnictví_a_reality, count: 787
14.categorie: Energie, count: 769
15.categorie: Životní_prostředí, count: 765
16.categorie: Životní_styl, count: 753
17.categorie: Počasí, count: 698
18.categorie: Burzy, count: 687
19.categorie: Finanční_služby, count: 684
20.categorie: Obchod, count: 684
21.categorie: Telekomunikace_a_IT, count: 645
22.categorie: Práce_a_odbory, count: 639
23.categorie: Zpravodajské_deníky, count: 600
24.categorie: Strojirenství, count: 573
25.categorie: Evropská_unie-zprávy, count: 553
26.categorie: Zemědělství, count: 544
27.categorie: Cestovní_ruch, count: 511
28.categorie: Sociální_problematika, count: 464
29.categorie: Školství, count: 438
30.categorie: Automobilový_průmysl, count: 431
31.categorie: Mix, count: 396
32.categorie: Burzy_peněžní, count: 365
33.categorie: Suroviny, count: 356
34.categorie: Potravinářství, count: 337
35.categorie: Media_a_reklama, count: 336
36.categorie: Pragensie, count: 325
37.categorie: Chemický_a_Farmaceutický_průmysl, count: 262
```

A.2 Experimentální mód

Druhý z módu je experimentální, který provádí trénování zvoleného modelu, uložení modelu a následně provede také vyhodnocení výkonu modelu. Tento mód požaduje 3 vstupní argumenty, kterými jsou název módu , cesta k datové sadě a název modelu.
Experimentální mód spustíme příkazem:

```
python3 Main.py experiments <dataset> <model>
```

Možné zadané modely(<model>):

- small-e-czech,
- fernet,
- robeczech

```
$ python Main.py experiments
/mnt/data/ctk_czech_text_corpus/czech_text_document_corpus_v20/ robeczech
Successfully opened dynamic library libcudart.so.11.0
Load all name of text files
Loading data...
Processed files: 0/11955
Processed files: 500/11955
Processed files: 1000/11955
Processed files: 1500/11955
Processed files: 2000/11955
Processed files: 2500/11955
...
...
```

A.3 Testovací mód

Posledním z módu je testovací, který provádí testovaní zvoleného modelu, který je uložený v souboru. Tento mód zvolený model načte do paměti a provede pouze jeho otestování výkonu. Tento proces provede pro všechny modely uloženy ve složce zadanou jako cestu k modelům. Tento mód by do programu přidán, aby nebylo potřeba pokaždé model znovu trénovat, pokud je potřeba pouze jeho otestování. Jinak testovací mód je součástí experimentálního mód. Testovací mód požaduje 4 vstupní argumenty název módu, cesta k datové sadě, název modelu a cestu ke složce, kde jsou uloženy námi naučeny modely.
Testovací mód spustíme příkazem:

```
python3 Main.py evaluation <dataset> <model> <slozka s modely>
```

Možné zadané modely(<model>):

- small-e-czech,
- fernet,
- robeczech

```
    $ python3 Main.py evaluation /mnt/data/czech_text_document_corpus_v20/
    small-e-czech
    ctk_models/Seznam/small-e-czech_epochs\=20_inputLen\=50_lr\=0.0001/
Successfully opened dynamic library libcudart.so.11.0
Hidden size: 256
Processing FOLD:  1
Load all name of text files
Loading data...
...
...

