FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Modul pro detekci kontextů v doméně internetového bankovnictví v českém jazyce
Student:	Bc. Samuel Fabo
Vedoucí:	Ing. Stanislav Kuznetsov
Studijní program:	Informatika
Obor / specializace:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Cílem práce je vytvoření modulu pro intent detection v doméně internetového bankovnictví. Úloha detekce kontextů (intent detection) patří do souboru technik zpracování přirozeného jazyka (angl. natural language processing – NLP) a jeho porozumění (angl. natural language understanding – NLU). Hlavním úkolem studenta bude vytvoření vhodných datasetů v českém jazyce a jejich obohacení o nové třídy z této oblasti. Dále student navrhne vhodné modely pro úlohu intent detection. Celé řešení předvede na demu chatbota.

1) Proveďte rešerši v oblasti „intent detection".
2) Vyberte vhodné datasety v cizích jazycích, které byste mohli použít pro obohacení vašeho českého datasetu.
3) Proveďte předzpracování dat.
4) Popište a implementujte metody pro porovnání kvality datasetu.
5) Vyberte a implementujte modely pro detekci kontextu.
6) Výsledky práce předveďte na demu chatbota.

Elektronicky schválil/a Ing. Karel Klouda, Ph.D. dne 22. února 2022 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práca

Modul pro detekci kontextů v doméně internetového bankovnictví v českém jazyce

Bc. Samuel Fabo

Katedra aplikované matematiky
Vedúci práce: Ing. Stanislav Kuznetsov

4. mája 2022

Poďakovanie

V prvom rade by som chcel poďakovať vedúcemu práce Stanislavovi Kuznetsovovi za vedenie práce a správne nasmerovanie, a svojej rodine, ktorá za mnou stála a podporovala ma počas celého štúdia na vysokej škole. Ďalej by som chcel poďakovať Petrovi Paščenkovi a firme Profinit za možnosť realizácie celej práce.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. V súlade s ustanovením § 46 odst. 6 tohoto zákona týmto udeľujem bezvýhradné oprávnenie (licenciu) k užívaniu tejto mojej práce, a to vrátane všetkých počítačových programov ktoré sú jej súčasťou alebo prílohou a tiež všetkej ich dokumentácie (ďalej len „Dielo“), a to všetkým osobám, ktoré si prajú Dielo užívať. Tieto osoby sú oprávnené Dielo používať akýmkoľvek spôsobom, ktorý neznižuje hodnotu Diela (vrátane komerčného využitia). Toto oprávnenie je časovo, územne a množstevne neobmedzené.

V Prahe 4. mája 2022 . . . . . .

České vysoké učení technické v Praze
Fakulta informačních technologií

Odkaz na túto prácu

Fabo, Samuel. *Modul pro detekci kontextů v doméně internetového bankovnictví v českém jazyce*. Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022. Dostupný aj z WWW: ⟨`https://gitlab.fit.cvut.cz/fabosamu/ni-dip/`⟩.

Abstrakt

V tejto práci sa zaoberáme výskumom a aplikáciou rôznych techník na riešenie problému intent detection (alebo aj detekcia kontextov, zámerov) v doméne českého bankovníctva. Intent detection je základ každého dobrého chatbota a ak je detekcia kvalitná a vyladená, udrží užívateľa dlhšie v kontakte so strojom. Keďže neexistujú voľne dostupné dátové sady v českom jazyku na túto doménu, museli sme dáta zozbierať sami. Neskôr sme spojili zozbierané vzorky intentov s voľne dostupnou sadou BANKING77, ktorú sme preložili do češtiny. Podarilo sa nám vyladiť model, ktorý mal na testovacej vzorke spojenej dátovej sady dobré výsledky presnosti. Nakoniec sme nasadili výsledný model do demonštračnej aplikácie.

Kľúčové slová intent detection, detekcia zámerov, detekcia kontextov, chatbot, strojové učenie, spracovávanie prirodzeného jazyka, porozumenie prirodzenému jazyku, klasifikácia textu, predspracovanie textu

Abstract

In this thesis, we research and apply various techniques to solve the intent detection problem in the Czech internet banking domain. The intent detector is a fundamental part of each chatbot and keeps the user longer in contact with the machine if a high-quality, fine-tuned detector is used. We needed to gather the training data on our own because there are no publicly available datasets in the Czech language for this domain. Later on, we merged gathered samples of intents with the publicly available dataset BANKING77, which we translated into the Czech language. We succeeded in fine-tuning a model, which had good accuracy results on the test set. We deployed the model to the production version of the demo application.

Keywords intent detection, chatbot, machine learning, natural language processing, natural language understanding, text classification, text preprocessing

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Motivácia a ciele práce

Internetové bankovníctvo je v dnešnej dobe veľmi často používané. Nejaký druh aplikácie, ktorou vie klient banky komunikovať s bankou samotnou, má dnes snáď už každá banka. Napriek tomu sa mnohí užívatelia v aplikáciách vedia len ťažko orientovať, prípadne chcú vykonať akýkoľvek úkon rýchlejšie.

S rozvojom umelej inteligencie a strojového učenia sa stáva komunikácia človeka so strojom jednoduchšia, ako kedykoľvek predtým a začali vznikať mnohé konverzačné systémy a tzv. chatboty, ktoré sú vľúdnejšie a príjemnejšie na používanie, pretože s nimi človek vie komunikovať vlastnou prirodzenou rečou. Preto sme sa vo firme Profinit [1] rozhodli, že vytvoríme chatbota, ktorý bude najskôr schopný rozoznávať slovné, písané požiadavky a príkazy klientov banky a tým ukážeme schopnosť firmy takýto systém vytvoriť a následne nasadiť.

Hlavným cieľom tejto práce je postavenie základného kameňa akéhokoľvek chatbota, a to síce detekcia kontextu – zámeru, intentu – užívateľa komunikujúceho svojím materinským jazykom (v tomto prípade čeština) v doméne bankovníctva. Táto aplikácia je demonštráciou úspešného spojenia umelej inteligencie a softvérového inžinierstva vo svete financií.

Štruktúra

V tejto práci sa zaoberáme prieskumom rôznych moderných metód spracovania prirodzeného jazyka, ktoré sú síce stále na úrovni prvotného výskumu (State of The Art), avšak použiteľné v praxi. Popíšeme históriu a súčasnosť spracovávania prirodzenej reči (v písanej, textovej forme), predstavíme si rôzne druhy tzv. chatbotov – konverzačných systémov – a typy algoritmov strojového učenia. Neskôr popíšeme rôzne typy neurónových sietí, základy modelovania

[1] `profinit.eu`

jazyka (angl. language modeling) a reprezentáciu textu vo vektorovom priestore. Na záver rešerše popíšeme tzv. transformer architektúry využívané predovšetkým na prevodné učenie (angl. transfer learning).

V praktickej časti predstavíme a analyzujeme použité dátové sady, popíšeme aplikáciu na zber trénovacích dát ako aj samotnú demonštračnú aplikáciu chatbota. Nájdeme najvhodnejší model pre intent detection a popíšeme výsledky najlepšieho modelu, ktorý tento problém rieši. Na koniec predstavíme demonštračnú aplikáciu chatbota s implementovanou detekciou zámeru užívateľa.

Časť I

Teoretická časť

Kapitola 1

Spracovanie prirodzenej reči

Spracovanie prirodzenej reči (anlg. Natural Language Processing (NLP)) je vedecká disciplína zaoberajúca sa prirodzenou ľudskou rečou, spracovaním textu, a hovoreného slova. Jedna z prvých zmienok siaha až do roku 1950, kedy Alan Turing sformuloval tzv. Turingov test, ktorý sa pokúša dať odpoveď na otázku či je nejaký stroj inteligentný. [6]

Táto disciplína je veľmi široká a pokrýva témy, ako napríklad odpovedanie na otázky (question answering), strojový preklad (machine translation), sumarizácia textu (text summarization) alebo generovanie popisu pre nejaký obrázok (image text generation). Popíšme jednotlivé obdobia vývoja NLP:

- *Ručne písané pravidlá* – populárna technika už pri počiatkoch počítačov, keď nebolo k dispozícii dostatok výpočtovej sily. Pravidlá – podmienky – boli tvorené ručne a expertne. Hlavnou výhodou bola možnosť absolútnej kontroly výstupu, takže nájsť problematické miesta aplikácie a následne ich opraviť nebolo ťažké. Oproti tomu bolo veľmi intelektuálne aj časovo vyčerpávajúce vytvoriť zoznam všetkých možných otázok a odpovedí. Preto sa dnes už táto metóda skoro nepoužíva. Priekopníkom chatbotov sa stala tzv. ELIZA, ktorá bola vyvinutá pod taktovkou Josepha Weizenbauma z MIT [7] v roku 1966.

- *Štatistický prístup* – používaný okolo prelomu milénií (približne 1990-2010). So vznikom mnohých webových stránok a rozširovaním internetu do rôznych kútov sveta vzniklo veľké množstvo textu, ktorý bol voľne dostupný. Tak vznikali tzv. korpusy, ktoré mohli byť použité napríklad na strojové preklady pomocou techník nesupervizovaného učenia. Samotné učenie sa stalo možným vďaka zvyšovaniu výkonu počítačov a superpočítačov.

- *Moderné NLP* – v posledných rokoch získalo mnoho úspechov aj vďaka vysokým výkonom moderných počítačov, používania grafických kariet

na náročné výpočty. Od popisu perceptronu [8] až po jeho „znovuobjavenie“ a rôzne úpravy a variácie použité na množstvo rôznych problémov. Viac o neurónových sieťach si povieme v nasledujúcej sekcii.

V tejto práci sa budeme zaoberať hlavne prístupom štatistickým, kde kvalita výsledkov je závislá na kvalite a štatistickému – strojovému – porozumeniu dát. Intervencia človeka je minimálna.

1.1 Chatboty historické a dnešné

Od čias, keď vznikli počítače, bola túžba ľudí po čo najjednoduchšej komunikácii s nimi bola neprehliadnuteľná. Samotný Alan Turing sníval o tom, že raz budú počítače tak inteligentné, že sa vyrovnajú intelektu človeka a človek sám nebude schopný rozpoznať či komunikuje so strojom alebo počítačom. O tom svedčí jeho článok z 1950, kde popísal tzv. imitačnú hru, neskôr známu ako turingov test (umelej inteligencie). [9]

1.1.1 Chatbot založený na pravidlách

Pravidlá môžu mať rôznu podobu, ako napríklad práve šablónovanie a použitie regulárnych výrazov ale aj založené na vyhľadávaní kľúčových slov z užívateľského vstupu. Pomocou týchto pravidiel tvoria odpovede podľa predpísaného scenáru, podobne ako sa herec drží svojho scenáru.

Uveďme príklad, kedy užívateľ chce zmeniť svoj pin-kód na bankomatovej karte. Možný vstup by mohol byť „Prosím, chcem si *zmeniť* svoj *pin* na *karte*.“, kde stroj – chatbot – by podľa výskytu kľúčových slov v texte (povedzme: „zmeniť“, „pin“, „karte“)

Prvý pokus o podobný stroj bol vyvinutý v laboratóriách MIT Josephom Weizenbaumom, ktorý dostal názov ELIZA. [7] Pomocou šablón a regulárnych výrazov – tzv. pattern matching – mala demonštrovať schopnosť komunikácie stroja s človekom. Radí sa teda medzi prvé chatboty (v tom čase nazývané tzv. „chatterboty“)

Turingov test bol však nahradený mnohými novými spôsobmi merania „inteligencie“ stroja, ako napríklad čínska miestnosť [10], ktorá však bola kritizovaná [11] ale aj podporovaná [12].

1.1.2 Dnešné účelové chatboty

Dnes sa chatboty vyznačujú hlavne tým, že plnia priania a príkazy užívateľov, a teda výsledkom „konverzácie“ užívateľa a stroja je splnená požiadavka na vykonanie nejakého špecifického úkonu. Oproti ELIZE alebo iných moderných čisto konverzačných chatbotov sa líši práve tým, že poskytnú riešenie nejakého problému a užívateľovi takýmto spôsobom vedia uľahčiť alebo aj zjednodušiť prácu. Môže ísť o textové alebo hlasové ovládanie telefónu, inteli-

gentnej domácnosti alebo aj internetového bankovníctva. Existujú aj chatboty určené čisto na konverzáciu s užívateľom, avšak v tejto práci sa nimi nezaoberáme.

Medzi dnešné – moderné – konverzačné systémy môžeme zaradiť napríklad Siri od spoločnosti Apple [13] pomocou ktorej vie užívateľ ovládať (nielen) hlasom svoj telefón, Google Assistant [14] ktorý dnes vie užívateľovi napríklad rezervovať termín u kaderníka prostredníctvom automatického telefonického hovoru alebo Erica [15] – virtuálny bankový asistent, pomocou ktorého vie klient banky odoslať peniaze na iný účet, zobraziť informácie o pohyboch na konte a množstvo iných funkcií (Erica sa dá ovládať aj hlasom).

Najdôležitejšia súčasť akéhokoľvek účelového chatbota je tzv. detekcia zámeru (úmyslu, intentu) užívateľa. Od toho sa neskôr odvíja samotná konverzácia, prípadne vykonanie príkazu alebo práce.

1.2 Intent detection

Alebo detekcia kontextov, zámerov, úmyslov (angl. *„intent detection“*). Jedná sa o problém klasifikácie textu (dokumentu) do $\boldsymbol{c}$ tried (intentov). Tento problém je v mnohých ohľadoch veľkou výzvou [16], hlavne preto, že na úspešnosti tohoto „odhaľovania“ zámeru/príkazu užívateľa závisí celá nasledujúca konverzácia užívateľa s chatbotom. Ak systém detekcie zlyhá, je to prvý kontaktný bod s užívateľom daného systému a môže ho to odradiť od ďalšieho používania. [17]

Problém je definovaný nasledovne: Majme množinu X, kde každý prvok je sekvencia znakov rôzne dlhá (pri intent detection je to jedna alebo viac viet v prirodzenom jazyku). Ďalej máme k dispozícii vektor Y, kde každý prvok je číselné priradenie do správnej triedy. Jedná sa teda o supervizované učenie, klasifikačný problém viacerých tried (bližšie v sekcii 2.1).

Odkedy začali byť počítače dostupnejšie, vznikali rôzne dátové sady určené na detekciu intentu, napríklad pre rezerváciu letu [18] alebo získanie informácie o verejnej doprave [19]. Dát pre tréning akéhokoľvek moderného modelu dnes stále nie je dosť, takže tento problém často naberá scenár tzv. few-shot learningu, kde k dispozícii je len obmedzený počet vzorkov pre intent [17] [20]. V kontexte konverzačných systémov je intent detection spájaný dohromady aj s tzv. rozpoznávaním entít v texte (angl. entity recognition) [21] [20] [22].

Niektoré výskumy sa sústreďujú priamo na jednotlivé moduly detekcie intentu (konkrétne transformery a modely podobné BERTovi, viď. sekciu 2.7) a skúšajú vylepšiť tieto modely pomocou rôznych techník pre prevedenie few-shot learningu do praxe [23]. V ďalších prácach sa pokúšajú riešiť úplnú absenciu dát pomocou tzv. zero-shot learningu, kde len na základe názvov intentov chcú daný vstup od užívateľa správne klasifikovať [24].

1.2.1 Vhodné dáta

V tejto práci sa zaoberáme hlavne získaním kvalitných a reálnych dát od užívateľov, a to špecificky pre doménu bankovníctva. Pokúšame sa vytvoriť scenár few-shot learningu a riešiť tento problém aj s menším počtom vzoriek (priemerne cca. 100 vzorkov pre intent/triedu). Vhodné modely na tento scenár sú práve predtrénované siete podobné BERTovi [25], so síce veľkým počtom parametrov, avšak so silnou štatistickou znalosťou jazyka. Ladenie takýchto modelov dosahuje dobré výsledky a je jednoduchšie ako tréning veľkého modelu odznova.

Existuje iba niekoľko dátových sád pre intent detection, ako napríklad HWU64 [26], kde sa nachádza 64 rôznych tried (zámerov), z 21 rôznych domén s počtom vzorkov 25 716. Ďalšia dátová sada hodná spomenutia pomenovaná CLINC150 [27] obsahuje 150 tried (zámerov) z 10 domén s počtom vzoriek 23 700. Tieto dátové sady sú však len ťažko schopné pokryť šírku domén, ktorých sa dotýkajú [17] a takisto sa nešpecializujú na doménu bankovníctva, takže použité byť nemohli. Ďalšou veľkou prekážkou pri budovaní akéhokoľvek účelového konverzačného systému orientovaného na jednu doménu je pri jazykoch iných ako angličtina práve nedostupnosť dát pre tvorbu akéhokoľvek systému [17].

Vytvorili sme preto spolu s kolegami vo firme Profinit vlastnú dátovú sadu pomocou Text2Bank aplikácie. Túto sadu sme nazvali TEXT2BANK13. Keďže táto sada má malý počet vzoriek a aj intentov – tried, bolo nutné ju obohatiť o ďalšie vzorky. Preto sme použili verejne dostupnú dátovú sadu BANKING77 [17], ktorú sme strojovo preložili a po predspracovaní zlúčili s TEXT2BANK13 pre vytvorenie sady vhodnej pre strojové učenie a demonštráciu finálnej verzie aplikácie Text2Bank. Tieto dáta sú popísané v praktickej časti tejto práce.

Kapitola 2

Aloritmy strojového učenia

V tejto kapitole popíšeme typy a konkrétne implementácie niektorých algoritmov strojového učenia (angl. machine learning).

2.1 Typy strojového učenia

Strojové učenie môžeme rozdeliť na štyri hlavné disciplíny [28] [29]:

Supervizované učenie

Je taktiež nazývané tzv. strojové učenie s učiteľom. Na základe historických dát – príznakov – a k nim pridruženým vektorom príslušností príznaku do danej triedy – máme za úlohu predikovať nové príznaky do správnej triedy.

Modelový príklad pre vysvetlenie: máme k dispozícií informácie o pacientoch, u ktorých sme našli príznaky chrípky a zároveň lekári pacientov vyšetrili, pozorovali a u každého určili, či chrípku mal alebo nemal. Máme teda zoznam "príznakov"(angl. features) a ku príznakom každého pacienta máme k dispozícii jeho finálny stav (mal alebo nemal chrípku). Supervizované učenie v tomto prípade znamená, že vytvoríme tzv. model, ktorý na základe predošlých dát určí, či nový pacient s iným zoznamom príznakov má alebo nemá chrípku a s akou pravdepodobnosťou. Tento modelový príklad zaradíme do problémov binárnej klasifikácie. Na predikciu môžeme použiť napríklad model logistickej regresie. [30]

V kontexte spracovania prirodzenej reči tu radíme spomínaný intent detection, ktorý sa prevádza na problém klasifikácie textu, a je popísaný v sekcii 1.2.

Nesupervizované učenie

Podobne ako pri supervizovanom učení máme k dispozícii príznaky, avšak vektor príslušnosti príznaku tu úplne chýba. V tomto prípade chceme nájsť v

dátach štruktúru pomocou štatistických alebo iných metód, prípadne porozumieť dátam a vhodne ich popísať. Tento typ sa bežne voláme strojové učenie bez učiteľa.

Semi-supervizované učenie

V tomto type máme k dispozícii veľké množstvo príznakov bez príslušnosti do danej triedy, ale k tomu len relatívne malé množstvo dát, kde poznáme triedu, do ktorej daný príznak patrí. Tento typ učenia je náročnejší, práve kvôli nedostatku labelovaných dát.

Reinforcement learning

Alebo tzv. posilňovacie učenie používa pozorovania zozbierané z interakcii s prostredím, a pomocou fitness funkcie sa algoritmus učí sekvenčne. Chýba tu vektor príslušnosti (na rozdiel od supervizovaného učenia) a zároveň model vie získavať spätnú väzbu z prostredia (na rozdiel od nesupervizovaného).

