VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÝ PŘEPIS ŘEČI S PODPOROU CODE SWITCHING

AUTOMATIC TRANSCRIPTION OF SPEECH SUPPORTING CODE SWITCHING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ŠTĚPÁN BÍLEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. IGOR SZŐKE, Ph.D.**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

156791

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Bílek Štěpán**
Program: Informační technologie
Název: **Automatický přepis řeči s podporou code switching**
Kategorie: Zpracování signálů
Akademický rok: 2023/24

Zadání:

1. Seznamte se s automatickým rozpoznáváním řeči a nástroji jako jsou ESPnet2, Whisper. Seznamte se s databází CommonVoice. Seznamte se s přístupy pro rozpoznávání řeči s častou změnou jazyka (code switching).
2. Adaptujte Whisper na českých, anglických, vietnamských a dalších jazycích s podporou code switching (změna jazyka uprostřed promluvy). Vyhodnoťte dosažené výsledky.
3. Seznamte se s dodaným rozpoznávačem řeči v ESPnet2. Replikujte výsledky z předchozího bodu. Ověřte úspěšnost code switching až na úrovni slov.
4. Průběžně vyhodnocujte úspěšnost, jak na standardních, tak na dodaných datasetech.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Shinji Watanabe, et al., "ESPnet: End-to-End Speech Processing Toolkit", Proceedings of Interspeech, 2018
- Xinyuan Zhou, et al., "Multi-Encoder-Decoder Transformer for Code-Switching Speech Recognition", 2020
- Ka Long Roy Chan, "Trilingual Code-switching in Hong Kong", Applied Linguistics Research Journal, 2019
- Dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:
Body 1, 2 a část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Szőke Igor, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 9.11.2023

Abstrakt

Tato práce se zabývá problematikou automatického rozpoznávání řeči. Zaměřuje se na rozpoznávání audia obsahující vícejazyčné promluvy, tzv. code-switching. Problém nedostatku vícejazyčných dat pro trénování je řešen kombinováním nahrávek v angličtině a němčině dohromady. Pro co největší přiblížení ke skutečné dvojjazyčné řeči je část datasetů tvořena spojováním nahrávek podobných mluvčích. Na vytvořených datech je trénován a testován model Whisper. Ten v původní neadaptované verzi dosahuje chybovosti až 70 %. Nejlepší modely trénované na kombinovaných datasetech dosahují chybovosti jen lehce přes 7 %. Výsledky této práce ukazují způsoby jak modely trénovat, aby dosahovaly co nejlepších výsledků.

Abstract

This thesis addresses the issue of automatic speech recognition, focusing on the recognition of audio containing multilingual speech, known as code-switching. The problem of a lack of multilingual data for training is addressed by combining recordings in English and German. To achieve the closest approximation to real bilingual speech, a portion of the datasets is created by merging recordings of similar speakers. The Whisper model is trained and tested on the created data. In its original unadapted version, the model achieves an error rate of up to 70 %. The best models trained on combined datasets achieve error rates slightly above 7 %. The results of this study demonstrate methods for training models to achieve the best possible performance.

Klíčová slova

Automatické rozpoznávání řeči, strojové učení, Whisper, code switching, fine-tuning, vícejazyčná řeč

Keywords

Automatic speech recognition, machine learning, Whisper, code switching, fine-tuning, multilingual speech.

Citace

BÍLEK, Štěpán. *Automatický přepis řeči s podporou code switching*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szőke, Ph.D.

Automatický přepis řeči s podporou code switching

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szőkeho Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Štěpán Bílek
9. května 2024

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Igorovi Szökemu Ph.D. za odborné rady a vedení této bakalářské práce.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V dnešní době, kdy se technologie rychle rozvíjí, se stává automatické rozpoznávání řeči (ASR) běžnou součástí našich životů. Běžně se setkáváme s hlasovými asistenty - v mobilních telefonech, hodinkách, automobilech. Videoserver YouTube umí k videím automaticky generovat titulky. Platformy pro videokonfernce umí vytvářet textové záznamy. Jednou z oblastí ASR je rozpoznávání řeči obsahující code-switching, tedy střídání jazyka v rámci jedné promluvy [14].