...
MICRO PRECISION: 0.8606962524004746 +/- 0.00544027794295209
MICRO RECALL: 0.8287063121959776 +/- 0.007296959463766781
MICRO F1: 0.8443596434321096 +/- 0.0030671855523575534
MACRO PRECISION: 0.8688198278713438 +/- 0.008271866652387298
MACRO RECALL: 0.8200669343371455 +/- 0.007768649783929356
MACRO F1: 0.84367191670769 +/- 0.0026229394750137756
EXACT MATCH: 0.6017065039924823 +/- 0.0032211278815973757
```

B Příloha

	pol	efin	pod	zak	mag	for	kul	spo	dpr	zdr	slz	mak	sta	ene	ekl	spl	met	bur	fin	obo	pit	odb	den	str	eur	zem	tur	sop	sko	aut	mix	bup	sur	ptr	med	prg	che
pol	0	168	1834	514	1333	1293	119	42	271	217	1360	242	336	149	240	101	46	16	99	53	66	206	137	32	213	121	102	135	133	20	51	7	46	46	106	111	21
efin	168	0	130	278	42	73	27	262	184	58	378	38	193	283	84	18	16	41	231	208	355	106	34	357	35	132	84	8	7	261	14	0	94	34	33	34	40
pod	1834	130	0	331	63	785	91	33	223	168	117	127	325	99	185	96	39	5	42	20	24	193	136	15	89	103	84	86	125	6	10	4	20	12	63	111	14
zak	514	278	331	0	92	299	53	100	286	87	136	41	98	64	77	42	24	9	45	64	90	17	64	30	58	32	33	57	44	23	32	4	21	14	47	45	37
mag	133	42	63	92	0	57	556	54	30	61	82	5	15	6	76	502	8	5	4	56	45	6	49	18	6	25	74	97	23	16	396	1	2	45	95	34	7
for	1293	73	785	299	57	0	21	27	73	124	34	174	73	62	101	5	6	8	65	20	23	120	33	14	95	70	10	64	61	11	41	6	18	36	62	30	15
kul	119	27	91	53	556	21	0	6	12	16	38	4	57	3	25	105	2	0	2	50	30	2	81	7	13	4	37	17	28	3	122	0	0	6	72	43	3
spo	42	262	33	100	54	27	6	0	1	6	322	7	8	0	0	31	1	0	27	7	3	3	70	0	3	1	14	2	6	0	25	0	0	1	13	6	0
dpr	271	184	223	286	30	73	12	1	0	11	40	11	149	16	31	31	273	1	8	9	5	85	11	34	35	5	33	5	10	0	2	0	8	0	10	52	8
zdr	217	58	168	87	61	124	16	6	11	0	22	12	14	0	27	40	20	0	34	84	8	220	38	3	121	115	19	30	10	1	17	0	0	207	24	52	25
slz	136	378	117	136	82	34	38	322	40	22	0	24	46	4	26	126	3	3	36	20	25	13	35	2	23	6	511	7	26	2	10	2	0	11	9	14	1
mak	242	38	127	41	5	174	4	7	11	12	24	0	36	9	17	1	4	32	197	41	12	138	15	2	132	5	13	17	16	1	2	40	4	3	10	12	43
sta	336	193	325	98	15	73	57	8	149	14	46	36	0	80	68	40	16	4	23	40	0	15	21	14	49	56	37	15	14	9	5	0	4	2	0	50	2
ene	149	283	99	64	6	62	3	0	16	0	4	9	80	0	221	5	22	127	6	11	15	12	9	24	50	127	4	1	3	10	0	1	324	1	6	15	120
ekl	240	84	185	77	76	101	25	0	31	27	26	17	68	221	0	50	32	1	2	14	23	0	23	28	61	123	19	4	16	24	40	0	20	4	8	26	12
spl	101	18	96	42	502	5	105	31	31	40	126	1	40	5	50	0	89	1	1	18	14	0	40	12	17	19	122	17	30	10	46	0	1	33	33	24	2
met	46	16	39	24	8	6	2	1	273	20	3	4	16	22	32	89	0	0	7	0	0	0	3	4	4	27	3	1	0	1	4	0	3	1	0	16	0
bur	16	41	5	9	5	8	0	0	1	0	3	32	4	127	1	1	0	0	25	15	11	3	1	11	29	8	3	1	0	8	1	249	156	2	1	0	119
fin	99	231	42	45	4	65	2	27	8	34	36	197	23	6	2	1	7	25	0	12	12	12	5	3	80	5	6	3	0	3	0	118	3	1	9	9	2
obo	53	208	20	64	56	20	50	7	9	84	20	41	40	11	14	18	0	15	12	0	92	7	6	132	76	28	9	20	0	122	18	2	12	99	23	13	23
pit	66	355	24	90	45	23	30	3	5	8	25	12	0	15	23	14	0	11	12	92	0	27	6	8	29	2	7	16	9	3	7	0	1	2	47	6	3