2.2 Naive Bayes classifier

Metódy naivného bayesa (angl. Naive Bayes (NB)) sú založené na aplikácii bayesovskej vety s naivným predpokladom nezávislosti príznakov $X = (X_1, \ldots, X_p)^T$ s podmienkou vysvetľovanej premennej $Y = y$. T.j. $\forall y \in \mathcal{Y}$ a $\boldsymbol{x} \in (x_1, \ldots, x_p)^T \in \mathcal{X}$ platí:

$$\mathrm{P}(\boldsymbol{X} = \boldsymbol{x}|Y = y) = \mathrm{P}(X_1 = x_1|Y = y)\cdot, \ldots, \cdot\mathrm{P}(X_p = x_p|Y = y).$$

Naivita znamená, že pre fixnú hodnotu vysvetľovanej premennej predpokladáme, že sú príznaky nezávislé. Výsledný Maximum a Posteriori (MAP) odhad (alebo predikcia $\hat{Y}$) bayesovského klasifikátora je teda:

$$\hat{Y} = \arg\max_{y \in Y} \prod_{i=1}^{p} \mathrm{P}(X_i = x_i|Y = y)\mathrm{P}(Y = y).$$

Ak je daný príznak X spojitou náhodnou veličinou, tak sa namiesto podmienenej pravdepodobnosti pre tento príznak vezme podmienená hustota pravdepodobnosti $f_{X|y}(x)$, čo je hustota pravdepodobnosti veličiny X podmienená javom $Y = y$ a odpovedá distribučnej funkcii $F_{X|y}(x) = \mathrm{P}(X_i \leq x_i|Y = y)$. Predikciu MAP odhadom prevedieme pomocou vzťahu

$$\hat{Y} = \arg\max_{y \in Y} \prod_{i=1}^{l} \mathrm{P}(X_i = x_i|Y = y) \prod_{i=l+1}^{p} f_{X_i|y}(x_i)\mathrm{P}(Y = y),$$

kde $X_1, \ldots, X_l$ sú diskrétne príznaky a $X_{l+1}, \ldots, X_p$ sú príznaky spojité. Tento klasifikátor je bežne používaný na klasifikáciu dokumentov a textu (napr.

spam detection) pomocou tzv. bag-of-words, avšak nahradili ho inteligentnejšie spôsoby klasifikácie. My tento model v tejto práci trochu ohneme a používame práve na zber dát, pretože jeho hlavný prínos je ľahká interpretovateľnosť výsledkov predikcie. [31] [32]

Častým modelom podmieneného rozdelenia je v spojitom prípade normálne rozdelenie $N(\mu_y, \sigma_y^2)$ so strednou hodnotou určenou parametrom μ_y a rozptylom σ_y^2. Podmienená hustota je teda $\forall y \in \mathcal{Y}$

$$f_{X|y}(x) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right)$$

a obvykle sa používajú Maximal Likelihood Estimation (MLE) odhady na výpočet $\hat{\mu}$ a $\hat{\sigma_y^2}$. Implementácia v knižnici `scikit-learn` je dostupná v balíčku `GaussianNB`. [33] [31] [32]

2.3 KMeans clustering

Keď hovoríme o zhlukovaní (angl. clustering), myslíme úlohu nesupervizovaného učenia. Algoritmus KMeans funguje iteratívne: Najprv inicializujeme model tak, že rozmiestnime k stredových bodov $\mu_1, \ldots, \mu_k$ do priestoru z $\mathbb{R}^p$, kde chceme nájsť k zhlukov. Iteratívne potom:

1. roztriedime body do zhlukov $C_i = \{x \in D | i = \arg\min_j ||x - \mu_j||\}$,
2. prepočítame body $\mu_1, \ldots, \mu_k$ ako geometrické stredy týchto zhlukov: $\mu_i \to \frac{1}{|C_i|} \sum_{x \in C_i} x$.

Beh algoritmu zastavíme, akonáhle je zmena hodnoty účelovej funkcie

$$G(C) = \sum_{i=1}^{k} \frac{1}{2|C_i|} \sum_{x,y \in C_i} (||x_i - \mu_j||^2)$$

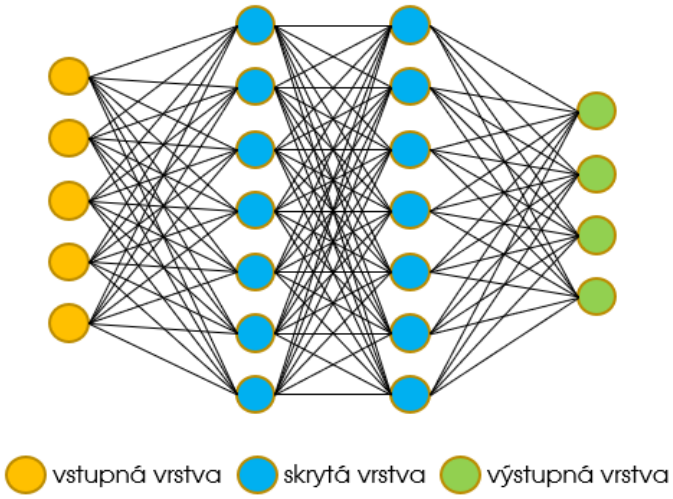
medzi jednotlivými iteráciami dostatočne malá. Implementácia bola použitá z knižnice `sklearn`. [34] [33]

KMeans sme v tejto práci použili pri zlučovaní dátových sád a pri analýze jednotlivých vzoriek intentov.

2.4 UMAP

Alebo angl. Uniform Manifold Approximation and Projection (jednotná priestorová aproximácia a projekcia) je **technika redukcie dimenzionality**, ktorú môžeme použiť podobne ako t-SNE [35], ale aj všeobecne pre nelineárnu redukciu dimenzie. Algoritmus funguje na troch rôznych predpokladoch o dátach:

1. dáta sú uniformne rozdelené na riemannovom priestore,



Obr. 2.1: Zjednodušená ukážka neurónovej siete [1]

2. riemannova metrika je lokálne konštantná (môže byť aproximovaná),

3. priestor je lokálne prepojený.

Z týchto predpokladov je možné modelovať priestor s fuzzy topologickou štruktúrou. Pre dáta s nízkodimenzionálnou projekciou vie UMAP nájsť embeddingy dát, ktoré majú najbližšiu možnú fuzzy topologickú štruktúru. [36]

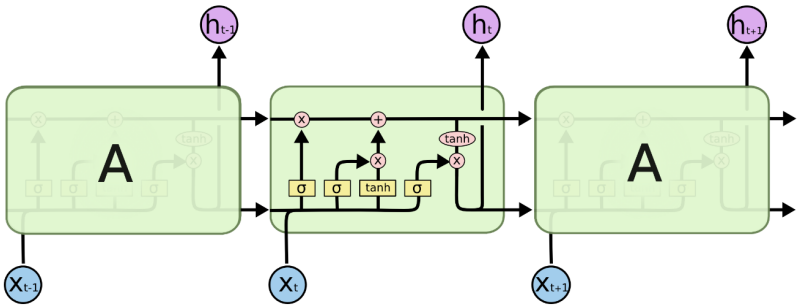
Redukciu dimenzie v tejto práci používame hlavne na vizualizáciu viac dim. priestorov do 2 dimenzií, kde sa dáta ľahšie interpretujú.

2.5 Neurónové siete

Neurónová sieť sa snaží simulovať to, čo veríme, že robí náš mozog. Samozrejme, že sú s tým spojené mnohé limitácie a nepreskúmané možnosti. Proces učenia simulujeme pomocou matice váh W, ktorá reprezentuje jednu vrstvu v neurónovej sieti. Týchto vrstiev môže byť v neurónovej sieti viac za sebou tak, aby sme získali robustný model. Vrstvy za vstupnou nazývame skryté. Základná rovnica pre jednu vrstvu siete je

$$o_l = f(W_l i_l + b_l),$$

kde o_l je výstupný vektor na l-tej vrstve dimenzie $R^{o\times 1}$, i_l je vstupný vektor dim. $R^{i\times 1}$, f je nelineárna aktivačná funkcia b_l je bias pre l-tú vrstvu a W_l je matica váh tejto vrstvy. Proces učenia je v tomto prípade určenie hodnôt matice W. Zjednodušenú verziu siete môžeme vidieť na obrázku 2.1. Zvyčajne sa váhy určia pomocou algoritmu spätnej propagácie. [37]



Obr. 2.2: Opakujúci sa modul v LSTM so štyrmi navzájom prepojenými vrstvami. [2]

2.5.1 Rekurentné Neurónové siete

Rekurentné siete (angl. Recurrent Neural Network (RNN)) sú špeciálne typy sietí, ktoré si vedia dobre poradiť so sekvenčnými dátami. Jej vstupom je sekvencia $(x_1, x_2, \ldots, x_N)$ a výstupom je sekvencia skrytých stavov $(h_1, h_2, \ldots, h_N)$, kde

$$h_t = f\left(W_x x_t + W_h h_{t-1} + b_n\right).$$

Skrytý stav reprezentuje kus špecifickej informácie o vstupe. Tento klasický prístup RNN však nie je schopný si pamätať dlhšie závislosti v sekvenciách. Preto sa používa vylepšená verzia, tzv. Long Short Term Memory (LSTM). Pridaná „pamäťová“ bunka, ktorá sa tu nachádza, lepšie modeluje dlhšie závislosti. Klasická architektúra LSTM je zložená z: [37]

- vstupnej brány: $i_t = f\left(W_i x_t + W_i h_{t-1} + b_i\right)$,
- zabúdacej brány: $f_t = f\left(W_f x_t + W_f h_{t-1} + b_f\right)$,
- výstupnej brány: $o_t = f\left(W_o x_t + W_o h_{t-1} + b_o\right)$,
- vnútorného stavu: $u_t = \tanh\left(W_u x_t + W_u h_{t-1} + b_u\right)$,
- pamäťovej bunky: $c_t = i_t \times u_t + f_t \times c_{t-1}$.

Násobenie po prvkoch je značené ako $\times$, x_t je vstupný vektor o veľkosti d v čase t a W je matica váh, ktoré sa sieť musí naučiť. Učenie prebieha pomocou gradientného zostupu. Základnú architektúru modelu môžeme vidieť na obrázku 2.2. [2] [37]

2.5.2 Obojsmerné LSTM (BiLSTM)

Obyčajné LSTM majú svoje limity a stále neboli dostatočne dobré pre spracovanie prirodzenej reči. Myšlienka obojsmerného spracovania priniesla možnosť tréningu za použitia všetkých minulých a budúcich častí vstupnej sekvencie,

a to v zadanom časovom okne. V skratke sa dá povedať, že BiLSTM sú dve nezávislé LSTM siete, avšak každá sa pozerá do kontextu okolia slova v čase t opačným smerom (dopredu a dozadu) a teda sú trénované nezávisle na sebe (pomocou back propagation). Výstup v čase t je pripojením h_t a h'_t. Počet parametrov siete je teda oproti LSTM dvojnásobný. [38]

2.6 Reprezentácia slov a viet vo vektorovom priestore

Populárny prístup k modelovaniu jazyka je transformácia slov do vektorov. Vytvorené vektory v sebe držia skryté informácie o reči, ako sú analógie alebo sémantika. Ak by sme chceli takéto vektory (embeddingy) vytvoriť, potrebujeme na to veľký obnos dát – tzv. korpus (napríklad Wikipedia [39]), kde sa nachádzajú milióny slov v rôznych kontextoch (a jazykov, takže sa dá modelovať každý jazyk zvlášť). Jedna z najpopulárnejších techník sa nazýva Word2Vec. [40]

2.6.1 Word2Vec

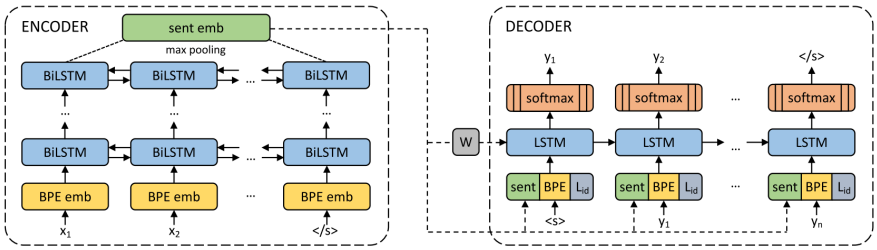
FastText dodáva dva modely pre výpočet slovných embeddingov (vektorov, reprezentácií): tzv. skipgram a Continuous Bag Of Words (CBOW). Skipgram sa učí predikovať cieľové slovo pomocou slov v jeho okolí a naopak CBOW predikuje cieľové slovo pomocou kontextu – okolia. Tento kontext je reprezentovaný ako množina slov vo fixne veľkom „okne“ okolo cieľového slova. [2] skipgram sa viac hodí pre menšie korpusy a je schopný dobre reprezentovať vzácne slová, kdežto CBOW sa rýchlejšie trénuje (vhodnejší pre veľké korpusy) a lepšie reprezentuje frekventované slová. Každé slovo z korpusu predtým rozdelíme na tzv. n-gramy, podslová. [3] Pre učenie reprezentácií na skrytej vrstve tohto modelu sa používa stochastický gradientný zostup, kde minimalizujeme

$$\sum_{t=1}^{T}\left[\sum_{c\in\mathcal{C}_t}\ell\left(s\left(w_t,w_c\right)\right)+\sum_{n\in\mathcal{N}_{t,c}}\ell\left(-s\left(w_t,n\right)\right)\right],$$

kde T je počet slov v korpuse (sekvencia slov $w_1,\ldots,w_T$), C_t sú indexy slov okolo w_t, $l(.)x \rightarrow \log(1+e^{-x})$ je stratová logaritmická funkcia, $s:(w,w_c)\rightarrow R$ je skórovacia funkcia pre slovo a jeho kontext a $\mathcal{N}_{t,c}$ sú náhodné negatívne vzorky kontextu c. Takýmto spôsobom vieme pomocou rekurentnej neurónovej siete získať reprezentácie slov – embeddingy – z jej skrytej vrstvy. [40]

[2] Problém predikcie cieľového slova je modelovaný ako množina nezávislých binárnych klasifikácií

[3] napr. pre $n = 3$ slovo kdepak $= \{<\text{kd}, \text{kde}, \text{dep}, \text{epa}, \text{pak}, \text{ak}>, <\text{kdepak}>\}$ kde <> sú špeciálne znaky začiatku a konca slova



Obr. 2.3: Architektúra LASERu pre učenie viacjazyčných embeddingov [3]

2.6.2 FastText

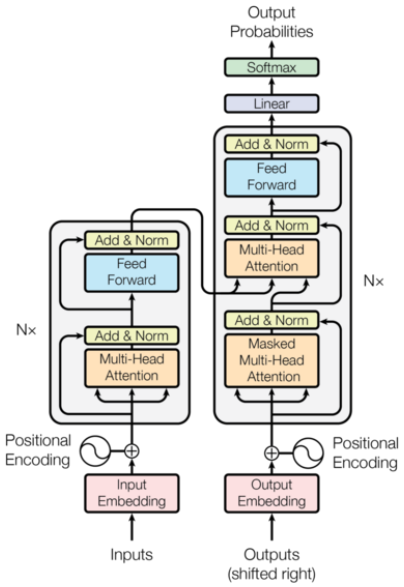
Je knižnica, ktorá používa model word2vec s n-gramami a špeciálnymi symbolmi pre začiatok a koniec slova pre rozlišovanie predpôn a prípon. Veľkosť slovníka je nastavená na $K = 2.10^6$ a každé slovo je reprezentované indexom v slovníku a množinou n-gramov. K dispozícii sú predtrénované modely na Wikipédii a Common Crawl korpuse v požadovanom jazyku (v našom prípade čeština a angličtina) [41]. Použitý bol model CBOW s $n = 5$, dimenziou skrytej vrstvy $d = 300$ (a teda aj veľkosť embeddingov je 300) a s oknom veľkosti 5 slov. [42] Tieto reprezentácie – embeddingy – sa získavajú pomocou trénovania rekurentnej neurónovej siete s 650 LSTM jednotkami [43].

FastText vieme použiť aj na získanie tzv. vetných embeddingov. Po určení slovných embeddingov z vety a po normalizácii ich vektorov vieme pomocou priemeru všetkých vektorov získať nový – vetný embedding (v rovnakom priestore, ktorý bol modelovaný slovnými embeddingami). [43]

2.6.3 LASER

Tým, že je bolo modelovanie jazyka od uvedenia word2vec spopularizované, boli vytvorené mnohé ďalšie nástroje pre reprezentáciu slov a viet prirodzeného jazyka. Pre potrebu tvorby embeddingov viet rôznych jazykov tak, aby sa výsledné vektory v priestore nachádzali blízko seba bol vyvinutý LASER – Language Agnostic SEntence Representations. Tento nástroj z dielne Facebooku vie reprezentovať vety z viac ako 90 jazykov a 28 rôznych znakových sád. [3]

Architektúra tohto nástroja je enkodér-dekodér, kde enkodér je zložený z 5 vrstiev BiLSTM [38]. LASER funguje na princípe podobnom, ako tzv. „sequence-to-sequence“ translácia, avšak chýba tu attention vrstva, ktorá je nahradená vektorom o fixnej veľkosti 1024. Tento vektor reprezentuje vstupnú sekvenciu. Embeddingy sú potom získavané aplikáciou operácie max-poolingu z tejto poslednej skrytej vrstvy enkodéra. Túto architektúru môžeme vidieť na obrázku 2.3. Keďže jazyk sekvencie nie je indikovaný pre enkodér ale len



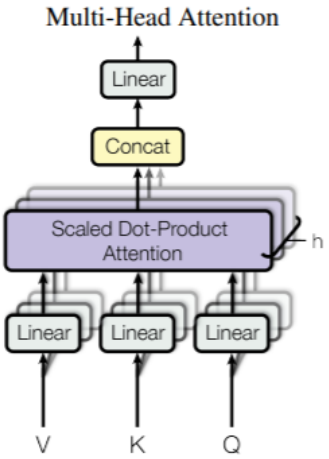
Obr. 2.4: Architektúra transformeru, naľavo enkodér pre vstupy, napravo dekodér pre výstupy (N krát za sebou). Podvrstvy každej z týchto dvoch častí sú pozornosť (attention) a dopredná neurónová sieť nasledujúca hneď za ňou.

pre dekodér, enkodér je nútený sekvencie ukladať na približne podobné miesto v priestore embeddingov. [3]

2.7 Transformery

V roku 2017 bola po prvýkrát popísaná architektúra takzvaného transformeru. Výskum sa vtedy zameriaval na strojový preklad a táto architektúra poskytla na tú dobu najlepšie výsledky pri strojovom preklade. [4]
Kľúčový komponent transformerov je tzv. „vrstva pozornosti“ (angl. attention layer). Tieto vrstvy „napovedajú“ celému modelu, na ktoré slová v sekvencii má upriamiť pozornosť a ktoré slová si môže dovoliť „ignorovať“. Táto architektúra nahradila doterajšie prístupy pomocou konvolučných neurónových sietí a paralelizáciou zrýchlila trénovanie. Architektúra sa skladá z enkodéru a dekodéru, pretože táto architektúra bola v prvom rade určená pre strojový preklad. Môžeme ju vidieť na obrázku 2.4. [4]

Enkodér mapuje vstupné sekvencie reprezentácií symbolov (tokenov) $(x_1, \ldots, x_n)$ do sekvencií súvislých reprezentácií $\mathbf{z} = (z_1, \ldots z_n)$. Zo $\mathbf{z}$, dekodér generuje výstupné sekvencie $(y_1, \ldots, y_n)$ symbolov (tokenov) jeden prvok za jednu časovú jednotku. Transformer je teda zložený zo zoskupení modulov pozorností a bodových, plne prepojených neurónových sietí ako pre enkodér, tak pre dekodér. Zobrazené sú na ľavej a pravej polke na obrázku 2.4. [4]



Obr. 2.5: Viac (h) hlavová pozornosť. „Linear“ je jedna z 3 sietí „Scaled dot product attention“ je definovaná v rovnici 1 a „Concat“ je operácia pripojenia výsledných vektorov z h hláv za seba. [4]

2.7.1 Attention – pozornosť

Na začiatku zo všetkých slov na vstupe vytvoríme embeddingy tokenov (pozícia v slovníku po tokenizácii). Na obrázku 2.4 vidíme celú architektúru, kde sa vstupná sekvencia pozičných embeddingov (pre zachovanie informácie o poradí tokenov zo vstupu) dostáva do vrstvy viac-hlavovej pozornosti. Pomocou troch neurónových sietí sa zo vstupných embeddingov (tokenov) vytvoria tri vektory: query Q, key K, a value V. Pomocou siete chceme nájsť podobnosť Q s K_i vektormi ostatných tokenov. Škálovaným skalárnym súčinom vektoru Q s ostatnými K_i dostaneme ohodnotenie (skóre) každého nasledujúceho vstupného embeddingu. Takto model vie, na ktoré nasledujúce tokeny má upriamiť pozornosť (čím vyššie skóre, tým vyššia pozornosť pre daný embedding). Výstupom z jednej hlavy pozornosti je

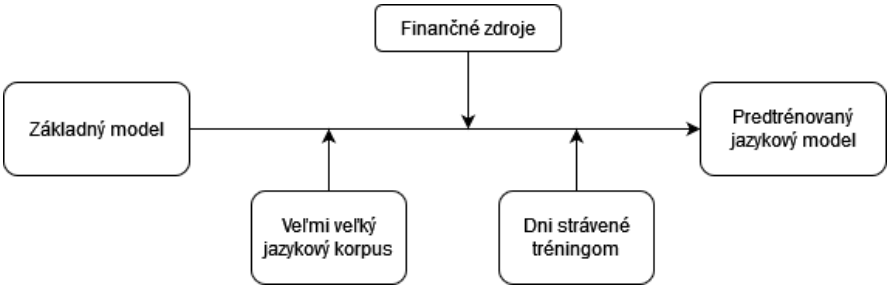
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{1}$$

kde d_k je dimenzia K. Softmax je použitý pre normalizáciu výsledných skóre každého tokenu. [4]

Takýchto hláv je viacero a každý výstup hlavy je pripojený za seba do jedného spoločného vektoru, ktorý je podhodený neurónovej sieti. Jej výstup ide do reziduálneho pripojenia vrstvovej normalizácie. [44] [45].