K tomuto může docházet při různých příležitostech. Výrazy z angličtiny pronikají do jiných jazyků. Při cestování do zahraničí můžeme mluvit stále svojí řečí, ale třeba vlastní jména nebo místní názvy vyslovujeme v řeči jiné. Toto jsou jen některé z případů, kdy je vhodné, aby si rozpoznavače řeči s code-switchingem poradily.

Hlavní překážkou pro trénování modelů pro rozpoznávání řeči obsahující střídání jazyka, je nedostatek dostupných dat, na kterých by mohly být rozpoznávací modely trénovány. Tato práce se zabývá tvořením datasetů, které obsahují nahrávky kombinované řeči. Využívá k tomu data z datasetu Mozilla Common Voice [2].

Datasety jsou tvořeny kombinací nahrávek v němčině a angličtině. Dále jsou rozděleny na datasty obsahující nahrávky vytvořené kombinací podobných a datasety vytvořené kombinací různých řečníků. Na těchto datech probíhá jak trénování, tak výsledné testování modelů. Práce má za úkol zjistit, jaký vliv má podobnost mluvčího na trénovacích datech na konečné výsledky rozpoznávacího modelu.

Modelem který je na takto vytvořených datech trénován na rozpoznávání code-switchingu je Whisper od společnosti OpenAI [18]. Jedná se o velký předtrénovaný model, který dosahuje velmi dobrých výsledku i při ladění na poměrně malém množství dat.

Whisper je trénován na různých datasetech, od běžných jednojazyčných, po datasety obsahující nahrávky vzniklé spojením více nahrávek v různých jazycích. Výsledné natrénované modely jsou poté testovány na různých testovacích datasetech. Opět jak jednojazyčných, tak kombinovaných. Výsledky jsou zaznamenávány a porovnávany, za účelem natrénování co nejpřesnějšího modelu.

Kapitola 2

Strojové učení a rozpoznávání řeči

Strojové učení je disciplína umělé inteligence, která se zabývá vytvářením algoritmů a technik pro analýzu dat a jejich schopnost se z těchto dat učit. Jeho cílem je umožnit počítačům adaptovat se na nové situace a provádět úkoly na základě datových vstupů. Strojové učení je založeno na vytváření programů tak, aby se mohly počítače učit z dat, místo aby byly pevně předem naprogramovány na specifické úkoly. To umožňuje počítačům rozpoznávat vzory, formulovat pravidla a provádět rozhodnutí na základě dat, nikoliv pracovat na základě předem daných konktrétnách instrukcí .[7]

2.1 Úvod do automatického rozpoznávání řeči

Automatické rozpoznávání řeči (ASR - z anglického Automatic Speech Recognition) představuje diciplínu v oblasti strojového učení, která se zabývá převedením mluveného slova na psaný text. ASR je aktivní oblastí výykumu více než padesát let. Řeč byla vždy považována za důležitý prvek pro lepší komunikaci mezi lidmi a stroji. Nicméně v minulosti se nikdy nestala skutečně užívaným prostředkem k dorozumívání se člověka a počítače.

V posledních letech se už ale setkáváme s tím, že některá zařízení ovládáme hlasem poměrně běžně. Tento trend byl umožněn z několika důvodů. Jedním z nich je Moorův zákon. Výpočetní sílá dostupná v dnešních zařízeních je o několik řádů vyšší, než před pouhými deseti lety. To umožňuje trénování složitějších modelů. Dalším z důvodů náhlého pokroku v oblasti ASR je přístup k datům. Díky neustálému navyšovaní možností internetu můžeme postavit modely na velkých datech získaných z reálných scénářů užívání. Třetím důvodem je obliba inteligentích zařízení, jako jsou mobilní telefony, asistenti v automobilech a další, kde je ovládání hlasem užitečnější a vhodnější, než jako náhrada klávesnice a myši u počítačů.[5]