odb	206	106	193	17	6	120	2	3	85	220	13	138	15	12	0	0	3	3	12	7	27	0	14	35	7	2	8	46	14	24	3	0	7	0	1	13	57	0	3
den	137	34	136	64	49	33	81	70	11	38	35	15	21	9	23	40	3	1	5	6	6	14	0	5	5	15	29	24	28	2	7	0	1	13	57	0	3
str	32	357	15	30	18	14	7	0	34	3	2	2	14	24	28	12	4	11	3	132	8	35	5	0	29	4	0	3	2	431	1	0	14	0	0	0	6
eur	213	35	89	58	6	95	13	3	35	121	23	132	49	50	61	17	4	29	80	76	29	7	5	29	0	42	22	8	16	17	0	21	11	112	8	7	5
zem	121	132	103	32	25	70	4	1	5	115	6	5	56	127	123	19	27	8	5	28	2	2	15	4	42	0	6	0	0	0	8	0	0	97	5	15	16
tur	102	84	84	33	74	10	37	14	33	19	511	13	37	4	19	122	3	3	6	9	7	8	29	0	22	6	0	6	15	0	8	2	0	11	4	6	1
sop	135	8	86	57	97	64	17	2	5	30	7	17	15	1	4	17	1	1	3	20	16	46	24	3	8	0	6	0	26	2	24	1	0	6	25	14	0
sko	133	7	125	44	23	61	28	6	10	10	26	16	14	3	16	30	0	0	0	0	9	14	28	2	16	0	15	26	0	1	7	0	0	3	7	23	3
aut	20	261	6	23	16	11	3	0	0	1	2	1	9	10	24	10	1	8	3	122	3	24	2	431	17	0	0	2	1	0	1	0	2	0	0	0	5
mix	51	14	10	32	396	41	122	25	2	17	10	2	5	0	40	46	4	1	0	18	7	3	7	1	0	8	8	24	7	1	0	0	0	3	24	1	4
bup	7	0	4	4	1	6	0	0	0	0	2	40	0	1	0	0	0	249	118	2	0	0	0	0	21	0	2	1	0	0	0	0	0	0	0	0	0
sur	46	94	20	21	2	18	0	0	8	0	0	4	4	324	20	1	3	156	3	12	1	7	1	14	11	0	0	0	0	2	0	0	0	0	2	2	120
ptr	46	34	12	14	45	36	6	1	0	207	11	3	2	1	4	33	1	2	1	99	2	0	13	0	112	97	11	6	3	0	3	0	0	0	0	7	9
med	106	33	63	47	95	62	72	13	10	24	9	10	0	6	8	33	0	1	9	23	47	12	57	0	8	5	4	25	7	0	24	0	2	0	0	78	1
prg	111	34	111	45	34	30	43	6	52	52	14	12	50	15	26	24	16	0	9	13	6	31	0	0	7	15	6	14	23	0	1	0	2	7	78	0	0
che	21	40	14	37	7	15	3	0	8	25	1	43	2	120	12	2	0	119	2	23	3	0	3	6	5	16	1	0	3	5	4	0	120	9	1	0	0

Tabulka B.1: Celá podoba concurrenc matrix

	PRECISION	RECALL
Politika	81.013%	79.359%
Firmy	89.608%	83.103%
Politika_ČR	81.488%	76.613%
Kriminalita a právo	90.795%	70.647%
Magazínový výběr	86.401%	73.089%
Parlamenty a vláda	85.460%	68.337%
Kultura	87.562%	81.382%
Sportovní zpravodajství	96.291%	92.651%
Doprava	94.060%	82.884%
Zdravotnictví	93.436%	81.599%
Služby	89.145%	77.053%
Makroekonomie	92.491%	66.934%
Stavebnictví a reality	84.099%	56.697%
Energie	94.501%	85.230%
Životní prostředí	82.464%	52.812%
Životní styl	76.824%	48.729%
Počasí	94.188%	87.993%
Burzy	98.543%	89.155%
Finanční služby	94.610%	70.821%
Obchod	86.677%	51.366%
Telekomunikace a IT	93.946%	76.115%
Práce a odbory	92.017%	67.312%
Zpravodajské deníky	99.327%	98.179%
Strojírenství	95.688%	79.791%
Evropská unie-zprávy	91.100%	41.314%
Zemědělství	93.307%	62.355%
Cestovní ruch	87.741%	71.839%
Sociální problematika	80.503%	38.784%
Školství	97.995%	66.447%
Automobilový průmysl	95.835%	89.703%
Mix	80.623%	35.344%
Burzy peněžní	97.150%	82.618%
Suroviny	97.211%	68.232%
Potravinářství	95.676%	43.946%
Media a reklama	96.667%	17.426%
Pragensie	95.556%	7.226%
Chemický a Farmaceutický průmysl	87.509%	32.283%

Tabulka B.2: Tabulka hodnot Precision a Recall pro 37 kategorií