2.7.2 Transfer learning a fine-tuning v NLP

Aj vďaka uvedeniu transformer architektúry – hlavne BERTa [25] a GPT [46] bolo umožnené veľmi efektívne spracovanie prirodzenej reči. Táto trans-



Obr. 2.6: Predtrénovací postup, zjednodušene. [5]

former architektúra bola určená na tvorbu predtrénovaných modelov, ktoré pomocou prevodného učenia (angl. transfer learningu) bolo možné použiť na riešenie mnohých problémov NLP pomocou vyladenia (angl. fine-tuning) predtrénovaného modelu. Dovtedy boli úlohy NLP veľkou výzvou a vyžadovali časovo a zdrojovo náročné trénovania modelov. [5]

Predtrénovanie prebieha na veľkom množstve dát so zámerom dobre zachytiť štatistické porozumenie jazyka. Model sa inicializuje s náhodnými váhami bez žiadnej predošlej znalosti dát. Zjednodušený postup predtrénovania môžeme vidieť na obrázku 2.6. Dôvody, prečo je použitie predtrénovaných modelov výhodné:

- Predtrénovaný model už bol natrénovaný na dátach, ktoré sú podobné úlohe, ktorú v NLP chceme riešiť. Model teda získanú znalosť môže využiť neskôr pri ladení.
- Predtrénovaný model je možné opätovne použiť a vyladiť na riešenie konkrétnej úlohy (odlišnej od úlohy v predtrénovacej časti) s vyššou šancou na získanie rozumných výsledkov.
- Tréning týchto modelov typicky trvá týždne, stojí veľa peňazí a zdrojov.
- Ladenie je preto menej náročné a vyžaduje skôr hodiny ako týždne na tréning.

Ladenie predtrénovaného modelu má teda nižší časový, finančný, dátový a aj environmentálny dopad. Ak teda nemáme k dispozícii dostatok vhodných dát, je rozumnejšie opätovne použiť hotový model s nadobudnutými znalosťami pre špecifickú úlohu, ktorú riešime. [5] [25] [47] [48]

2.7.3 BERT

Hlavnou úlohou tohoto modelu je možnosť opätovného použitia hotovej siete pre rôzne úlohy NLP. Predtrénovanie tohoto modelu prebieha na neoznačených (nelabelovaných) štrukturovaných dátach – dokumentoch, takže nie je nutné skoro žiadne predspracovanie. Vstupné sekvencie sú tokenizované pomocou

Wordpiece algoritmu [49]. Architektúra BERTa je viacvrstvový obojsmerný transformer enkodér popísaný v sekcii 2.7. Z toho vznikla skratka BERT (angl. Bidirectional Encoder Representations from Transformers). [25]

Masked Language Modeling (MLM)

Alebo maskované modelovanie jazyka, na ktoré sa bežne odkazuje ako na tzv. Cloze úlohu doplnenia chýbajúceho slova do vety [50]. V praxi sa náhodne nahradia tokeny v sekvenciách (vetách) vstupného textu špeciálnym tokenom `[MASK]`. Model sa snaží tieto zamaskované tokeny predikovať správne. V základnom BERT modeli to bolo 15% všetkých tokenov. Aby bola zaručená schopnosť následného ladenia (pretože sa `[MASK]` token pri následnom ladení nevyskytuje), spomínaných 15% náhodne vybraných tokenov z celého korpusu je z (1) 80% nahradených špeciálnym tokenom `[MASK]`, (2) 10% náhodným tokenom a zvyšných (3) 10% nezmeneným. Posledný skrytý vektor tohto tokenu je použitý na jeho predikciu so stratovou funkciou krížovej entropie. [25]

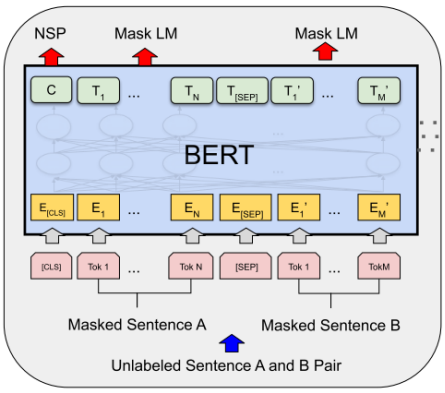
Next Sentence Prediction (NSP)

Alebo predikcia nasledujúcej vety je úloha, ktorá zaručí „porozumenie“ vzťahu medzi dvoma vetami výsledného modelu, ktoré nie je priamo zachytený pomocou modelovania jazyka. Preto je tento predtrénovaný model vhodný použiť na úlohy odpovedania na otázky a dedukciu z textu.
NSP je prevedené na úlohu binárnej klasifikácie, kde pre trénovacie páry viet A a B označia polovicu viet ako nasledujúcu a pre zvyšnú polovicu párov nahradia nasledujúcu vetu náhodne vybranou a označia ako nenasledujúcu. Model sa počas tréningu snaží túto úlohu vyriešiť a tvorcovia tvrdia, že trénovanie na tejto úlohe výrazne pomáha pri vyššie spomínaných úlohách odpovedania na otázky a dedukciu. [25]

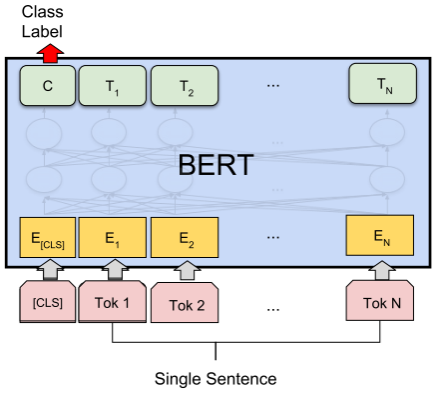
Architektúra a tréning

Na obrázku 2.7a vidíme ukážku architektúry BERTa, kde sú na vstupe páry viet A a B. V každej vete sa môže nachádzať zamaskovaný token. Vety sú oddelené špeciálnym tokenom `[SEP]` pre oddelenie kontextu a na začiatku každého páru je pridaný špeciálny token `[CLF]` a jeho výstup je práve predikcia nasledujúcej vety. Pri ladení sa tento token používa na výstup akejkoľvek klasifikácie, viditeľné na obrázku 2.7b. Tieto tokeny sú prevedené na ich embedding reprezentáciu a následne spracované obojsmernými enkodér časťami transformeru vďaka krížovej pozornosti (modrá časť diagramu).

Ak chceme model použiť, napríklad, na úlohu intent detection – klasifikáciu textu, všetky výstupy skrytej vrstvy Ti „ignorujeme“ a ponecháme len výstup klasifikačnej vrstvy C. [51] [25]



(a) Predtrénovanie BERTa [25]



(b) BERT pre klasifikáciu textu [25]

Obr. 2.7: Naľavo: spôsob predtrénovania celého obojsmerného modelu BERTa. Napravo: BERT model určený ladenie pre úlohu klasifikácie textu.

2.8 Použité predtrénované modely

V tabuľke 2.1 vidíme popis jednotlivých predtrénovaných modelov. Použité dáta sú národný český korpus (Nat) [52], texty Wikipédie (v češtine) [39], Stiahnuté články z českých médií a Czech Colossal Clean Crawled Corpus (C5) [53], czes [54] a W2C [55].

Všetky tieto predtrénované siete sú vďaka `Huggingface.co` hubu [5] verejne dostupné na ich stránkach. Nachádza sa tu veľké množstvo predtrénovaných modelov. Modely trénované na vyššie spomenutých českých dátach sú vypísané viditeľné na tabuľke 2.1.

meno	Arch.	Vocab	Dáta	#param
CZERT-B	BERT	40 tis.	Nat+Wiki+News, 37GB	109 M
CZERT-A	ALBERT	40 tis.	Nat+Wiki+News, 37GB	12 M
FERNET	BERT	100 tis.	C5, 93GB	164 M
Robe-Czech	RoBERTa	52 tis.	Nat+Wiki+Czes+W2C	125 M
Slavic-BERT	BERT	120 tis.	Wiki, 4 jazyky	177 M

Tabuľka 2.1: Meno, použitá architektúra, veľkosť slovníka, dáta použité pri predtrénovaní, počet parametrov modelu (milióny)

2.8.1 CZERT

Model B je základný BERT popísaný v [25], a model A je základný ALBERT – podobný základnému BERT modelu, ale s nižším množstvom parametrov. [56] [57]

2.8.2 FERNET

Tento model má rovnakú architektúru ako BERT [25], avšak použitý je iný tokenizér (SentencePiece), ktorý bol expertne vyladený tak, aby boli zachované všetky české znaky a len rozumné množstvo grafém iných jazykov. Je to z toho dôvodu, že v korpuse sa nachádzajú aj texty iných jazykov. [58]

2.8.3 RobeCzech

Tento predtrénovaný model je založený na architektúre RoBERTa [59], ktorý je rozšírením BERTa so zmenami v predtrénovaní. Model sa trénoval dlhšie, väčšími dávkami nad objemnejšími dátami. Takisto bola vyhodená úloha predikcie ďalšej vety (NSP), tréning prebehol nad dlhšími sekvenciami a dynamicky sa upravovalo maskovanie slov pri úlohe MLM. [60]

2.8.4 SlavicBERT

Predtrénovaný model je založený na architektúre BERTa [25], avšak s pridanou vrstvou Conditional Random Fields (CRF) [61]. Model je viacjazyčný, dokáže spracovať 4 rôzne jazyky (ruština, buhlarčina, čeština a poľština). Radí sa teda medzi viacjazyčné modely. [62]

Kapitola 3

Použité metriky a podobnosti

V tejto kapitole popíšme kosínovú podobnosť (vzdialenosť) dvoch vektorov, levenshteinovu vzdialenosť dvoch reťazcov a sémantickú granularitu zoznamu viet. Neskôr bežne používané metriky ako je presnosť (accuracy), balancovanú presnosť (balanced accuracy) a F1 skóre. Tieto metriky sú vhodné na tzv. multiclass problém klasifikácie, pretože ich hodnoty dobre zachytávajú a popisujú kvalitu merania. [33]

3.1 Kosínová podobnosť

je definovaná ako normalizovaný skalárny súčin:

$$K(X,Y) = \frac{< X, Y >}{(||X|| * ||Y||)}$$

kde X a Y sú rozdielne zoznamy vzoriek. Táto definícia kosínovej podobnosti $K(.,.)$ je z knižnice `scikit-learn` [33] a má na vstupe dva zoznamy vektorov.

Predefinujme pre jednoduchosť túto funkciu po zložkách:

$$K(x,y) = \frac{\langle x, y \rangle}{||x|| \cdot ||y||}$$

kde x a y sú nejaké vektory reálnych čísel rovnakej veľkosti, $\langle x, y \rangle$ je ich skalárny súčin a $||x||$ je L2 norma vektoru x.

Často budeme mať k dispozícií zoznam viet X, kde x_i je prvkom zoznamu (jedna veta).

Ak by sme chceli vypočítať pre zoznam X maticu podobností A (každá veta s každou), použijeme definíciu po zložkách pomocou vyššie definovanej kosínovej podobnosti $K(.,.)$:

$$A_{i,j} = K(f(x_i), f(x_j))$$

- kde $x_i, x_j \in X$, $i, j \in 1..|X|$,
- f je embedovacia funkcia $f(x) = (a_1, a_2, a_3, ...a_n)$, $a_i \in R$ a x je nejaká veta v prirodzenom jazyku.

Kedykoľvek v texte použijeme výraz $K(\text{"}veta1\text{"}, \text{"}veta2\text{"})$ alebo $K(x, y)$, myslíme tým výraz $K(f(\text{"}veta1\text{"}), f(\text{"}veta2\text{"}))$ alebo $K(f(x), f(y))$ pre zjednodušenie.

3.2 Levenshteinova vzdialenosť

Levenshteinova vzdialenosť je definovaná ako najmenší počet jedno-znakových úprav (vloženie, vymazanie, nahradenie) potrebných pre zmenu slova A na slovo B. Uveďme príklad: Slovo „pas“ „pes“ má vzdialenosť 1, pretože pre získanie identických slov je nutné nahradiť druhý znak v slove „pes“ písmenom „a“. [63]

3.3 Sémantická granularita

Rozdiel medzi pojmami „podobnosť“ a „granularita“ je veľký. Kým pri podobnosti skúmame akúsi blízkosť v priestore (myšlienkach, pojmoch pri prirodzenej reči) a teda by sme mohli podobnosť definovať skôr ako vzdialenosť, tak pri granularite skúmame akúsi príslušnosť k danej doméne a zámeru. [64]

Majme k dispozícii zoznam viet (popisy nejakého zámeru). O sémantickej granularite tohto zoznamu povieme, že je:

- vysoká, ak všetky vety z tohto zoznamu sú si významovo podobné,
- nízka, ak sa v tomto zozname nachádzajú zoskupenia viet, ktoré sú v rámci zoskupenia významovo podobné ale nie až tak podobné krížovo medzi zoskupeniami (v rámci jednoho zoznamu).

Pre lepšiu ilustráciu uveďme príklad:

- vety „Rád by som si od teba požičal auto.“ a „Požičiaš mi auto?“ majú vysokú sémantickú granularitu,
- „Požičiaš mi auto?“ a „Koľko peňazí mi viete požičať na auto“ majú nízku sémantickú granularitu, keďže zámer pýtajúceho je úplne odlišný.

3.4 Použité metriky

Zaoberáme sa úlohou klasifikácie do c tried (multiclass, single label). Sledujme teda metriky hlavne pre multiclass klasifikáciu.

3.4.1 Konfúzna matica

Zadefinujme **konfúznu maticu**, kde sledujeme počty predikovaných hodnôt $\hat{Y}_i$ oproti pravdivým hodnotám Y_i.

	$Y = 1$	$Y = 0$	$\sum$
$\hat{Y} = 1$	TP	FP	$\hat{N}_+ = \mathrm{TP} + \mathrm{FP}$
$\hat{Y} = 0$	FN	TN	$\hat{N}_- = \mathrm{FN} + \mathrm{TN}$
$\sum$	$N_+ = \mathrm{TP} + \mathrm{FN}$	$N_- = \mathrm{FP} + \mathrm{TN}$	$N = \mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}$

Z konfúznej matice napočítajme:

- True positive rate (TPR) tiež známa ako **senzitivita** alebo **recall** alebo **hit rate**.
- False positive rate (FPR) tiež známa ako **false alarm rate** alebo **chyba I. rádu**.
- False negative rate (FNR) tiež známa ako **miss rate** alebo **chyba II. rádu**.
- True negative rate (TNR) tiež známa ako **špecificita** alebo **selektivita**.

Predchádzajúce definície sú pre prehľadnosť zobrazené v tejto tabuľke:

	$Y = 1$	$Y = 0$
$\hat{Y} = 1$	$\frac{\mathrm{TP}}{N_+} = \mathrm{TPR}$	$\frac{\mathrm{FP}}{N_-} = \mathrm{FPR}$
$\hat{Y} = 0$	$\frac{\mathrm{FN}}{N_+} = \mathrm{FNR}$	$\frac{\mathrm{TN}}{N_-} = \mathrm{TNR}$

Všetky tieto miery sú odhadmi (diskrétnej) podmienenej pravdepodobnosti $\mathrm{P}(\hat{Y} = \hat{y} \mid Y = y)$. [65]

3.4.2 Presnosť

Zadefinujme si presnosť (accuracy), odhad $\mathrm{P}(\hat{Y} = Y)$:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{N} = \frac{1}{N} \sum_{i=0}^{N-1} 1(\hat{Y}_i = Y_i),$$

kde 1(.) je funkcia indikátoru výskytu. [33]

3.4.3 Balancovaná presnosť

Presnosť – accuracy – definovaná vyššie nie je vhodná pre nebalancované dáta, kde rozdelenie počtu vzoriek pre triedu nie je rovnomerné. Môžeme to obísť definovaním balancovanej presnosti:

$$\mathrm{ACC}_{bal} = \frac{1}{|C|} \sum_{i}^{|C|} \left(\frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FN_i}} \right),$$

kde C je množina tried a teda TP_i a FN_i sú napočítané pre každú triedu zvlášť. Je to teda priemer recallov každej triedy. [33]

3.4.4 TOP_k Presnosť

Pri multiclass klasifikačnom probléme pre c tried model predikuje pravdepodobnosti

$$\hat{p}_i(x) = \hat{\mathrm{P}}\left(Y = i | X = x\right) \forall i = 1, \ldots, c.$$

Pri TOP_k presnosti (accuracy) sledujeme to či sa klasifikátor, ktorý predikuje pravdepodobnostné ohodnotenie všetkých tried $\hat{\boldsymbol{p}}$, mal na jednej z najvyšších k pravdepodobností pravdivú triedu y_i:

$$\mathrm{ACC}_k = \frac{1}{N} \sum_{i}^{N} \sum_{j=1}^{k} 1(\hat{p}_{i,j} = y_i).$$

Táto metrika je vhodná pre klasifikátory, ktorých výstupom nie je iba samotný predikovaný label, ale pravdepodobnostná distribúcia všetkých labelov. [33]

3.4.5 F1 skóre

Najprv definujme precíznosť (precision), alebo *positive predictive value*

$$PPV = \frac{TP}{\hat{N}_+}.$$

F1 skóre je definované ako harmonický priemer PPV (precision) a TNR (recall) pre binárny klasifikátor $\mathrm{P}(\hat{Y} = 1 \mid Y = 1)$

$$F_1 = \frac{2}{1/\mathrm{PPV} + 1/\mathrm{TPR}} = 2\frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}}.$$

Táto metrika je užitočná pri nebalancovaných dátach.

Ak chceme rozšíriť klasifikáciu na multiclass s množinou tried C, a zachovať citlivosť na nebalancované dáta, definujme vážené F_1 skóre:

$$\text{averaged } F_1 = \frac{1}{\sum_{c \in C} |\hat{y}_c|} \sum_{c \in C} |\hat{y}_c| \, F_1(y_c, \hat{y}_c).$$

Poznamenajme, že hodnota tohto priemerovaného skóre nemusí ležať medzi precision a recall. [33]

Časť II

Praktická časť

Kapitola 4

Pilotná aplikácia Text2Bank

Pre tvorbu detektoru kontextu sme najskôr museli zozbierať vhodné dáta. Pod taktovkou firmy Profinit sme vytvorili webovú aplikáciu „Text2Bank“, ktorej úlohou bolo zozbierať rôzne vzorky 10 vybraných tried (intentov). Jednoduchý klasifikátor, ktorý jednotlivé vstupy od užívateľov zaraďoval do vybraných tried bol postavený na báze naivného bayesa a kľúčových slov.

4.1 Princíp fungovania

Ku každému kľúčovému slovu[4] bolo v matici naivného bayesa priradený bayesov faktor rovný 1, ktorý slúžil ako indikácia potenciálneho výskytu v intente. Trénovanie modelu teda nebolo nutné, keďže samotná matica vznikla expertne – práve pomocou kľúčových slov.

Matica naivného bayesa teda mala k riadkov a c stĺpcov. Na riadku i je vektor, kde každý jeho prvok je bayesovským faktorom – výskyt daného kľúčového slova v danej triede. Samotná detekcia intentu prebiehala nasledovne:

1. rozdeľ pomocou knižnice `nltk` [66] celý vstup na slová – časti vety,
2. jednotlivé časti vety predspracuj nasledovne: ponechaj len slová; zahoď stop slová, čísla a znakovo zmiešané slová (čísla a písmená),
3. pomocou levenshteinovej vzdialenosti nájdi ku každému slovu v danom vstupe najbližšie kľúčové slova (ich počet je parametrizovateľný)
4. pre dané slovo zo vstupu spočítaj pravdepodobnosť príslušnosti intentu a urob priemer pravdepodobností všetkých slov,
5. vypíš tri intenty s najvyššími hodnotami pravdepodobností.

[4] ktoré bolo upravené do koreňovej podoby jednotlivých slov napr. slovo „půjčit“ tu je v tvaroch „půjč“, „půjči“. Vďaka tomu sme mohli pomocou levenshteinovej vzdialenosti hľadať približné zhody

Obr. 4.1: Ukážka výsledku detekcie intentu v pilotnej aplikácii Text2Bank.

4.2 Popis aplikácie

Zber prebiehal pomocou webovej aplikácie, ktorú môžeme vidieť na obrázku 4.1, kde vybraní užívatelia zadávali vety intentov a tento „klasifikátor“ ponúkal **tri** triedy s najvyššou hodnotou pravdepodobnosti. Užívateľ po vyhodnotení zvolil triedu, ktorú intentom myslel (každý výsledok bolo tlačidlo) a ak sa tam nenachádzala, zvolil ju spomedzi všetkých tried z dropdown menu a/alebo pripísal poznámku k vyhodnoteniu a/alebo samotnému intentu. Touto metódou sa vytvorila dátová sada text2bank – každé kliknutie na výsledok alebo trieda vybraná z dropdown menu spolu so spätnou väzbou bola zaznamenaná do databáze.

Na obrázku 4.1 vidíme zadaný vstup – zámer – a dole prvé tri najlepšie

výsledky klasifikácie naivného bayesa. Každý výsledok má priradené pravdepodobnostné ohodnotenie. Pod výsledkami sa nachádza tzv. „dropdown“ menu pre prípadný výber myslenej triedy, ak sa nenachádza vo výsledkoch a pole pre spätnú väzbu od užívateľa.