Vstupem do rozponávače řeči je zvuk. Ten se zpracuje do podoby spektrogramu, tedy grafu, který znázorňuje jak se různé frekvence zvuku mění v čase. Tím vlastně vzniká „obrázek“, tedy matice čísel, se kterou už dokáže počítač pohodlně pracovat. Rozpoznávač z tohoto spektrogramu umí na základě tréninku a učení se na jiných spektrogramech vygenerovat přepis mluvené řeči.

2.2 Historie rozpoznávání řeči

Prvním softwarem, který by se dalo považovat za rozpoznávač řeči byl program Audrey, ten dokázal rozeznávat číslice, vyslovené jedním hlasem. Tento software je založen na loka-

lizování formantů (oblasti lokálního maxima jednotlivých tónů) ve výkonovém spektru.[16] V roce 1971 DARPA začala financovat výzkum na porozumění řeči, o kterém si mysleli, že je klíčem k automatickému rozpoznávání řeči. Tato myšlenka se však později ukázala jako mylná. [16] V roce 1976 se konala první konference ICASSP (International Conference of Acoustics, Speech and Signal Procesing - Mezinárodní konference o zpracování zvuku, řeči a signálů).[17] Téhož roku byl na univerzitě Carnegie Mellon vyvynut systém Harpy, který byl schopen rozeznat 1000 slov [12]

V polovině osmdesátých let výykumný tým IBM (vedený Čechoameričanem Frederikem Jelinkem) vyvinul hlasem aktivovaný psací stroj, který uměl poznat 20 000 slov. Dřívější způsoby rozpoznávání se soustředily na napodobení způsobu, jakým řeč zpracovává lidský mozek, Jelinek s jeho týmem však zvolili postup více se zakládající na statistickém modelování a použití HMM (Hidden Markov Model). Lingvisté se v té době domnívali, že HMM není k rozpoznávání řeči vhodný, protože vstupy příliš zjednodušuje, na to aby dokázal zohlednit všechny složité části jazyka. Přesto se HMM stal nejpouživánějším přístupem k ASR až do konce osmdesátých let.[16]

V osmdesátých letech byl představen také n-gram jazykový model. Ten je založený na shlukování slov do n-tic, pracuje na základě pravděpodobnosti, že se určitá posloupnost slov vyskytne v textu. V roce 1987 vznikl Katzův back-off model, který umožnil použití n-gramů s různými délkami. Společnost CSELT využívala skryté Markovovy modely k rozpoznávání jazyků.

Rychlý pokrok umožňovalo hlavně rychlé zvyšování výpočetního výkonu tehdejších počítačů. Na konci programu DARPA z roku 1976 měli výzkumnící nejvýkonější počítač s pamětí 4MB RAM, kterému trvalo rozpoznání řeči dlouhé 30 sekund zhruba 100 minut[17].

Jedním z prvních případů, kdy se začalo automatické rozpoznávání využívat v praxi byl produkt Dragon Dictate. Společnost AT&T nasadila v roce 1992 službu zpracování hovorů s rozpoznáváním hlasu, která sloužila k přepojování telefonních hovorů bez použití lidského operátora.

V začátcích nultých let bylo rozpoznávání řeči stále ovlivňováno tradičními přístupy, jako jsou skryté Markovovy modely kombinované s dopřednými (feedforward) umělými neuronovými sítěmi.[19]