Pomocou regulárnych výrazov sa z textu extrahujú **3 typy entít**: suma (ak sa vyskytuje v texte číslo a ak sa tu vyskytuje názov meny, tak je tiež extrahovaná), číslo účtu (ak je v správnom tvare s lomkou) a IBAN (pomocou knižnice schwifty [67])

4.3 Detailnejší rozbor fungovania aplikácie

Detekcia intentu v tejto aplikácii zvládala jednoduchšie vstupy, ako napríklad vetu „Toš, jak velký peníze si můžu půjčit?“, ktorú klasifikovala správne do triedy „Chci půjčit/kolik mi půjčíte“, pretože bolo prítomné slovo „půjčíte“, ktoré je dva znaky vzdialené od slova „půjčit“. Oproti tomu so vstupom „kdy můžu přijít vykradnout banku? :)“ si už, bohužiaľ, neporadila. Neboli vytvorené kľúčové slová a ani samotná trieda, ktorá by popisovala tento vstup. Klasifikátor si s tým však poradil po svojom. Keďže sa tu nachádzalo slovo „banku“, ktoré je vzdialené len 1 znak od kľúčového slova „banky“ tak bola táto otázka klasifikovaná do triedy „Sjednat schůzku na pobočce“[5].

Kvôli jednoduchosti klasifikátora a absencie pozornosti kontextu správy napríklad vstup „kolik mám naspořeno?“ aplikácia zaradila do triedy „Sjednat Hypotéku“, pretože sa tu nachádzalo slovo „mám“, ktoré je vzdialené dva znaky od kľúčového slova „dům“ príslušné tejto triede. Je teda zrejmé, že tento model má svoje chyby a problémy a nie je dostatočne robustný pre použitie.

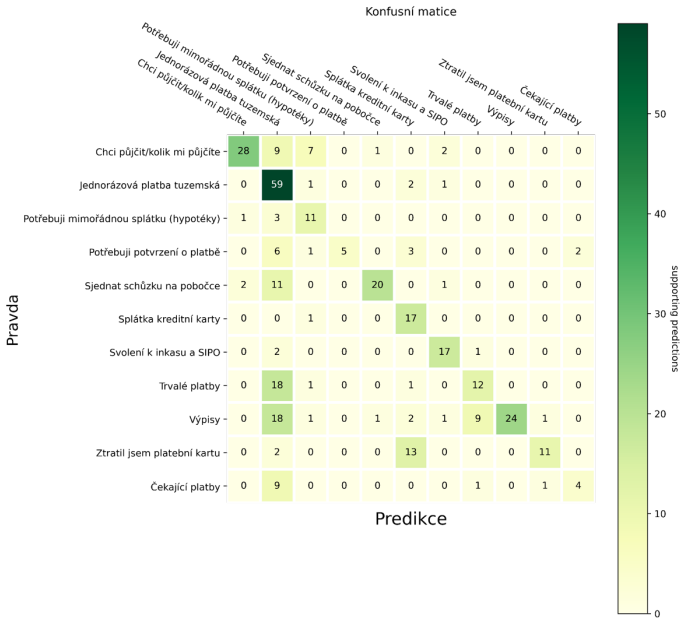
4.4 Predspracovanie zozbieraných dát

Po zozbieraní 500+ vzoriek bolo prevedené čistenie dát, expertné priradenie správnych tried k vstupom, ak sa užívateľ zmýlil. Taktiež boli vyhodené niektoré nehodiace sa vstupy ako napríklad „1=1; SELECT * FROM dual;“ alebo „Vtip dne“, prípadne „zmena pin“, čo nepatrilo ani do jednej z tried. Vyhodili sme duplicity, ktoré vznikli po doplnení diakritiky.

Po vyčistení dát ostalo približne 400 vzoriek od 20-tich rôznych užívateľov. Kľúčové slová boli expertne obohatené doplnením a úpravou niektorých z nich. Organicky vznikli nové triedy, konkrétne trieda „limity“ a „Sjednat hypotéku“.

Bohužiaľ, predspracovanie dát bola do veľkej miery manuálna práca, keďže sme museli expertne preverovať sémantiku a správnosť daných vstupov. Prebehla manuálna gramatická a diakritická korekcia vstupov pomocou voľne

[5]čo by sme mohli v prenesenom význame určiť ako správny výsledok klasifikácie ale pre účely správnosti a konzistentnosti sme tento vstup z dátovej sady radšej vyhodili.



Obr. 4.2: Konfúzna matica predikcií klasifikácie užívateľských vstupov počas zberu dát. Na osi x sú počty predikovaných klasifikácií do danej triedy a na osi y sú pravdivé hodnoty tried klasifikácie, určené užívateľmi a manuálne skontrolované.

dostupných korektorov, ako napríklad LINDAT korektor [68], aby sme mali dáta vyčistené a vhodné na neskorší tréning.

4.5 Výsledky

Klasifikácia pomocou pilotného modelu mala **presnosť 0,64 a F1 skóre 0,63** na všetkých zozbieraných vstupoch. Konfúznu maticu presných výsledkov môžeme vidieť na 4.2. Novovzniknuté triedy, „limity“ a „Sjednat hypotéku“ do analýzy nezahŕňame.

V prílohe na obrázku A.1 môžeme vidieť konfúznu maticu klasifikácie pomocou druhej, vylepšenej verzie, kde sme pridali nové kľúčové slová, ktoré sa objavili vo vzorkách. **Presnosť** modelu sa zvýšila na **0,82 a F1 skóre na 0,83**, čo sú uspokojivé výsledky, ak berieme do úvahy jednoduchosť úlohy a algoritmu.

Výhodou tohoto typu modelu je jeho rýchlosť, jasná interpretácia výsledku. Opravou kľúčových slov je možné rýchle zvýšenie kvality modelu. Bohužiaľ, oprava sa nedá vykonať automatizovane, iba manuálne.

Problém nastáva, ak narastie počet tried a ak sa veľa kľúčových slov v niektorých triedach prekrýva. V tom momente je model nepoužiteľný. Takisto môžeme považovať model za „preučený“, pretože pri abstraktnejších a kom-

plexnejších vstupoch prestáva použitie kľúčových slov dávať zmysel a model zle generalizuje. Dobrou ukážkou sú vety: „Chci bydlet ve vlastním, ale nemám na to peníze.“, kde je význam zámeru skrytý a nenachádzajú sa tu žiadne uchopiteľné kľúčové slová, alebo: „Potřeboval bych se sejít ohledně hypotéky“, kde sa zlučujú zámery hypotéky a schôdzky do niečoho abstraktnejšieho a ťažko predikovateľného pomocou tohoto typu modelu, napriek tomu že zámer bol jednoznačne „Sjednat schůzku na pobočce“.

Kapitola 5

Finálna verzia intent detection

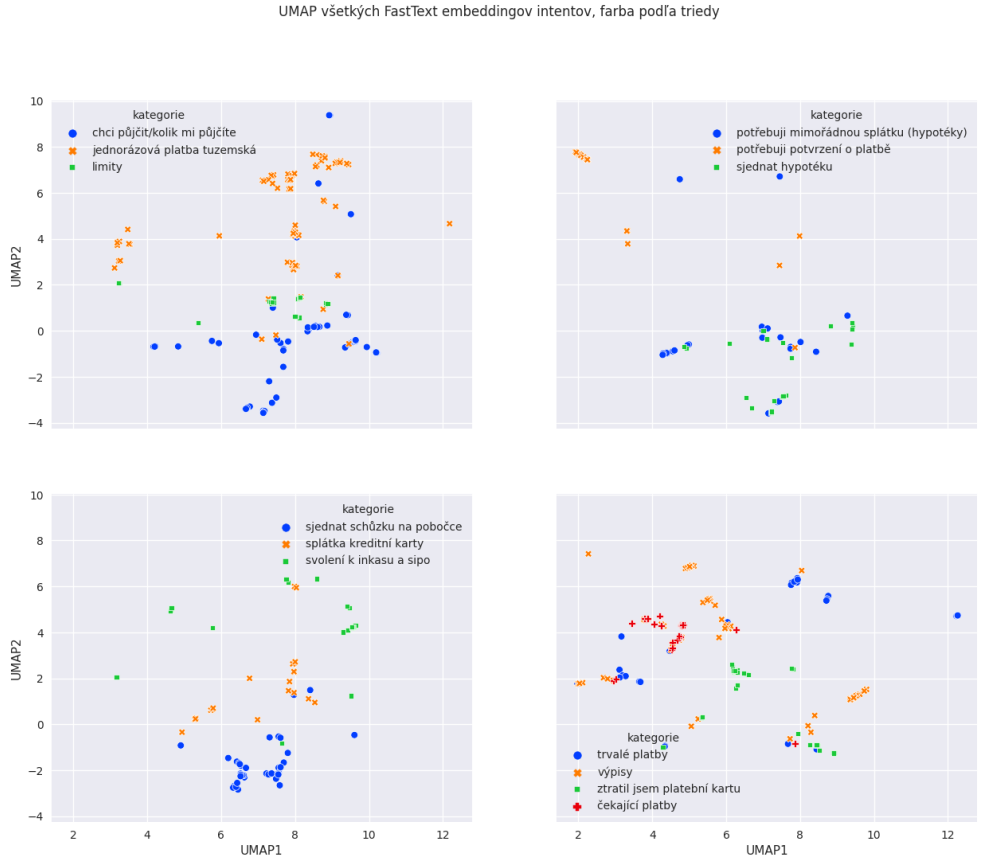
V tejto kapitole popíšeme použité dáta spolu a analyzujeme ich kvalitu. Neskôr popíšeme intent detection na oboch sadách zvlášť ako aj na spojených dátach. Popíšeme finálnu verziu intent detektoru a zozbierané výsledky reálneho používania.

5.1 Dátová sada TEXT2BANK13

Dátová sada TEXT2BANK13 obsahuje 491 unikátnych vzoriek vstupov pre 13 rôznych tried – intentov. Vybrané triedy, ktoré sme zbierali ako aj počet zozbieraných vzoriek môžeme vidieť na tabuľke 5.1. Priemerný počet slov na vetu po jednoduchej tokenizácii je 5,15.

trieda	počet intentov
jednorázová platba tuzemská	88
výpisy	69
chci půjčit/kolik mi půjčíte	59
sjednat schůzku na pobočce	41
trvalé platby	37
ztratil jsem platební kartu	33
potřebuji mimořádnou splátku (hypotéky)	25
svolení k inkasu a sipo	24
čekající platby	24
sjednat hypotéku	23
splátka kreditní karty	23
limity	22
potřebuji potvrzení o platbě	22

Tabuľka 5.1: Počty vzoriek jednotlivých intentov pre každú triedu dátovej sady TEXT2BANK13



Obr. 5.1: Rozdeľme triedy do 4 skupín a zobrazme príslušne ofarbené FastText embeddingy intentov v 2D priestore pomocou UMAPu.

Popíšme štruktúru vzoriek pomocou vhodných zobrazení.

5.1.1 Sémantická granularita dátovej sady

Vytvorili sme embeddingy všetkých vzorkov z TEXT2BANK13 pomocou FastTextu a redukovali dimenziu embeddingov na 2 pomocou UMAP. Vizualizovali sme všetky vzorky na 2D grafoch a odlíšili farbami[6], ako vidíme na obrázku 5.1. Chceli sme ukázať či sa vzorky rôznych tried od seba líšia v priestore embeddingov a zistiť ako veľmi sú dáta zašumené.

Na obrázku 5.1 vidíme, že napriek niekoľkým outlierom vznikajú v UMAPe zhluky vzoriek rovnakého intentu. Výsledok pozorovania je síce ťažšie interpretovateľný, avšak získavame približnú predstavu o separabilnosti intentov v tejto dátovej sade. Dáta sa môžu javiť ako zašumené, ale spôsobené

[6]Ak by sme dali všetky triedy do jednej projekcie, uvidíme skôr šum a sem-tam zhluk intentov rovnakej triedy. Nebolo by to vhodné na vizualizáciu, preto sme rozdelili všetky triedy do 4 kúskov (v poradí ako sú v sade)

je to podľa všetkého nedokonalosťou UMAPu a FastTextu a tým, že sa v dátach nachádzajú abstraktnejšie vzorky, ktoré nie sú slovami príliš podobné ostatným z rovnakého intentu. Zhluky však ukazujú, že vzorky intentov sú dostatočne odlišné medzi intentami. Podobnú vizualizáciu avšak pre každú triedu zvlášť môžeme vidieť v prílohe na obrázku A.3.

Po hlbšej analýze dát, ktorú môžeme vidieť v prílohe, kapitola A vysvitlo, že trieda limity má, zdá sa, príliš nízku sémantickú granularitu a bolo nutné ju zo sady vyhodiť. Takisto vybrané vety z tejto triedy sa neukázali dostatočne podobné názvu triedy, ktorý je, zdá sa, príliš krátky a má príliš široký význam.

5.1.2 Detekcia intentov sady

Na tejto skromnej dátovej sade sme vykonali klasifikáciu vzorkov dát – detekciu intentov. Výsledky môžeme vidieť na tabuľke 5.2. Použili sme (v poradí):

- FastText pre slovné embeddingy (w.e.), kde veta bola rozbitá na slová a klasifikovali sme embeddingy jednotlivých slov pomocou Gaussian Naive Bayes (gNB). Výsledná predikcia vzorky bola určená ako priemer z pravdepodobnostných rozdelení, ktoré gNB dal.
- FastText pre vetné embeddingy (s.e.), po vyhodení nehodiacich sa slov, napr. čísel alebo špec. znakov a klasifikátor bol znova gNB.
- LASER, FERNET a FastText vetné embeddingy (s.e.) a predikcia príslušnosti vzorky prebiehala na základe najvyššej získanej hodnoty kosínovej podobnosti s embeddingom akéhokoľvek názvu triedy.
- Prvú a druhú verziu intent detektoru Text2Bank (NB a levenshteinova vzdialenosť) len na základe kľúčových slov.

prístup	valid TEXT2BANK13		
	F1	t3 ACC	ACC
FastText w.e. & gNB	40,46	66,30	39,10
FastText s.e. & gNB	**70,26**	**82,60**	**70,60**
	celá TEXT2BANK13		
LASER s.e.	40,46	61,02	49,58
FERNET s.e.	28,02	46,73	28,58
FastText s.e.	32,26	54,28	39,16
Text2Bank I.	63,00	72,00	64,00
Text2Bank II.	**79,36**	**88,16**	**77,10**

Tabuľka 5.2: Výsledky detekcie zámerov na dátach TEXT2BANK13.

Výsledky z tabuľky 5.2 ukazujú, že klasifikácia pomocou kľúčových slov je oproti vetným embeddingom lepšia (výsledky klasifikácie prebehli na celej

sade, pretože tu nebola nutnosť tréningu – používame predtrénované modely). Zároveň tu však môžeme vidieť, že FastText vetné embeddingy dali celkom rozumné výsledky na validačnej vzorke tejto sady. Veľké modely neurónových sietí nebolo možné použiť, pretože trénovacích dát je málo a teda riešime few-shot learning situáciu.

5.2 Dátová sada BANKING77

Casanueva et. al. [17] popísal a prvý použil novú dátovú sadu orientovanú na doménu bankovníctva. Dátová sada BANKING77 (v skratke B77) obsahuje 77 tried (zámerov) a 13 083 vzoriek s dostatočne vysokou sémantickou granularitou. Dáta sa nachádzajú verejne dostupné na stránke github.com[7]. Tieto dáta boli zozbierané zo zákazníckej podpory konkrétnej zahraničnej banky. Oproti datasetom HWU64 [26] a CLINC150 [27] obsahujú vzorky viac slov (priemerne 12) a zároveň sa jednotlivé intenty od seba nelíšia slovami ale práve kontextom. [17]

Tým, že táto dátová sada bola vytvorená v anglickom prostredí a tak aj jazyk vzoriek je angličtina. Tento problém sme vyriešili za použitia Google Cloud Platform [69], konkrétne „Translation API“ [70]. Strojovým prekladom sme boli schopný získať vcelku kvalitné a hlavne reálne dáta od užívateľov.

5.2.1 Kvalita prekladu

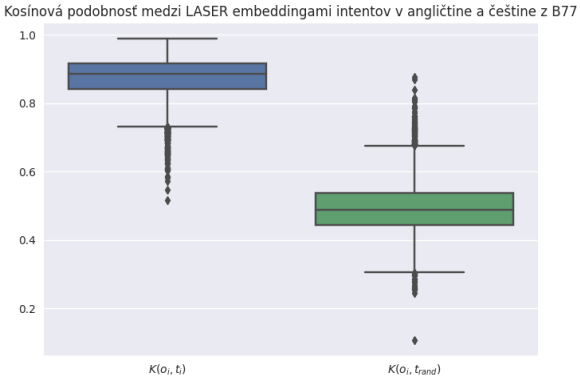
Nástroj LASER, dokáže tvoriť viacjazyčné embeddingy, stačí teda zvoliť jazyk a podľa slov tvorcov by reprezentácia rovnakých viet v rôznych jazykoch mala byť veľmi podobná. Boli sme teda schopní overiť kvalitu prekladu tak, že sme porovnali:

- kosínové podobnosti $k_i = K(o_i, t_i), i \in 0..\#intent$, kde o_i je originálny intent v angličtine, t_i je originálna veta preložená do češtiny,
- kosínové podobnosti $k_i = K(o_i, t_{rand}), i \in 0..\#intent$, kde o_i je originálny intent v angličtine, t_{rand} je náhodne vybraný intent z datasetu preložený do češtiny.

Hodnoty podobností môžeme vidieť na obrázku 5.2 pomocou mediánu a kvartilov vo forme boxplotu.

Preklad sa teda javí byť v poriadku, pretože medián z obr. 5.2 vľavo okolo 0,9 hovorí o vysokej podobnosti originálnych viet v angličtine a preložených viet. Oproti tomu náhodné preklady majú relatívne k boxplotu vpravo nízky medián podobnosti, čo je podľa očakávaní. Odľahlé body $K(o_i, t_{rand})$, ktoré

[7] https://github.com/PolyAI-LDN/task-specific-datasets/tree/master/banking_data



Obr. 5.2: Hodnoty kosínovej podobnosti LASER embeddingov anglických a českých viet zo sady BANKING77. Naľavo vidíme hodnoty podobností medzi originálnym anglickým a príslušným preloženým textom do češtiny, napravo hodnoty originálnych textov k náhodnému preloženému textu zo sady

majú vysokú mieru podobnosti, patria podľa všetkého do tej istej triedy. Podobná analýza prebehla na prekladoch názvov tried – intentov. Môžeme ju nájsť v prílohe na obr. B.1.

5.2.2 Sémantická granularita

V prílohe, kapitola B a v priloženej analýze na CD sme skúmali sémantickú granularitu vzoriek dát zo sady BANKING77, aby sme odhalili existenciu intentov, ktorých vzorky sa na seba potenciálne sémanticky podobajú a či sú všetky intenty od seba dostatočne sémanticky odlišné. Najprv sme pomocou KMeans algoritmu vytvorili niekoľko zhlukov z názvov tried a následne sme zobrazili LASER embeddingy vzoriek dát jednotlivých intentov pomocou UMAPu a zafarbili podľa príslušného intentu. Tieto zobrazenia môžeme vidieť v sekcii B.2.

Z tejto analýzy sme zistili, že granularita sa zdá byť až na výnimky rozumne vysoká. Ak sa vyskytli problémy, boli podľa všetkého spôsobené zlým prekladom, výnimočne zlým zaradením vzorky k príslušnému intentu alebo jednoducho nedokonalosťou LASERu a/alebo UMAPu. Niektoré triedy sú si významovo veľmi podobné, a preto sa na UMAP zobrazeniach javili veľmi blízko seba, prípadne sa niekedy prekrývali. Mnohé prekryvy však boli ignorované, pretože sme neskôr tieto triedy intentov vyhodili. Boli pre naše účely aplikácie v doméne českého bankovníctva nepotrebné.

5.2.3 Intent detection

Na tabuľke 5.3 vidíme výsledky klasifikácie rôznych predtrénovaných modelov, ktoré sme si vďaka väčšiemu množstvu trénovacích dát mohli dovoliť použiť.

prístup	valid BANKING77 CZ		
	F1	top3 ACC	ACC bal
CZERT-A	89,87	96,46	88,12
CZERT-B	**92,25**	**97,59**	**92,24**
RobeCzech	83,00	95,00	84,15
Slavic-BERT	90,88	97,31	90,88
FERNET-C5	92,09	97,21	92,05
	celá BANKING77 CZ		
LASER s.e.	**36,20**	**54,49**	**34,37**
FERNET s.e.	7,25	18,73	8,86
FastText s.e.	19,91	35,56	20,19
	celá BANKING77 EN		
LASER s.e.	34,35	53,95	33,45

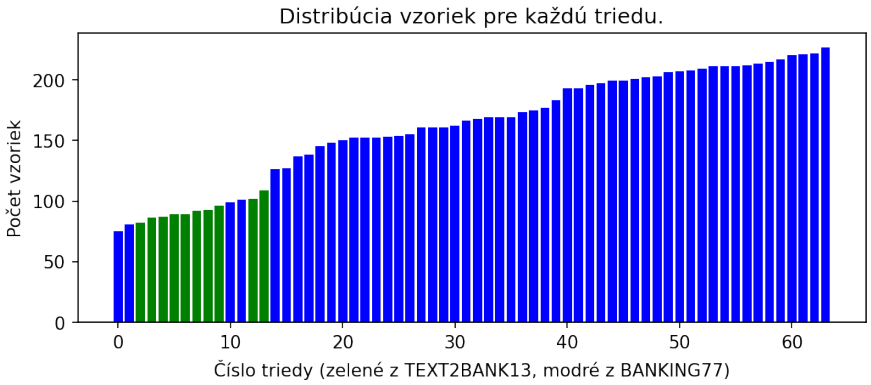
Tabuľka 5.3: Výsledky klasifikácie strojovo preloženej (testovacej) dátovej sady BANKING77 (ak boli použité vetné embeddingy, výsledky sú z celej sady). Zľava: názov predtrénovaného modelu (s.e. znamená klasifikácia pomocou vetných embeddingov podobných s názvom intentu), F1 skóre, TOP_3 presnosť, nebalancovaná presnosť (validačná množina bola vyvážená).

Namerané hodnoty F1 skóre, TOP_3 presnosti a balancovanej presnosti. Prvých 5 modelov v tabuľke sme zostavili pomocou techniky fine-tuningu. Zmrazili sme predtrénované české BERT modely a za túto zmrazenú architektúru nasadili klasifikačnú vrstvu. Celý model sme trénovali, dokým sa nezačali pretínať trénovacie a validačné loss krivky. Na tomto mieste sme model exportovali a výsledky meraní na validačnej sade dali do tabuľky. Veľkosť dávky (batch) bola 16 vzoriek. Testovacia sada nebola použitá, overovali sme týmto iba kvalitu modelov medzi sebou.

Stredná časť z tabuľky 5.3 sú výsledky klasifikácie pomocou podobnosti embeddingov vzoriek s názvami tried. Táto technika okrem LASERu nevykazovala väčšie úspechy. Posledná časť ukazuje výsledky klasifikácie vetných modelov nad vzorkami v anglickom jazyku.

5.3 Dátová sada TEXT2BANK64

Keďže sme chceli, aby náš systém intent detection bol rozmanitejší ako v počte intentov, tak v počte vzoriek, previedli sme analýzu zlučiteľnosti sád TEXT2BANK13 a BANKING77 v prílohe, kapitola C. Najprv sme však z TEXT2BANK13 museli vyhodiť intent s nízkou sémantickou granularitou (trieda „limity“). Konkrétne intenty zaoberajúce sa navýšením kreditu (top-up) boli z BANKING77 vyhodené, pretože sa tieto zámery v prostredí klasických českých bánk nevyskytujú. V kapitole C.1 sme (nielen) pomocou LASERu a kosínovej podobnosti našli kandidátske intenty pre zlúčenie. V ďalších



Obr. 5.3: Zobrazenie počtu vzoriek TEXT2BANK64 pre každý intent zvlášť.

sekciách tejto kapitoly sme analyzovali možnosti zlúčenia. Táto analýza ukázala, že zlúčiť je možné iba intenty „ztratil jsem platební kartu“ z TEXT2BANK13 s „ztracená nebo odcizená karta“ z BANKING77 a potom „čekající platby“ z TEXT2BANK13 s „čekající převod“ z BANKING77.

Dospeli sme k záveru, že sadu TEXT2BANK13 bolo treba rozšíriť o niekoľko nových vzoriek, pretože väčšie modely s mnohými parametrami majú problém s malým počtom trénovacích dát – napriek tomu, že sme používali modely predtrénované. Toto dopĺňanie prebehlo expertne (manuálne):

- nahradením niektorých slov ich synonymami,
- pridaním slov, ktoré nemenili význam vety,
- pridaním zdvorilostných fráz na začiatku a na konci vety,
- rozdelením alebo skladaním viet do súvetí,
- miernou zmenou slovosledu a zmenou poradia vedľajších viet.

Všetky tieto zmeny boli vykonané tak, aby príslušná novovzniknutá veta dávala stále rovnaký zmysel a významovo sedela k danému intentu, ku ktorému patrila a zároveň bola unikátna. Dosiahli sme tým rozumný počet vzoriek – najmenej početná trieda z rozšírenej TEXT2BANK13 má stále viac vzoriek ako najmenej početná trieda z BANKING77.

Dátová sada TEXT2BANK64 bola zostavená pomocou vyššie spomenutých postupov. Táto sada má 64 intentov (tried klasifikácie) a 10 357 vzoriek. Všetky tieto vzorky pochádzajú z jedinej domény – bankovníctvo. Distribúciu počtu vzoriek pre každú triedu môžeme vidieť na obrázku 5.3. Zelené stĺpčeky sú triedy z TEXT2BANK13 a modré z BANKING77.

5.3.1 Intent detection

Z analýz a detekcií intentov datasetov TEXT2BANK13 a BANKING77 sme usúdili, že trénovať jednoduchšie modely nie je pre takýto komplexný problém dostačujúce. Preto sme zvolili iba predtrénované transformer modely na českých dátach. Výsledky na validačnej množine zvolenej ako 10% frakcia celej sady[8] TEXT2BANK64 rôznych transformer modelov môžeme vidieť v tabuľke 5.4.

model	acc	bal acc	TOP_3 acc	f1
CZERT-B	90,04	90,09	90,08	**97,88**
FERNET	89,06	89,19	88,95	97,30
ROBECZECH	**92,46**	**92,57**	**92,83**	97,30
Slavic-BERT	89,28	89,38	89,64	96,14

Tabuľka 5.4: Výsledky predikcie validačných vzoriek. Zľava: vyladený predtrénovaný model, presnosť (nebalancovaná), presnosť (balancovaná), TOP_3 presnosť, F1 skóre

Pre lepší pohľad na schopnosť klasifikácie zlúčenej sady zobrazujeme na tabuľke 5.5 rovnaké validačné metriky, avšak len z validačných vzoriek oboch logických častí.

sada	valid BANKING77			valid TEXT2BANK13		
model	acc	acc_bal	f1	acc	acc_bal	f1
CZERT	89,89	89,77	89,94	91,06	90,83	92,89
FERNET	89,19	88,98	89,16	88,89	88,58	88,10
ROBECZECH	**92,27**	**92,35**	**92,24**	**95,06**	**94,91**	**95,64**
Slavic-BERT	89,19	89,32	89,23	92,59	92,41	92,54

Tabuľka 5.5: Výsledky predikcie validačnej vzorky, zvlášť pre vzorky tried patriace pod BANKING77 (B77) a TEXT2BANK13 (T2B). Zľava: Vyladený predtrénovaný model, presnosť (nebalancovaná), presnosť (balancovaná), F1 skóre

Najlepšie výsledky na validačnej množine nám dal predtrénovaný model RobeCzech [60], aj keď má hodnotu F1 skóre o niečo nižšiu ako model CZERT-B. Hodnoty sledovaných metrík na validačnej množine modelu RobeCzech sú pomerne vysoké, čo sme chceli dosiahnuť. Rozdiely medzi modelmi nie sú markantné, zdá sa, že sa v dátach nachádzajú vzorky, ktoré sú veľmi ťažko pochopiteľné a majú podľa všetkého skrytý význam, ktorý nie je na prvý „pohľad“ jasný.

[8] Rozdelenie na trénovaciu, validačnú a testovaciu množinu bolo prevedené stratifikovane – teda pomerne k počtom vzorkov na každú triedu s predom zvoleným náhodným seedom pre replikáciu

meno	#ep	train loss	valid loss
CZERT	17,00	0,55	**0,53**
FERNET	15,00	0,56	0,55
ROBECZECH	33,00	**0,53**	0,54
Slavic-BERT	20,00	0,68	0,67

Tabuľka 5.6: Zľava: Vyladený predtrénovaný model, počet epoch (kým trénovacia stratová funkcia dáva hodnoty nižšie ako evaluačná), hodnota stratovej funkcie na trénovacej a validačnej množine

V tabuľke 5.6 môžeme vidieť hodnoty stratových funkcií a počty epoch učenia transformerov.

meno	loss	f1	bal acc	acc	TOP_3 acc
ROBECZECH	0,64	90,60	90,85	90,64	97,30

Tabuľka 5.7: Sledované metriky výsledného modelu na testovacej množine.

Aby sme vedeli, akú má najlepší model prediktívnu schopnosť, otestovali sme ho na testovacej množine dátovej sady TEXT2BANK64, ktorá je 10% frakciou z celej sady. Tieto vzorky model predtým „nevidel“, takže sme zamedzili únikom dát. Hodnoty sledovaných metrík môžeme vidieť v tabuľke 5.7. RobeCzech dosiahol vynikajúce výsledky ako na validačnej, tak testovacej množine.

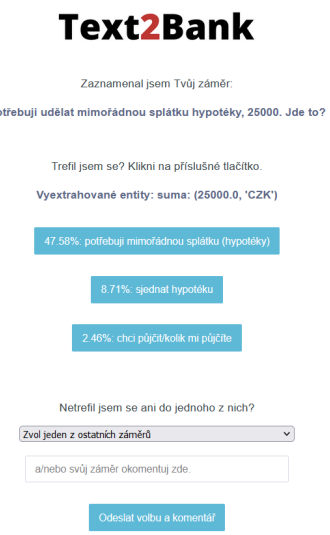
Presnosť na testovacej vzorke okolo hodnoty 90 je naozaj skvelým výsledkom a len dokazuje fakt, že je vhodné použiť vyladený predtrénovaný model na klasifikáciu textu.Tento vyladený model dokáže očividne zachytiť aj drobné odchýlky a skryté významy v jednotlivých vzorkách intentov. Model je teda schopný generalizovať, a preto bol nasadený do produkčnej verzie aplikácie. Trénovaciu a validačnú krivku stratovej funkcie môžeme vidieť v prílohe na obrázku C.4

5.4 Finálna verzia intent detection v aplikácii Text2Bank

Oproti prvej a druhej verzii demonštračnej aplikácie Text2Bank bola finálna verzia mierne upravená. Vyhodili sme nutnosť prihlasovať sa užívateľským menom a heslom – udržiavame iba zadané meno užívateľa a presný čas zadania svojho mena/emailu do aplikácie. Použitý intent detektor je model s najlepšou prediktívnou schopnosťou – RoBERTa model RobeCzech od ÚFALu. Túto demonštračnú aplikáciu môžete sami vyskúšať na webovej stránke [9]. Na

[9] `text2bank.profinit.cz`

(a) Ukážka úspešnej klasifikácie intentu „Nerozpoznávam platbu kartou“



(b) Ukážka úspešnej klasifikácie intentu „Potřebuji mimořádnou splátku“

Obr. 5.4: Ukážky úspešnej klasifikácie vstupov od užívateľa v aplikácii Text2Bank

obrázkoch 5.4 sú ukážky úspešnej klasifikácie užívateľských vstupov (trieda na prvom tlačidle je správna).

5.4.1 Modul neistoty

Po vzore iných konverzačných systémov, ako napríklad Siri od Apple [13] sme pridali dodatočný modul neistoty. Pokiaľ aplikácii zadáme napríklad vetu „Aké je dnes počasie?“, vyhodnocovať a klasifikovať túto vetu by nemal, pretože sa významovo nejedná o otázku z oblasti bankovníctva.

Modul neistoty funguje na princípe kosínovej podobnosti vzoriek všetkých dát všetkých intentov celej sady TEXT2BANK64. Po každom zadanom vstupe do aplikácie spočítame kos. podobnosť vstupu so všetkými vzorkami a ak najvyššia hodnota bude nižšia ako dopredu zvolená prahová hodnota, vstup vyhodnotíme ako irelevantný a oznámime užívateľovi, že ho aplikácia nedokáže spracovať a nevie, o čo sa jedná. Ukážku vstupu, pri ktorom si modul nie je istý vidíme v prílohe na obrázku C.5b

Na to, aby sme mohli spočítať kosínové podobnosti, musíme zo všetkých vzoriek a vstupu vytvoriť embeddingy, ktoré následne porovnávame. Po vyskúšaní modelov LASER, FERNET sentence embeddings a FastText sentence embeddings sa ukázalo, že najrozumnejšie bude použiť model FastText s prahovou hodnotou kos. podobnosti 0,7. Určili sme to po poslednom zbere dát, kde sa ukázalo, že práve táto hodnota pri týchto typov embeddingov vie naozaj dobre

rozlíšiť náhodný text / náhodné vstupy od užívateľov, od tých, ktoré intent detektor spracovať vie.

5.4.2 Reálne používanie finálnej verzie aplikácie Text2Bank

meno	acc	TOP_3 acc
ROBECZECH – real	57,96	78,34

Tabuľka 5.8: Presnosť (acc) a TOP_3 presnosť určená reálnymi užívateľmi.

Zozbierali sme 157 použiteľných vzoriek intentov od 33 reálnych užívateľov a klasifikovali zvoleným intent detektorom – RobeCzech. Len 8 zo všetkých užívateľov bolo ochotných napísať viac ako 5 vstupov. Výsledky ich testovania (spokojnosť s výstupom detektoru) môžeme vidieť na tabuľke 5.8. Z dát bolo viditeľné, že od intent detektoru chceli až 11 nových intentov, ktoré aplikácia nepoznala (vytvoriť nový účet, nahrať nové doklady, neprišla overovacia sms, atď.) Ako bolo popisované vyššie, úloha intent detektoru je v konverzačnom systéme veľmi dôležitá, pretože dáta ukázali, že ak po čase prestal intent detektor na neznáme intenty reagovať, užívatelia mali tendenciu odchádzať.

Je zrejmé, že intent detektor trénovaný na malý počet intentov nemôže pokryť celú škálu problémov, ktoré klienti banky môžu mať. Preto je táto presnosť vcelku dobrý výsledok. Priepasť medzi TOP_3 a obyčajnou presnosťou na tabuľke 5.8 však ukazuje, že model nemal k dispozícii dosť široké spektrum trénovacích dát (dostatočne rozmanité vzorky), aby mohol naozaj kvalitne generalizovať. Práve preto je úloha intent detection náročná, pretože nie sú voľne k dispozícii naozaj reálne ale kvalitné dáta od užívateľov, nieto v českom jazyku.

Zaujímavé bolo sledovať, že na niektoré jednoduché vstupy v slovenčine[10] bol intent detektor schopný reagovať a správne klasifikovať. To len potvrdzuje, že použiť predtrénovaný model na veľkom objeme dát je na takúto úlohu rozumné.

[10] napríklad vstup „Nemám na borovičku“ bol správne klasifikovaný ako „chci půjčit/kolik mi půjčíte“

Záver

Cieľom a obsahom tejto práce bolo vytvorenie modulu chatbota pre intent detection (detekcia kontextov, úmyslov) v doméne internetového bankovníctva v českom jazyku.

Urobili sme rešerš v oblasti intent detection a tým sme získali prehľad o najmodernejších používaných metódach na riešenie tohto problému. Popísali a použili sme rôzne techniky spracovávania a porozumenia prirodzeného jazyka.

Vytvorili sme základný, pomerne jednoduchý model intent detektoru pre zber dát. Zozbierali sme niekoľko stoviek vzoriek 13 intentov, ktoré sme neskôr obohatili o verejne dostupné vzorky z dátovej sady BANKING77, ktoré boli preložené do češtiny.

Analyzovali sme kvalitu dátových sád metódami strojového učenia a pred samotným návrhom intent detektoru sme získané dáta predspracovali. Potom sme vyladili model schopný klasifikácie až 64 rôznych intentov v doméne českého bankovníctva a vytvorili aplikáciu na demonštráciu jeho funkčnosti. Demo aplikácia chatbota je verejne dostupná na webovej stránke[11].

Zo skúmaných techník klasifikácie textu (konkrétne intent detection) sme vybrali predtrénovaný model RobeCzech [60], ktorý sme vyladili na našu špecifickú úlohu a dosiahli sme dobré výsledky (balancovaná presnosť 90% na testovacej vzorke). Tiež sme do aplikácie zaradili doplnkový modul neistoty.

Ďalšími krokmi by mohlo byť získanie spolupráce s akoukoľvek bankou so záujmom o chatbota, a tým zisk väčšieho množstva trénovacích dát s konkrétnejšími intentami (povedzme zo záznamov zákaznickej podpory). Takto získané reálne dáta by pomohli pokryť takmer celé spektrum problémov, ktoré ľudia veľmi často s bankou riešia a vďaka intent detektoru by bola podpora

[11] `text2bank.profinit.cz`

klientom banky automatizovateľná.

Ďalším dôležitým rozšírením by bola tvorba scenára neskoršej komunikácie s chatbotom pre akýkoľvek intent, ktorý vyžaduje ďalšie získanie informácií od užívateľa a prípadne samotné vykonanie príkazu alebo odpoveď na jeho otázku ohľadom bankovníctva, aby bol intent detection modul naplno využitý.

Literatúra

[1] Muráň, J.: Úvod do neurónových sietí - Umelá Inteligencia.sk. 2019. Dostupné z: https://umelainteligencia.sk/uvod-do-neuronovych-sieti/

[2] Bressmann, T.: Self-inflicted cosmetic tongue split: A case report. *Journal of the Canadian Dental Association*, ročník 70, č. 3, 2004: s. 156–157, ISSN 14882159, arXiv:1011.1669v3. Dostupné z: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[3] Artetxe, M.; Schwenk, H.: Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, ročník 7, 2019: s. 597–610, doi: 10.1162/tacl_a_00288, 1812.10464. Dostupné z: https://github.com/pytorch/fairseq

[4] Vaswani, A.; Shazeer, N.; Parmar, N.; aj.: Attention is all you need. In *Advances in Neural Information Processing Systems*, ročník 2017-Decem, Neural information processing systems foundation, jun 2017, ISSN 10495258, s. 5999–6009, doi:10.48550/arxiv.1706.03762, 1706.03762. Dostupné z: https://arxiv.org/abs/1706.03762v5

[5] Wolf, T.; Debut, L.; Sanh, V.; aj.: HuggingFace's Transformers: State-of-the-art Natural Language Processing. oct 2019, doi:10.48550/arxiv.1910.03771, 1910.03771. Dostupné z: https://arxiv.org/abs/1910.03771v5

[6] Turing, A. M.: Computer Machinery and Intelligence. *Mind*, ročník LIX, č. 236, oct 1950: s. 433–460, ISSN 00264423, doi:10.1093/MIND/LIX.236.433. Dostupné z: https://academic.oup.com/mind/article/LIX/236/433/986238http://mind.oxfordjournals.org/cgi/doi/10.1093/mind/XVIII.1.326

[7] Weizenbaum, J.: ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, ročník 9, č. 1, jan 1966: s. 36–45, ISSN 15577317, doi:10.1145/365153.365168. Dostupné z: https://dl.acm.org/doi/abs/10.1145/365153.365168

[8] Freund, Y.; Schapire, R. E.: Large margin classification using the perceptron algorithm. *Machine Learning*, ročník 37, č. 3, dec 1999: s. 277–296, ISSN 08856125, doi:10.1023/A:1007662407062. Dostupné z: https://link.springer.com/article/10.1023/A:1007662407062

[9] Turing, A. M.: COMPUTING MACHINERY AND INTELLIGENCE. Technická zpráva, 1950.

[10] Searle, J. R.: Minds, brains, and programs. *Behavioral and Brain Sciences*, ročník 3, č. 3, 1980: s. 417–424, ISSN 14691825, doi:10.1017/S0140525X00005756. Dostupné z: https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/abs/minds-brains-and-programs/DC644B47A4299C637C89772FACC2706A

[11] Hauser, L.: Searle's Chinese Box: Debunking the Chinese Room Argument. *Minds and Machines 1997 7:2*, ročník 7, č. 2, 1997: s. 199–226, ISSN 1572-8641, doi:10.1023/A:1008255830248. Dostupné z: https://link.springer.com/article/10.1023/A:1008255830248

[12] Cole, D.: The Chinese Room Argument. In *The {Stanford} Encyclopedia of Philosophy*, editace E. N. Zalta, Metaphysics Research Lab, Stanford University, {w}inter 2 vydání, 2020.

[13] Apple Inc.: Siri - Apple. 2011. Dostupné z: https://www.apple.com/siri/

[14] Google: Google Assistant — Your own personal Google. 2019. Dostupné z: https://assistant.google.com/https://assistant.google.com/{%}0Ahttps://assistant.google.com/intl/en{_}uk/

[15] Bank of America: Erica - Virtual Financial Assistant From Bank of America. Dostupné z: https://promotions.bankofamerica.com/digitalbanking/mobilebanking/erica

[16] Wen, T.-H.; Vandyke, D.; Mrkši´cmrkši´c, N.; aj.: A Network-based End-to-End Trainable Task-oriented Dialogue System. *the Association for Computational Linguistics*, ročník 1, 2017: s. 438–449.

[17] Casanueva, I.; Temčinas, T.; Gerz, D.; aj.: Efficient Intent Detection with Dual Sentence Encoders. Association for Computational Linguistics (ACL), mar 2020, s. 38–45, doi:10.18653/v1/2020.nlp4convai-1.5, 2003.04807. Dostupné z: https://arxiv.org/abs/2003.04807v1

[18] Hemphill, C. T.; Godfrey, J. J.; Doddington, G. R.: The {ATIS} Spoken Language Systems Pilot Corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, {P}ennsylvania, June 24-27,1990*, 1990. Dostupné z: https://aclanthology.org/H90-1021

[19] Raux, A.; Langner, B.; Bohus, D.; aj.: Let's Go Public! Taking a spoken dialog system to the real world. In *9th European Conference on Speech Communication and Technology*, 2005, s. 885–888, doi:10.21437/interspeech.2005-399.

[20] Le, T. A.: Sequence labeling approach to the task of sentence boundary detection. In *ACM International Conference Proceeding Series*, Association for Computing Machinery, jan 2020, ISBN 9781450376310, s. 144–148, doi:10.1145/3380688.3380703.

[21] Lorenc, P.: Joint model for intent and entity recognition. 2021, 2109.03221. Dostupné z: https://github.com/petrLorenc/poster2020http://arxiv.org/abs/2109.03221

[22] Wang, Y.; Shen, Y.; Jin, H.: A Bi-model based RNN Semantic Frame Parsing Model for Intent Detection and Slot Filling. Technická zpráva.

[23] Mehri, S.; Eric, M.: Example-Driven Intent Prediction with Observers. oct 2021, s. 2979–2992, doi:10.18653/v1/2021.naacl-main.237, 2010.08684. Dostupné z: http://arxiv.org/abs/2010.08684

[24] Xue, S.; Ren, F.: Intent-enhanced attentive Bert capsule network for zero-shot intention detection. *Neurocomputing*, ročník 458, oct 2021: s. 1–13, ISSN 18728286, doi:10.1016/j.neucom.2021.05.085.

[25] Devlin, J.; Chang, M. W.; Lee, K.; aj.: BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, ročník 1, Association for Computational Linguistics (ACL), oct 2019, ISBN 9781950737130, s. 4171–4186, doi:10.48550/arxiv.1810.04805, 1810.04805. Dostupné z: https://arxiv.org/abs/1810.04805v2

[26] Liu, X.; Eshghi, A.; Swietojanski, P.; aj.: Benchmarking Natural Language Understanding Services for Building Conversational Agents. In *Lecture Notes in Electrical Engineering*, ročník 714, 2021, s. 165–183, doi:10.1007/978-981-15-9323-9_15, 1903.05566. Dostupné z: https://www.luis.ai/home

[27] Larson, S.; Mahendran, A.; Peper, J. J.; aj.: An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the*

2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics, 2019, s. 1311–1316, doi:10.18653/v1/D19-1131. Dostupné z: https://aclanthology.org/D19-1131

[28] Flach, P.: *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, 2012, ISBN 1107422221, 9781107422223, 396 s.

[29] Oladipupo, T.: Types of Machine Learning Algorithms. In *New Advances in Machine Learning*, IntechOpen, feb 2010, ISBN 978-953-307-034-6, doi:10.5772/9385. Dostupné z: https://www.intechopen.com/chapters/10694

[30] Tolles, J.; Meurer, W. J.: Logistic regression: Relating patient characteristics to outcomes. aug 2016, doi:10.1001/jama.2016.7653. Dostupné z: https://jamanetwork.com/journals/jama/fullarticle/2540383

[31] Karel Klouda, Juan Pablo Maldonado Lopez, D. V.: *Využití Bayesovy věty*. 2022.

[32] Zhang, H.: The optimality of naive Bayes, flairs conference. 2004. Dostupné z: www.aaai.org

[33] Buitinck, L.; Louppe, G.; Blondel, M.; aj.: {API} design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, s. 108–122.

[34] Klouda, Karel, Kovalenko, A.; Pablo Maldonado Lopez, J.; Vašata: BI-VZD přednáška 4. 2022. Dostupné z: https://courses.fit.cvut.cz/BI-VZD/lectures/files/BI-VZD-03-cs-slides.pdf

[35] Van Der Maaten, L.; Hinton, G.: Visualizing data using t-SNE. *Journal of Machine Learning Research*, ročník 9, 2008: s. 2579–2625, ISSN 15324435.

[36] McInnes, L.; Healy, J.; Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. feb 2018, doi:10.48550/arxiv.1802.03426, 1802.03426. Dostupné z: https://arxiv.org/abs/1802.03426v3http://arxiv.org/abs/1802.03426

[37] SAZLI, M. H.: A brief review of feed-forward neural networks. *Communications, Faculty Of Science, University of Ankara*, 2006: s. 11–17, doi:10.1501/0003168.

[38] Chiu, J. P.; Nichols, E.: Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, ročník 4, 2016: s. 357–370, doi:10.1162/tacl_a_00104, 1511.08308. Dostupné z: http://nlp.stanford.edu/projects/glove/

[39] Wikimedia: Wikimedia Downloads. 2013: str. 2013. Dostupné z: https://dumps.wikimedia.org/

[40] Mikolov, T.; Chen, K.; Corrado, G.; aj.: Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 2013, 1301.3781. Dostupné z: http://ronan.collobert.com/senna/

[41] Grave, E.; Bojanowski, P.; Gupta, P.; aj.: Word vectors for 157 languages FastText. 2019. Dostupné z: https://fasttext.cc/docs/en/crawl-vectors.html

[42] Mikolov, T.; Grave, E.; Bojanowski, P.; aj.: Advances in pre-training distributed word representations. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2019, ISBN 9791095546009, s. 52–55, 1712.09405. Dostupné z: https://commoncrawl.org/2017/06

[43] Artetxe, M.; Schwenk, H.: Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, ročník 7, 2019: s. 597–610, doi:10.1162/tacl_a_00288, 1812.10464. Dostupné z: http://www.isthe.com/chongo/tech/comp/fnv

[44] Rush, A.: The Annotated Transformer. 2019, s. 52–60, doi:10.18653/v1/w18-2509. Dostupné z: http://nlp.seas.harvard.edu/2018/04/03/attention.html

[45] He, K.; Zhang, X.; Ren, S.; aj.: Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 2016-Decem, 2016, ISBN 9781467388504, ISSN 10636919, s. 770–778, doi:10.1109/CVPR.2016.90, 1512.03385. Dostupné z: http://image-net.org/challenges/LSVRC/2015/

[46] Radford, A.; Narasimhan, T.; Salimans, T.; aj.: [GPT-1] Improving Language Understanding by Generative Pre-Training. In *Preprint*, 2018, s. 1–12. Dostupné z: https://gluebenchmark.com/leaderboard

[47] Jordan, M. I.; LeCun, Y.; Solla, S. A.; aj.: Advances in neural information processing systems : proceedings of the first 12 conferences. 2001: s. 3079–3087.

[48] Howard, J.; Ruder, S.: Universal language model fine-tuning for text classification. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, ročník 1, 2018, ISBN 9781948087322, s. 328–339, doi:10.18653/v1/p18-1031, 1801.06146. Dostupné z: http://nlp.fast.ai/ulmfit.

[49] Wu, Y.; Schuster, M.; Chen, Z.; aj.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. 1609.08144v2.

[50] Taylor, W. L.: "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Quarterly*, ročník 30, č. 4, 1953: s. 415–433, ISSN 0022-5533, doi:10.1177/107769905303000401.

[51] Peltarion: English BERT. Dostupné z: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/english-bert

[52] Křen, M.; Cvrček, V.; Čapka, T.; aj.: {SYN} v4: large corpus of written Czech. 2016, doi:11234/1-1846. Dostupné z: http://hdl.handle.net/11234/1-1846

[53] Raffel, C.; Shazeer, N.; Roberts, A.; aj.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, ročník 21, oct 2020: s. 1–67, ISSN 15337928, doi:10.48550/arxiv.1910.10683, 1910.10683. Dostupné z: https://arxiv.org/abs/1910.10683v1

[54] czes. 2011. Dostupné z: http://hdl.handle.net/11858/00-097C-0000-0001-CCCF-C

[55] Majliš, M.: {W2C} – Web to Corpus – Corpora. 2011. Dostupné z: http://hdl.handle.net/11858/00-097C-0000-0022-6133-9

[56] Lan, Z.; Chen, M.; Goodman, S.; aj.: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. sep 2019, doi:10.48550/arxiv.1909.11942, 1909.11942. Dostupné z: https://arxiv.org/abs/1909.11942v6http://arxiv.org/abs/1909.11942

[57] Sido, J.; Pražák, O.; Pribán, P.; aj.: Czert - Czech BERT-like Model for Language Representation. In *International Conference Recent Advances in Natural Language Processing, RANLP*, Incoma Ltd, mar 2021, ISBN 9789544520724, ISSN 13138502, s. 1326–1328, doi:10.26615/978-954-452-072-4_149, 2103.13031. Dostupné z: https://arxiv.org/abs/2103.13031v3

[58] Lehečka, J.; Švec, J.: Comparison of Czech Transformers on Text Classification Tasks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ročník 13062 LNAI, Springer Science and Business Media Deutschland GmbH, jul 2021, ISBN 9783030895785, ISSN 16113349, s. 27–37, doi:10.1007/978-3-030-89579-2_3, 2107.10042. Dostupné z: https://arxiv.org/abs/2107.10042v1

[59] Liu, Y.; Ott, M.; Goyal, N.; aj.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. jul 2019, doi:10.48550/arxiv.1907.11692, 1907.11692. Dostupné z: https://arxiv.org/abs/1907.11692v1http://arxiv.org/abs/1907.11692

[60] Straka, M.; Náplava, J.; Straková, J.; aj.: RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ročník 12848 LNAI, Springer Science and Business Media Deutschland GmbH, may 2021, ISBN 9783030835262, ISSN 16113349, s. 197–209, doi:10.1007/978-3-030-83527-9_17, 2105.11314. Dostupné z: http://arxiv.org/abs/2105.11314http://dx.doi.org/10.1007/978-3-030-83527-9{_}17

[61] Blei, D. M.; Ng, A. Y.; Jordan, M. I.; aj.: Conditional random fields: An introduction. *Neural Computation*, ročník 18, č. 4-5, 2004: s. 1–9, ISSN 15324435, doi:10.1162/jmlr.2003.3.4-5.993, 1111.6189v1. Dostupné z: http://www.cs.princeton.edu/{~}blei/lda-c/{%}5Cnpapers2://publication/doi/10.1162/jmlr.2003.3.4-5.993{%}5Cnpapers2://publication/uuid/4001D0D9-4F9C-4D8F-AE49-46ED6A224F4A{%}5Cnpapers2://publication/uuid/7D10D5DA-B421-4D94-A3ED-028107B7F9B6{%}5Cnhttp://www.crossref.

[62] Arkhipov, M.; Trofimova, M.; Kuratov, Y.; aj.: Tuning Multilingual Transformers for Language-Specific Named Entity Recognition. Association for Computational Linguistics (ACL), sep 2019, s. 89–93, doi:10.18653/v1/w19-3712. Dostupné z: https://aclanthology.org/W19-3712

[63] Andoni, A.; Onak, K.: Approximating edit distance in near-linear time? In *SIAM Journal on Computing*, ročník 41, 2012, ISSN 00975397, s. 1635–1648, doi:10.1137/090767182, 1109.5635.

[64] Yan, X.; K Lau, R. Y.; Li, X.; aj.: Toward a Semantic Granularity Model for Domain-Specific Information Retrieval. *ACM Trans. Inf. Syst*, ročník 29, 2011, doi:10.1145/1993036.1993039. Dostupné z: http://doi.acm.org/10.1145/1993036.1993039

[65] Vašata, D.: Evaluace Modelů, přednáška 2 kuzru NI-ADM. 2022: str. 6. Dostupné z: https://courses.fit.cvut.cz/NI-ADM/lectures/files/NI-ADM-02-en-handout.pdf

[66] Bird, S.; Klein, E.; Loper, E.: *Natural language processing with Python: analyzing text with the natural language toolkit*. Ö'Reilly Media, Inc.", 2009.

[67] Domke, M.: schwifty · PyPI. Dostupné z: https://pypi.org/project/schwifty/

[68] Richter, M.; Stra\vnák, P.; Rosen, A.: Korektor–A System for Contextual Spell-checking and Diacritics Completion. In *Proceedings of the 24th International Conference on Computational Linguistics (Coling 2012)*, editace M. Kay; C. Boitet, {IIT} Bombay, Mumbai, India: Coling 2012 Organizing Committee, 2012, s. 1–12.

[69] Google: Cloud Computing Services — Google Cloud. 2019. Dostupné z: https://cloud.google.com/https://cloud.google.com/{%}0Ahttps://cloud.google.com/?utm{_}source=youtube{&}utm{_}medium=unpaidsocial{&}utm{_}campaign=cka-20190408-management-of-smart-buildings{%}0Ahttps://cloud.google.com/

[70] Google: Cloud Translation API — Google Cloud. Dostupné z: https://cloud.google.com/translate/docs/apishttps://cloud.google.com/translate/pricing

Dodatok **A**

Analýza dát TEXT2BANK13 (zozbierané dáta z Profinitu)

V tejto časti analýzy sa zaoberáme hlavne vyťažovaním informácií o (1) štruktúre dátovej sady, (2) sémantickej granularite vzoriek intentov, (3) podobností vzoriek medzi sebou, (4) podobností vzoriek intentov s názvami príslušnej triedy.

Celá analýza sa nachádza v priloženom súbore `explore_simil_t2b.ipynb`.

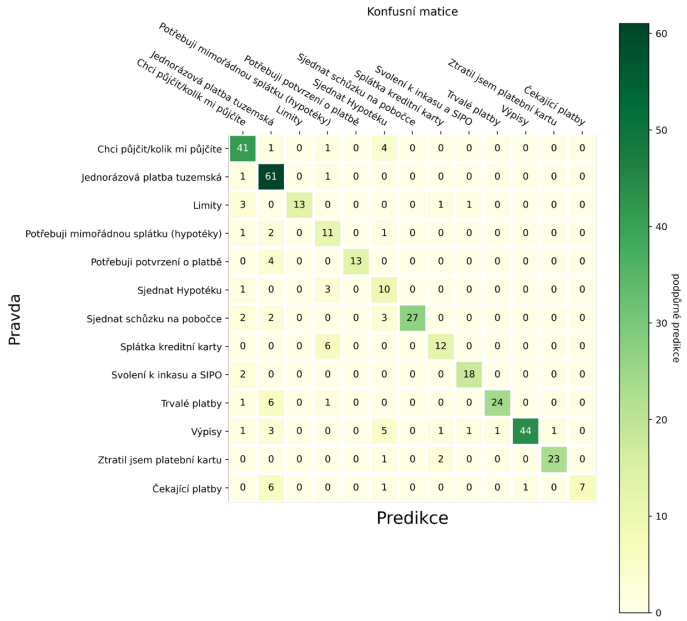
A.1 Predikcie druhého klasifikátoru Text2Bank

ktorý vznikol po úprave prvého modelu môžeme vidieť na obrázku A.1. Jedná sa o Konfúznu maticu predikcií užívateľských vstupov počas zberu dát. Os x: počty predikcií vzorky danej triedy, os y: pravdivé hodnoty tried klasifikácie.

A.2 Kosínové podobnosti vzoriek $K(x_i, x_j)$ v rámci triedy „limity“

Z tohto intentu vyberieme 5 vzoriek pre lepšiu vizualizáciu (matica bola skonštruovaná najprv na všetkých vetách z triedy limity a potom boli vybraté najzaujímavejšie vzorky). Chceme ukázať či sa vzorky v rámci svojej triedy na seba podobajú. Použime FastText pre tvorbu vetných embeddingov a kosínovú podobnosť pre výpočet podobností.

Chceme ukázať, v akej kvalite sme dáta zozbierali, či sa v tejto triede nájdu aspoň dve-tri vety s rovnakou sémantikou, poskytnúť malú ukážku akú má táto náhodne zvolená trieda sémantickú granularitu. Na obázku A.2 vidíme kosínové podobnosti: $K(x_i, x_j)$, kde x_k je vzorka z triedy „limity“. Očakávanie pri týchto vizualizáciách boli vysoké čísla na celej matici podobnosti, to sa stalo len v niektorých prípadoch. Vzorky „jaký mám limit na kartě“ a „jaký mám limit na *platební* kartě“ s hodnotou 0,91: jedno slovo pridané do úplne rovnakej vety nám dá vysokú mieru podobnosti, podľa očakávaní. Oproti tomu



Obr. A.1: Konfúzna matica predikcií klasifikácie užívateľských vstupov počas zberu dát

vety „kolik mám maximálne povolený výběr z bankomatu“ a „limit platby na internetu“ majú nízku mieru podobnosti s hodnotou 0,45. Jedná sa o úplne odlišné zámery, avšak oba patria v tejto sade do tej istej triedy. Napríklad „limit platby na internetu“ je nepodobná so skoro všetkými vybranými vetami. Táto veta má však zvláštnu vetnú štruktúru a slovosled, je možné, že je to spôsobené práve týmto.

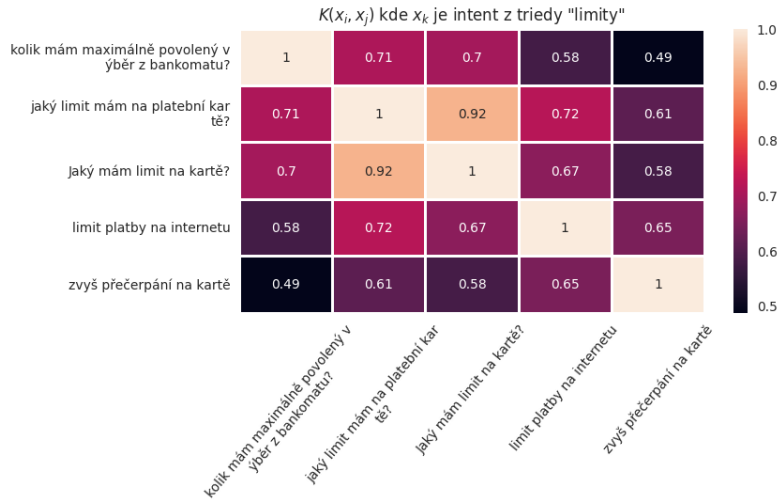
Granularita triedy „limity“ je príliš nízka, preto sme sa tento zámer z dátovej sady rozhodli úplne vyhodiť.

A.3 UMAP z FastText embeddingov a KMeans klastering všetkých intentov, každá trieda separátne

Pozrime sa na UMAP embeddingy z FastText embeddingov, pre lepšiu vizualizáciu pomocou zhlukovacieho algoritmu KMeans, nájdime pre každú triedu tri zhluky a farebne odlíšme každý zhluk v 2D UMAP priestore. Vypíšme cca tretinu viet z každého nájdeného klastru.

Chceme overiť hypotézu či existujú triedy, kde vetné embeddingy intentov k príslušnej triede sú navzájom veľmi odlišné. Ak existujú, treba ich nájsť a vytvoriť prípadné podtriedy a ak neexistujú, triedu do podtried nerozbíjame[12]

[12]Parametre UMAPu boli: n_neighbors: 5, min_dist: 0,001, metric: cosine



Obr. A.2: Kosínové podobnosti $K(x_i, x_j)$ z triedy „limity“

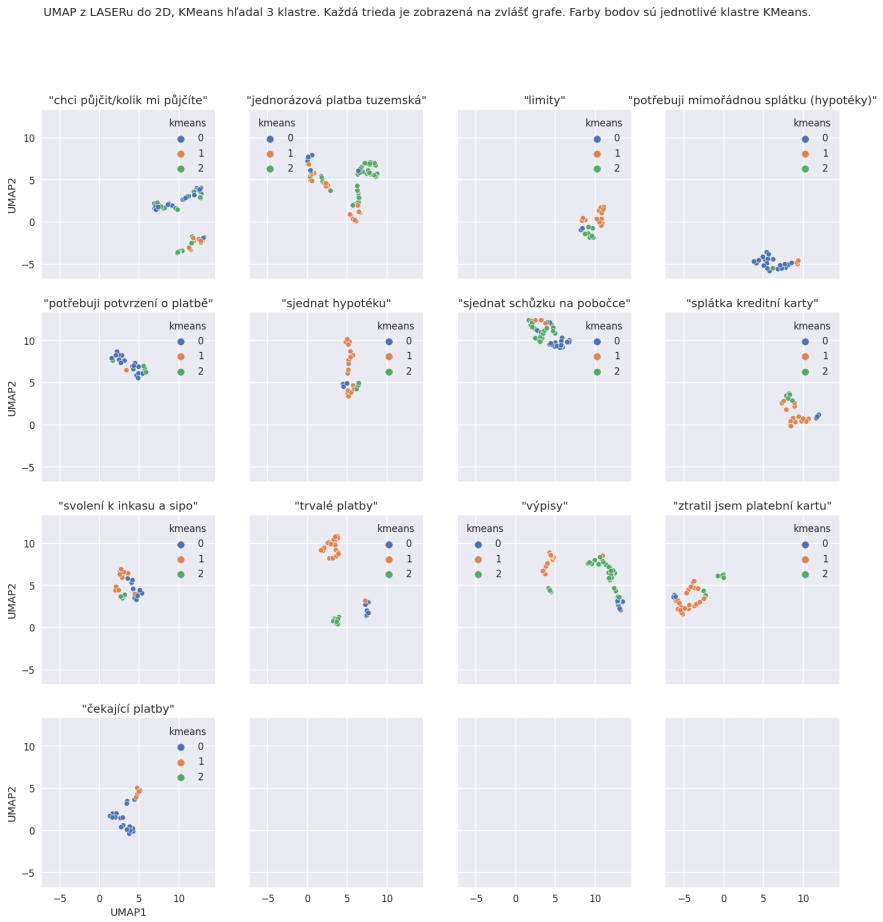
Grafy na obrázku A.3 ukazujú UMAP zobrazenie FastText embeddingov vzorkov pre každú triedu zvlášť do 2D priestoru. Farby bodov na grafoch sú podľa čísla zhluku nájdeného pomocou KMeans algoritmu

V priloženej analýze `explore_simil_t2b.ipynb` bola vypísaná náhodná tretina vzoriek z každého zhluku pre každú triedu zvlášť. Popíšme si niektoré zaujímavé triedy:

- pri triede „chci si půjčit“ vidíme, že klaster 2 združuje vety s číslovkou v texte, klaster 0 sú len otázky,
- triedu „limity“ UMAP ani KMeans nebol schopný rozdeliť do viac ako dvoch klastrov – v klastri 2 vidíme číslovky,
- trieda „trvalé platby“ boli UMAPom aj KMeans rozdelené na 3 klastre, kde 0 reprezentuje dlhšie a komplikovanejšie vety, v 1 sa vyskytuje spojenie „trvalé platby“ a v 2 „trvalý příkaz“,
- trieda „svolení k inkasu a SIPO“ je KMeansom naklastrované v podobnom duchu ako „chci si půjčit“,
- ostatné triedy sa alebo ťažko interpretujú alebo nie sú zaujímavé.

FastText je očividne náchylný nielen na dĺžku vety, ale aj na čísla vo vetách. Preto pri embeddovaní viet pri výslednom modeli čísla z vety vyhadzujeme pre väčšiu presnosť. Hypotéza sa, zdá sa, potvrdila len pre niektoré triedy (intent ktorý je otázkou má zjavne iný zámer ako nejaká akcia/príkaz).

Síce sa na obr. A.3 javí, že napr. trieda „jednorázová platba tuzemská“ má tak 3 zhluky, pri výpise vzorku viet v tejto triede bolo jasné, že odlíšenie je minimálne a závisí to len na čísliciach alebo dĺžke textu, pri iných sú to práve otázky, ktoré priestor rozdeľujú. Mnohé triedy (napr. limity) sú po zobrazení do UMAP 2D priestoru pokope. Vzorky sú tu však podľa predchádzajúceho



Obr. A.3: UMAP z FastText embeddingov pre každú triedu zvlášť

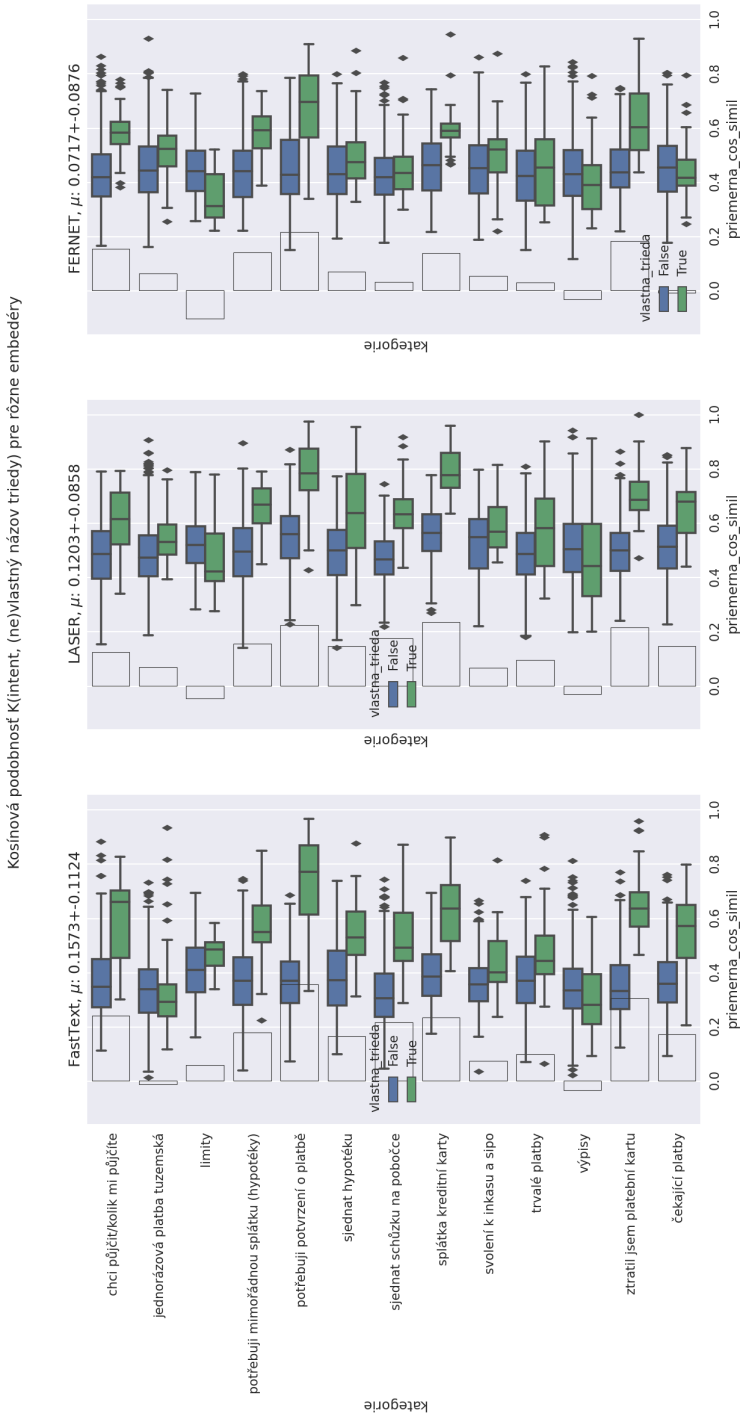
skúmania dosť rozličné. FastText nám teda pri rozdeľovaní do podtried asi nepomôže.

A.3.1 Kosínová podobnosť K(intent, vlastný názov triedy) pre tri embedéry

Pozrime sa na embeddingy všetkých intentov a názvov všetkých tried a porovnajme týmto niektoré vybrané modely, ktoré embeddingy vedia vytvoriť (FastTest, LASER FERNET sentence embeddings) navzájom medzi sebou pomocou kosínovej podobnosti K(intent, názov (ne)príslušnej triedy)

Na obrázku A.4 vidíme boxploty kosínovej podobnosti medzi intentom a názvom triedy.

- Zelené boxploty sú vzorky intentov porovnané s názvom príslušnej triedy $K(x_{t_i}, c_t)$ (x_{t_i} sú vzorky triedy č. t a c_t je názov triedy t)



Obr. A.4: Porovnanie sentence embederov

- Modré boxploty sú hodnoty podobnosti vzoriek so všetkými ostatnými názvami tried $K(x_{t_i}, c_{o \neq t})$.

Vetné embedingy boli vytvorené za pomoci FastText, LASER a FERNET sentence embeddings. Barplot naľavo každého grafu je rozdiel príslušných a neprísluš ných hodnôt podobností, aby sme videli, ako dobre vie daný model dané vzorky z triedy zaradiť a či to nebolo náhodné. Hodnoty μ v popise každého z troch grafov je priemer týchto rozdielov, $\pm$ smerodajná odchýlka. Platí, že čím vyšší priemer, tým lepšia schopnosť „klasifikovať“ danú vzorku do správnej triedy.

Nízke hodnoty podobností sa nachádzajú hlavne pri málovravných názvoch ako „výpisy“ a „limity“. Jediný FastText si s nimi vedel ako-tak poradiť. a vysoké hodnoty podobností sa nachádzajú hlavne pri triede „potvrzeni o platbe“ a teda modely vedia na tejto triede najlepšie rozlišovať (granularita vzorkov je dostatočne vysoká).

Znova sa ukazuje, že je vhodnejšie použiť dlhší a konkrétnejší názov triedy a od toho by sa, samozrejme odvíjala granularita dát a taktiež by bola predpokladaná vyššia podobnosť K(intent, trieda). Nasvedčujú tomu vysoké hodnoty pri jednoduchších a menej obšírnych názvoch tried ako je napr. „potvrzeni o platbe“. V popredí sa drží FastText aj napriek malej dimenzii embeddingov. Avšak netreba zabúdať na tzv. „prekliatie dimenzie“, kde platí, že s vyššou dimenziou sú všetky body v priestore bližšie seba. Pripomeňme si veľkosti dimenzií, v ktorých vety reprezentujeme: LASER: 1024, FERNET: 768, fasttext: 300.

Ak by sme chceli vytvoriť klasifikáciu len na základe podobnosti embeddingov, dosiahli by sme nasledovné:

- **LASER: F1 40,46, presnosť 42,85**
- FERNET: F1 28,02, presnosť 28,36
- FastText: F1 32,26, presnosť 34,89

Napriek tomu, že sa FastText javil schopnejšie rozlišovať vzorky, LASER ho predbehol. Každopádne pre vysokú rozlišovaciu schopnosť bol FastText zvolený ako detektor istoty.

A.4 Záver analýzy

Bola prevedená analýza podobností rôznych viet z datasetu TEXT2BANK13 a boli hlbšie preskúmané podobnosti vybraných viet z 2 tried. Trieda limity má, zdá sa, príliš nízku sémantickú granularitu a bolo by vhodné ju vyhodiť zo sady. Ukázalo sa, že je vhodné použiť dlhší, konkrétnejší názov triedy a ako embedovaciu funkciu viet zvoliť FastText. Nižšia schopnosť LASERu

rozlišovať vety to bolo podľa všetkého vysokou dimenziou embedding priestoru. Klasifikovali sme celý dataset pomocou kľúčových slov, avšak výsledky tejto metódy berieme len ako prezentáciu toho, ako bola sada vytvorená.

Mnohé časti analýzy v práci vynechávame, avšak je k dispozícii v prioženom CD.

Dodatok **B**

Analýza dátovej sady BANKING77

V tejto analýze sa zaoberáme overovaním správnosti strojového prekladu intentov aj názvov tried, možnosťou klasifikácie len za pomoci embeddingov a kosínovej podobnosti intentu a názvu triedy a porovnáme 3 rôzne embedovacie modely – zobrazením embeddingov intentov pomocou UMAPu v 2D Priestore.

Celá analýza sa nachádza v priloženom súbore `explore_simil_b77.ipynb`.

B.1 Kosínová podobnosť LASER embeddingov názvov tried datasetu BANKING77

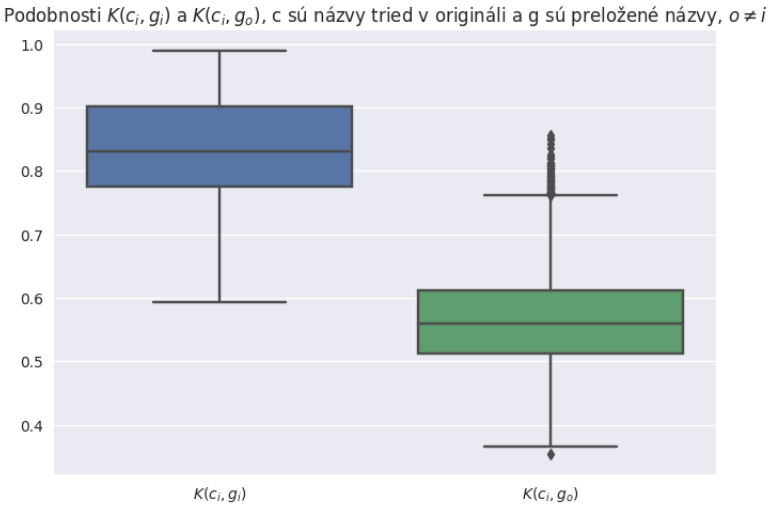
Preklady nemusia vždy vyjsť správne. Poďme odhaliť tie, ktoré nemusia byť dobre preložené.

Chceme ukázať kosínové podobnosti:

- $K(c_i, g_i)$, kde c_i sú názvy tried v origináli – angličtine a g_i sú preložené názvy do češtiny
- $K(c_i, g_o)$, kde c_i sú názvy tried v origináli – angličtine a g_o sú preložené názvy do češtiny, kde $o \neq i$
- pozrime sa na názvy tried, ktorých preklady neboli najpodobnejšie svojmu originálu v kontexte LASER embeddingov a kosínovej podobnosti.

Boxplot na obr. B.1 zobrazuje kos. podobnosti:

- naľavo: názov triedy k tej istej triede v inom jazyku,
- napravo: názov triedy k inému názvu triedy v inom jazyku



Obr. B.1: Kosínové podobnosti názvov tried v angličtine a češtine

Medián hodnôt kosínových podobností prekladov a originálov je nad 0,8, čo signalizuje vysokú mieru podobnosti a medián hodnôt kos. pod. originálov ku všetkým iným prekladom je okolo 0,5. To je nízka miera podobnosti, podľa očakávaní. Outliery s vysokými hodnotami signalizujú vysoké podobnosti s inými prekladmi a nízke hodnoty podobností signalizujú niekoľko horších prekladov.

V priloženej analýze zobrazujeme tabuľku s originálnym znením názvu triedy, preloženým názovm triedy, kosínovo najpodobnejší preklad, kosínovo druhý najpodobnejší preklad. Tabuľka neukazuje všetkých 77 tried, len tie, kde sa preklad nerovná prvej najvyššej podobnosti spomedzi všetkých prekladov názvov tried.

Napr. názov „pending card payment“, nemá ako prvú najvyššiu podobnosť so svojim prekladom „čekající platba kartou“, ale „poplatek za platbu kartou“. Sémanticky spolu názvy nesúvisia a podobnosť so správne preloženou triedou je až duhá v poradí. Všimnime si tiež názov „pending top up“, ktorý nemá ako prvú najvyššiu podobnosť so svojim prekladom „čeká na dobití“, ale „kolík blokován“ a dokonca ani druhý najpodobnejší preklad „dobíjení kartou“ nie je správny. Nepodobných prekladov je tu dohromady 9, čo je cca. 12% z celkového počtu tried.

Zdá sa, že preklady názvov tried B77 datasetu sú až na cca 12% prípadov správne (podľa kosínovej podobnosti LASER embeddingov). Po drobných úpravách prekladov by názvy tried mohli byť naozaj dostatočne rozličné a správne.

B.1.1 Podobnosť embeddingov rôznych modelov ako klasifikátor na sade BANKING77

Vytvorme vetné embeddingy pre všetky intenty a triedy datasetu BANKING77. Použime FastText, LASER a FERNET sentence embeddings. Vytvorme kosínové podobnosti pre každý intent s každým názvom triedy Upravme vektory k držiace podobnosť $k_i = K(x_i, c_j)$, $i \in 0..\#viet, j \in 0..\#tried$ kde x je vektor intentov a c je vektor unkátnych názvov tried do pravdepodobnostnej distribúcie.

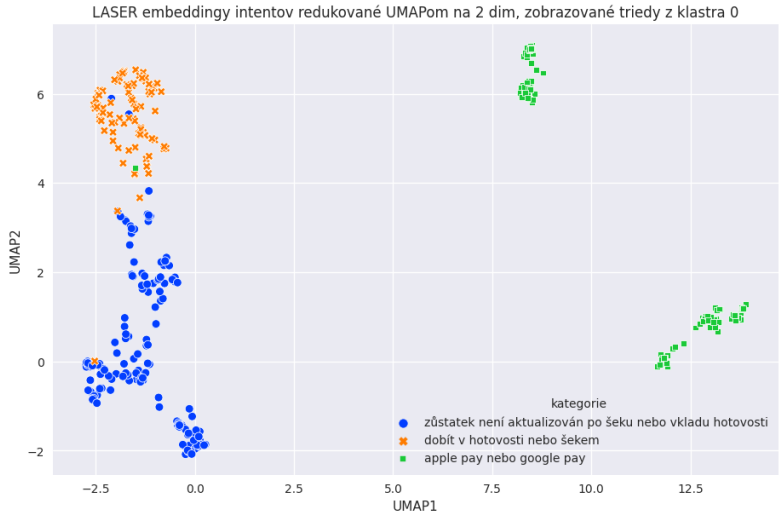
Ak by sme klasifikovali vzorky v angličtine (origináli) pomocou LASERu, výsledky sú vcelku slabé: acc: 33,55, f1: 34,34. V češtine tiež slabé: acc: 34,60, f1: 36,20. FERNET nebol schopný dať presnosť a f1 ani nad 10, a FastText sa držal na hodnotách okolo 20. Všetko je to veľmi málo a takéto modely by boli nepoužiteľné v praxi. Zero shot learning v tomto prípade neprichádza do úvahy.

B.1.2 Podobnosť intentov a názvu triedy kam patrí z b77 (české preklady)

Pozrime sa na hodnoty kos. podobností $K(x_i, c_k)$, kde x_i je intent patriaci do svojej pridruženej triedy c_k. Chceme tým ukázať podobnosti LASER embeddingov intentov a ich prislúchajúcich názvov tried. Zobrazme tabuľku, kde budú triedy zoradené podľa najvyššej priemernej podobnosti $K(x_i, c_k)$ spomenutej vyššie.

názvy tried	kos. pod.	počet
podpora země	0.390231	112
věkový limit	0.436819	92
přijímání peněz	0.440600	80
kompromitovaná karta	0.447557	72
propojení karet	0.452827	120
...	...	...
visa nebo mastercard	0.663339	115
aktivovat moji kartu	0.692173	141
získání virtuální karty	0.712787	82
virtuální karta nefunguje	0.738870	32
získat jednorázovou virtuální kartu	0.754992	78

Tabuľka ukazuje top 5 najmenej a najviac názvov tried, ktoré sa podobajú svojim vzorkom (v zmysle LASER embeddingov). Klasifikácia by pomocou tejto metódy bola možná, ale nedostačujúca.



Obr. B.2: LASER embeddingy zhluku 0 zobrazené pomocou UMAPu. Farba podľa príslušnej triedy.

B.2 UMAP z LASER embeddingov intentov z 8 klastrov tried

Použime KMeans na LASER embeddingoch názvov tried, pretože chceme zobraziť UMAP embeddingov intentov patriacich do rôznych tried. Celý UMAP priestor so všetkými triedami by bol nečitateľný a nič by sme tam nevideli. Všetky body by boli príliš natesno. Preto poďme nájsť triedy, ktoré sú si najviac podobné, aby sme mohli overiť, že sa niektoré až príliš podobajú alebo že by sme chceli nejaké intenty z nejakej triedy vložiť do inej, atď. Proste nájsť zaujímavé zistenia medzi triedami. Zobrazenie klastrovania je v priloženom analytickom súbore, ako aj vizualizácie, ktoré nie sú príliš zaujímavé a nezmestia sa tu.

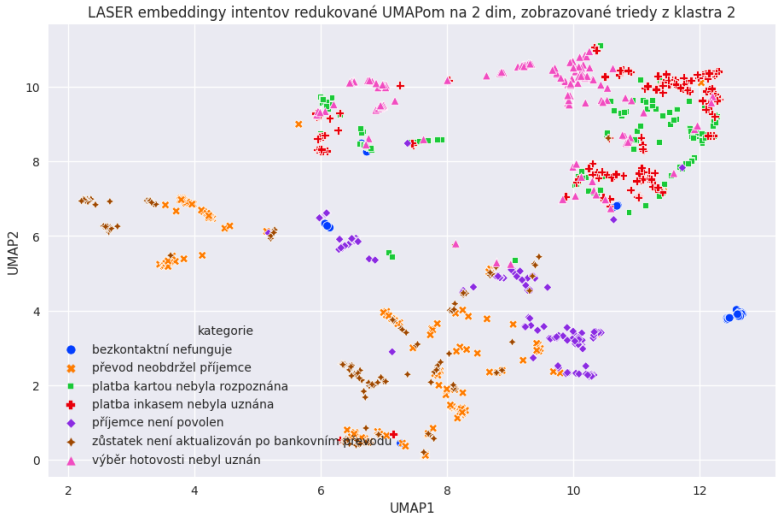
Pre každý nájdený klaster (zhluk) tried zobrazme UMAP redukciu LASER embeddingov intentov každej triedy

Hypotéza

- Existujú triedy, ktoré sa na seba sémanticky podobajú?
- Sú všetky triedy od seba sémanticky dostatočne odlišné?

B.2.1 Zhluk 0

Na obr. B.2 vidíme zobrazenie nultého klastru s troma triedami. Triedy sa zdajú byť dostatočne odlišné a zdá sa, že trieda „apple pay nebo google pay“ má dve podtriedy, čo môžeme vidieť na dvoch separátnych zoskupeniach na



Obr. B.3: LASER embeddingy zhluku 2 zobrazené pomocou UMAPu. Farba podľa príslušnej triedy.

UMAP redukcii zvyšné dve triedy sú odlíšené od seba, avšak je vidno, že sa na UMAP zobrazení spájajú Triedu „apple pay nebo google pay“ by sme možno chceli rozbiť na podtriedy na zvýšenie sémantickej granularity, ale neurobíme to, pretože samotný názov triedy signalizuje, že tu dve podtriedy majú byť.

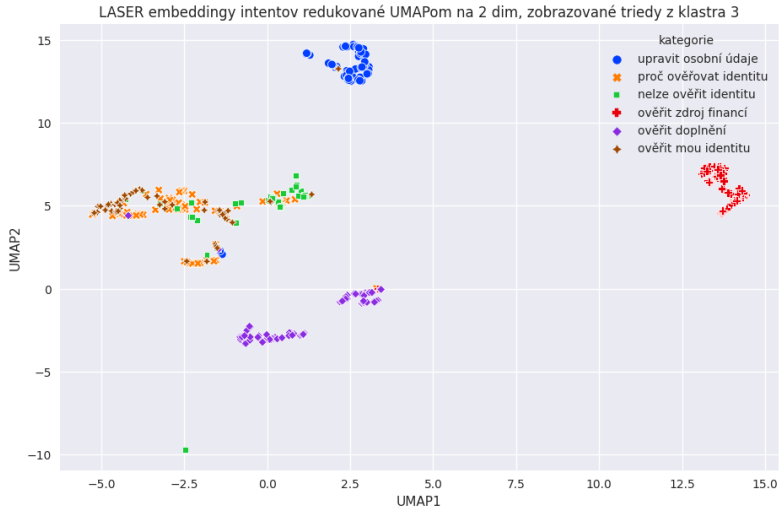
B.2.2 Zhluk 2

Na obr. B.3 vidíme zobrazenie LASER embeddingov vzoriek z klastru 2. Tried je tu viac, zdá sa, že si UMAP nevie dobre poradiť s embeddingami (napriek skúšaniu rôznych parametrov). Zoskupenie tried, ktoré sa javia najviac pomiešané, nazvime ho „uznania“:

- „platba kartou nebyla rozpoznána“,
- „platba inkasem nebyla uznána“,
- „výběr hotovosti nebyl uznán“,

Ďalšie zoskupenie tried, ktoré sa javí byť zmiešané, nazvime ho „prevody“:

- „převod neobdržel příjemce“ (tento preklad je veľmi zlý),
- „zůstatek není aktualizován po bankovním převodu“,
- trochu sa tu mieša aj „příjemce není povolen“, ale len okrajovo.



Obr. B.4: LASER embeddingy zhluku 3 zobrazené pomocou UMAPu. Farba podľa príslušnej triedy.

Triedy zo zoskupenia „uznania“ majú zjavne rozdielnu sémantiku, takže nie je dôvod na paniku. LASER a UMAP si s tým asi jednoducho nevedel dať rady.

Triedy zo zoskupenia „prevody“ majú asi tiež rozdielnu sémantiku, keďže sa intenty podľa všetkého líšia kontextuálne a na základe rozličných slov pre iný druh úkonu

B.2.3 Zhluk 3

Na obr. B.4 vidíme možný zhluk tried

- „proč ověřovat identitu“,
- „nelze ověřit identitu“,
- „ověřit mou identitu“,

inak sa triedy javia dostatočne odlišné. Trieda „ověřit doplnění“ je priestorovo odlíšená v rámci seba samej

Nie je dôvod na paniku, očividne je sémantika týchto tried s dostatočne vysokou granularitou, jediný potenciálny zhluk 3 tried je spôsobený podľa všetkého podobnými slovami v jednotlivých vzorkách

Trieda „ověřit doplnění“ má, zdá sa, dve podtriedy, avšak overili sme výpisom, že to tak nie je.

B.2.4 Záver pozorovaní zhlukov tried medzi sebou

Niektoré triedy vykazujú známky nízkej sémantickej granularity, avšak sú to skôr výnimky a asi kontextuálne ťažko pochopiteľné intenty, ktoré to spôsobujú.

Nenašli sa príliš veľké prekryvy (čo sa týka sémantiky jednotlivých intentov), aj keď sa tak môže zdať.

Verdikt: ukazuje sa, že tento dataset je dostatočne kvalitný

Pre kontext českého bankovníctva je však nutné mnohé triedy **vyhodiť**, pretože by boli nepoužiteľné, napríklad triedy, ktoré súvisia s top-up navýšením kreditu.

B.3 Záver analýzy

Pozreli sme sa na kvalitu prekladu intentov z b77 datasetu pomocou boxplotu – až na pár outlierov sa hodnoty hýbu okolo 0,85-0,93, čo je pomerne vysoká miera podobnosti

Preverili sme, že preklady názvov tried sú rozumne skonštruované – znova až na pár (cca 12%) výnimiek sú názvy dostatočne podobné v zmysle kos. podobnosti embeddingov

Klasifikovali sme intenty pomocou rôznych embedovacích modelov a kosínovej podobnosti s názvom príslušnej triedy – sledované metriky ukázali, že najvodnejší model je LASER, avšak jeho „klasifikačná“ schopnosť je veľmi nízka ako pre orignál anglické dáta, tak pre české (accuracy 0,33 pre originály a 0,34 pre české preklady)

Pomocou tabuľky a boxplotov sme ukázali, že triedy vykazujú veľmi nízku podobnosť intentov k názvu triedy a tie, ktoré vykazujú vyššiu podobnosť, tak ich mediánové a priemerné hodnoty sú stále pomerne nízke - použitie LASERu ako klasifikátora teda naozaj nie je vhodné

Pomocou KMeans sme rozdelili názvy tried do 8 klastrov a v každom klastri sme skúmali UMAP zobrazenia LASER embeddingov do 2D priestoru. - každý klaster vykazoval do vysokej miery oddeliteľnosť a distinktnosť intentov medzi triedami – pár výnimiek bolo spôsobených podľa všetkého alebo zlým prekladom, alebo nedokonalosťou LASER a UMAP embeddingov. - hlbšie sme tieto dôvody neskúmali, nie je to tak podstatné.

B.3.1 Verdikt

Zistili sme, že strojové preklady textov nie sú dokonalé, avšak postačujúce pre naše účely. Dáta sú v pomerne vysokej kvalite a s dostatočne vysokou sémantickou granularitou. Tento dataset je teda použiteľný pre naše účely, ale – je treba vyhodiť triedy, ktoré sú pre kontext českého bankovníctva irelevantné – názvy tried, ktoré po vyhodení ostanú je treba preložiť expertne – popri tomto všetkom manuálne skontrolovať chyby a vytiahnuť pre každú triedu reprezentatívne a kontextuálne rozdielne intenty pre potenciálnu tvorbu znalostnej báze

Dodatok C

Analýza zlučiteľnosti dát TEXT2BANK13 a BANKING77 dohromady

V tejto analýze sa venujeme podobnostiam názvov tried v oboch datasetoch a možnosti zlúčenia niektorých tried z oboch datasetov.

Celá analýza sa nachádza v priloženom súbore `explore_simil_b77VSt2b.ipynb`.

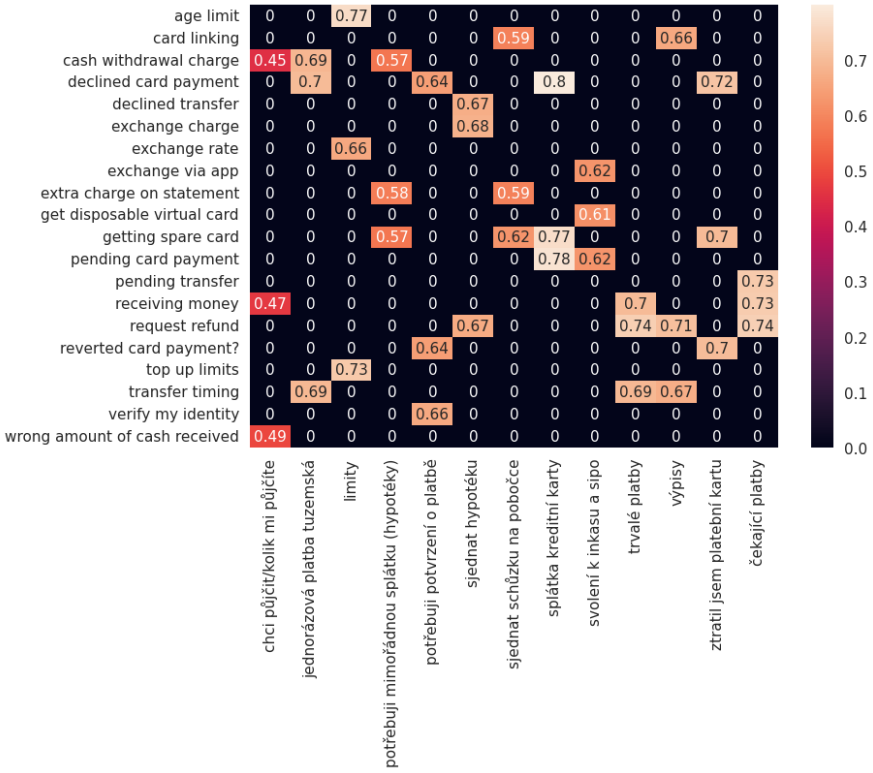
C.1 Podobnosť názvov tried datasetov BANKING77 a TEXT2BANK13

Chceme zistiť či sú nejaké triedy z oboch datasetov zlúčiteľné dohromady. Nájdime teda najpodobnejšie názvy tried (pomocou LASER embeddingov a kosínovej podobnosti) a vytvorme maticu podobností spočítanú pomocou $K(b_i, t_j)$, kde b_i sú názvy tried B77 datasetu a t_j sú názvy tried t2b datasetu. Môžeme ju vidieť na obr. C.1.

Na vizualizácii vidíme, že trieda „chci půjčit. . .“ nie je podobná skoro s ničím. Môže to byť spôsobené lomítkom v názve. „limity“ sú podobné s „age limit“ a „top up limits“, kde sa obsahovo najpodobnejšia zdá práve trieda „top up limits“. „trvalé platby“ by mohli byť podobné s „transfer timing“, avšak toto ešte overíme. „čekající platby“ by mohli byť podobné s „pending transfer“.

asi iba 3 páry z top3 výsledkov vyššie sa javia ako dostatočne sémanticky podobné:

- „top up limit“ ku „limity“,
- „pending transfer“ ku „čekající platby“,
- „transfer timing“ ku „trvalé platby“.



Obr. C.1: Najpodobnejšie názvy tried z datasetu banking77 ku názvom tried z datasetu text2bank (pomocou kosínovej podobnosti LASER embeddingov). Top 3 najpodobnejšie sú vyznačené, ostatné sú pre prehľadnosť nahradené 0.

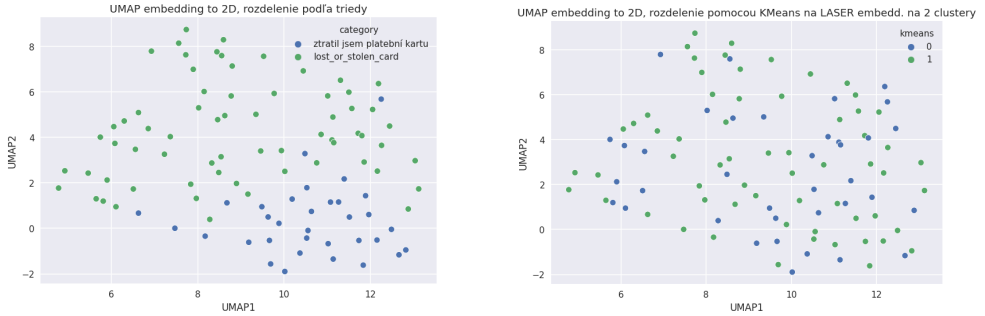
Bolo prevedené manuálne („pozriem a vidím“) párovanie EN ku CZ triedam a bol objavený ďalší možný 1 pár: „Lost or stolen card“ ku „ztratil jsem platební kartu“

V nasledujúcich 2 sekciách prevedieme rozbor tried, ktoré sme vybrali na zlúčenie. Ostatných kandidátov vynechávame, sú v priloženej analýze.

C.2 Rozbor tried „Lost or stolen card“ z (B77) a „ztratil jsem platební kartu“ (t2b)

Pre presnosť použime originálne znenia intentov z B77 datasetu a použime LASER s anglickým jazykom, intenty z t2b embedujme v českom jazyku. Chceme tak overiť či sú tieto triedy zlúčiteľné dohromady.

Najpodobnejšie vzorky „Ukradli mi kartu!“ z t2b13 „Someone stole my card!“ z b77 majú vysokú kos. podobnosť 0,836. Najmenej podobné sú vzorky „co když nevím najít kartu“, „I cant find my card.“ s hodnotou podobnosti 0.615 – čo je stále pomerne vysoká hodnota a očividne sú tieto dve vzorky



(a) LASER embd. farba podľa triedy (b) LASER embd. farba podľa zhluku

Obr. C.2: LASER embeddingy a z nich KMeans zhluky tried „Lost or stolen card“ z (B77) a „ztratil jsem platební kartu“ z (t2b)

veľmi podobné.

C.2.1 Zhlukovanie intentov z oboch tried

Na obr. C.2 vidíme vizualizáciu embeddingov intentov z oboch tried v 2 dimenziách. KMeans hľadalo zhluky v LASER embeddingoch, ale zobrazujeme pomocou UMAPu. Na C.2a je farebné odlíšenie bodov v 2D UMAP priestore podľa prislúchajúcej triedy. Vidíme tu, že sa napriek tomu, že body oboch tried sa príliš neprekrývajú – UMAP nebol schopný nájsť rozličnosť v embeddingoch. To, že sú body rovnakej triedy blízko seba a tvoria pomyselný zhluk mohlo byť spôsobené práve rozličnými jazykmi

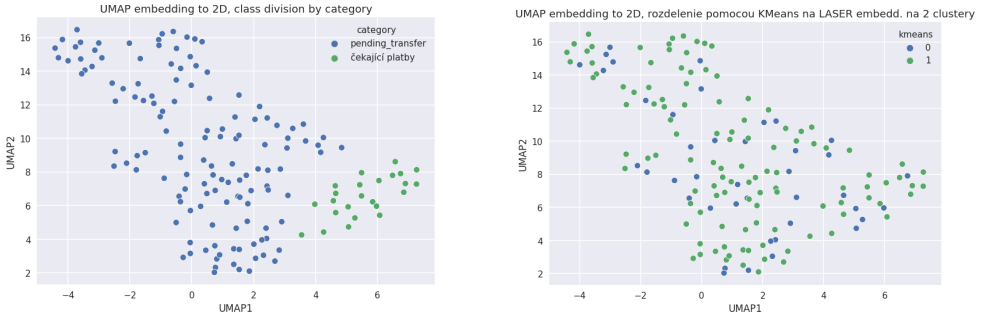
Použili sme KMeans algoritmus na nájdenie klastrov (v tomto prípade by sme očakávali práve 2 klastre) Na obr. C.2b je viditeľné, že ak **v priestore laser embeddingov** existujú aspoň 2 klastre, sú určite rozličné od originál delenia podľa triedy

KMeans klastrovanie je pomerne chaotické, to značí, že tieto dve triedy, zdá sa, majú spoločný prekryv. Je to viditeľné hlavne na UMAP vizualizácii na obr. C.2a. Potvrdzuje to aj vizualizácia KMeans klastrov na UMAPe. Tieto dve triedy zlučujeme do jednej.

C.3 Rozbor tried „pending transfer“ (B77) a „čekající platby“ (t2b)

Pre presnosť znovu použime originálne znenia intentov z B77 datasetu a použime LASER s anglickým jazykom.

Najpodobnejšie vzorky „mám nějaké naplánované platby?“ z t2b13 „Is my transfer pending?“ z b77 majú pomerne vysokú kos. podobnosť 0,74, a významovo sú si celkom blízke. Najmenej podobné sú vzorky „rezervace“, „I



(a) LASER embd. farba podľa triedy (b) LASER embd. farba podľa zhluku

Obr. C.3: LASER embeddingy a z nich KMeans zhluky tried „pending transfer“ (B77) a „čekající platby“ (t2b)

have a pending transaction“ s hodnotou podobnosti 0,51 – čo je veľmi nízka podobnosť, a sémanticky nesúvisia.

C.3.1 Zhlukovanie intentov z oboch tried

Na obr. C.3 vidíme vizualizáciu embeddingov intentov z oboch tried v 2 dimenziách (pomocou UMAPu) a KMeans klastrovanie LASER embeddingov.

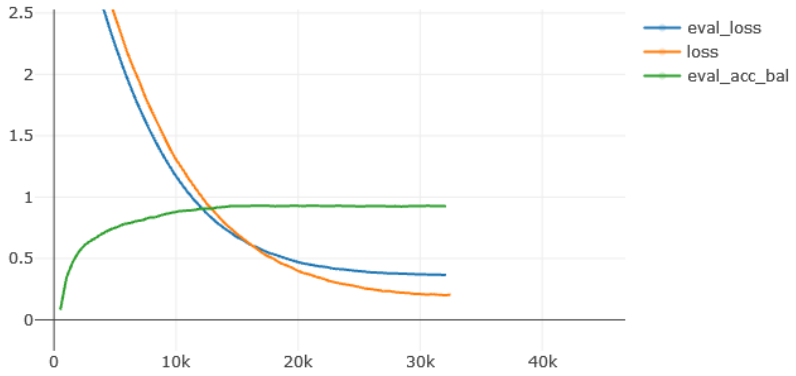
UMAP vizuálne vyseparoval rozdiely medzi embeddingami oboch tried, avšak zdá sa, že podtried by tu mohlo byť viac (cca, 4-6) Bolo prevedené KMeans klustrovanie nad LASER embeddingami. Hľadané boli 2 rozpady. Kmeans si nebol schopný poradiť so separovaním žiadnej triedy. Ak by sme zvýšili počet rozkladov na 4, rozdelenie by bolo podobne chaotické. Podľa všetkého je „zmätený“ – to by mohlo signalizovať zlučiteľnosť oboch tried (do istej miery).

Zdá sa, že tieto dve triedy by mohli byť zlučiteľné podľa KMeans. UMAP však ukázal, že tento priestor embeddingov je nejakým spôsobom rozdeliteľný. Otázkou zostáva či to bolo spôsobené iným jazykom (a teda trochu inou reprezentáciou v embedding priestore) alebo naozaj distinktnosťou oboch tried.

C.4 Záver Analýzy

Ukázali sme podobnosť medzi embeddingami názvov tried t2b a b77. Iba 3 páry z top3 výsledkov vyššie sa javili ako dostatočne sémanticky podobné. Pri pokuse o zlúčenie vyššie spomenutých tried z UMAP analýz vysvitlo, že zlučiteľné triedy by boli iba

- „Lost or stolen card“ a „ztratil jsem platební kartu“ a
- „pending_transfer“ a „čekající platby“.



Obr. C.4: Trénovacia (oranžová), validačná (modrá) krivka modelu RobeCzech a Balancovaná presnosť (zelená). Os x sú jednotlivé kroky tréningu a os y hodnoty stratovej funkcie a presnosti

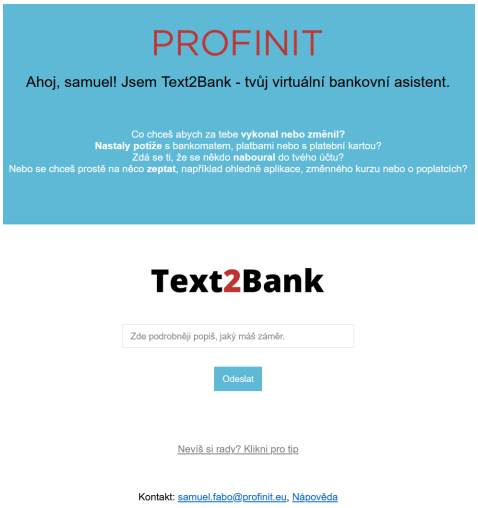
Manuálne sme prešli všetky intenty zo sady BANKING77 a ponechali iba intenty relevantné pre doménu českého bankovníctva. Tzv. top-up (navýšenie peňazí na účte – „dobitie kreditu“) intenty boli teda vyhodené. Triedy, ktoré sme ponechali sú v priloženom súbore. Zo sady TEXT2BANK13 sme vyhodili triedu „limity“, pretože mala veľmi nízku sémantickú granularitu.

C.5 Tréning modelu RobeCzech

Trénovaciu a validačnú krivku môžeme vidieť na obr. C.4. Čas tréningu bol 50min a prebiehal na dvoch GPU kartách `GeForce␣RTX␣208`.

C.6 Ukážka klasifikácie modelu RobeCzech

Na obrázku C.5a môžeme vidieť úvodnú obrazovku tejto demonštračnej aplikácie po zadaní mena/emailu. Na obrázku C.5b môžeme vidieť funkčnosť modulu neistoty v modeli



(a) Úvodná stránka finálnej aplikácie Text2Bank

(b) Úspešné rozpoznanie neznámeho zámeru aplikáciou Text2bank

Dodatok D

Zoznam použitých skratiek

NLP Natural Language Processing

NB Naive Bayes

MLE Maximal Likelihood Estimation

MAP Maximum a Posteriori

gNB Gaussian Naive Bayes

CBOW Continuous Bag Of Words

RNN Recurrent Neural Network

LSTM Long Short Term Memory

BiLSTM Obojsmerná LSTM

MLM Masked Language Modeling

NSP Next Sentence Prediction

Dodatok **E**

Obsah priloženého CD

- `README.md` ..... stručný popis obsahu CD
- `text2bank_app` ..... adresár so spustiteľnou aplikáciou Text2Bank
- `TEXT`
  - `DP-fabo-utf8.tex` ..... zdrojová forma práce vo formáte LaTeX
  - `DP-fabo-utf8.pdf` ..... text práce vo formáte PDF
- `intent_detector` ..... python súbory pre tréning
  - `clf_robeczech.py` ..... tréning výsledného modelu
  - `clf_*.py` ..... tréning ostatných modelov
- `data_text2bank` ..... všetky použité dáta
  - `merged-2022-03-24.csv` ..... výsledný zlúčený dataset