2.3 Code-switching

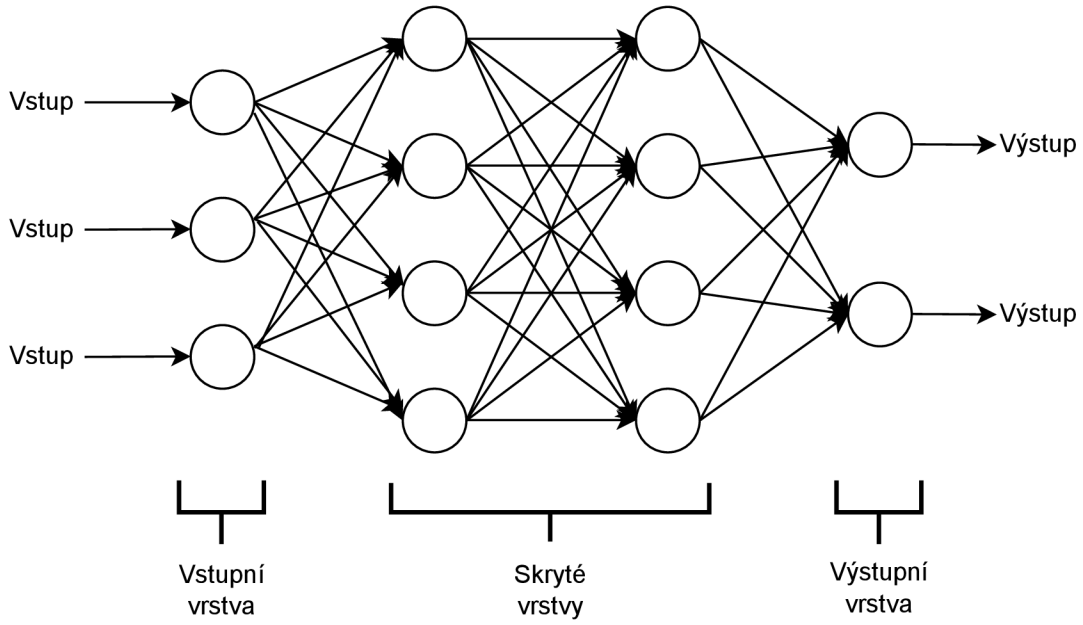
Code switching, neboli přepínání jazyků, představuje výzvu pro automatické rozpoznávání řeči, kdy mluvčí často kombinují více jazyků v jediné promluvě. Tento fenomén může být zvláště běžný v mezikulturních komunitách a multilinguálních prostředích. Při vývoji systému automatického rozpoznávání řeči (ASR) je tedy vhodné zahrnout mechanismy schopné efektivně zpracovávat code switching, aby bylo rozpoznávání řeči fungující a účinné i v takových podmínkách. Běžným případem code switchingu, kterým se denně setkáváme, může být třeba použití nějakého anglického slova v běžné české mluvě. Například „Dej tu fotku na story.“ Může se jednat také o použití místního názvu z jednoho jazyka v jazyce jiném. Napříkald „Dále se vydáme na Baker Street.“, nebo „Keep walking until you reach Zelný trh.“ [14]

2.4 Neuronové sítě

Umělá neuronová síť je matematický model inspirovaný biologickým nervovým systémem, který se skládá z několika vrstev umělých neuronů, spojených vahami.[7] Každý neuron příjmá vstupy, ty upráví podle vah (vynásobí je váhovými koeficienty) a poté na ně aplikuje aktivační funkci. Tento výstup se stane vstupem neuronů na další vrstvě neuronových sítí. Tyto vrstvy lze rozdělit do tří kategorií:

- **Vstupní** vrstva příjimá vstupy do sítě
- **Skryté** vrstvy jsou mezi vstupní a výstupní vrstvou, jejich neurony aplikují na své vstupy váhové koeficienty a aktivační funkci a posílají výstupy na další vrstvy
- **Výstupní** vrstva generuje výstupy sítě na základě vstupů z předešlé skryté vrstvy

Učení neuronové sítě probíhá tak, že jsou v průběhu trénování aktualizovány jednotlivé váhy pomocí některého z učících algoritmů, například algoritmu zpětného šíření chyby (backpropagation). Klasická neuronová síť bývá také nazývána jako feedforward neuronová síť, protože všechny informace proudí jen jedním směrem V případě, že má neuronová síť větší počet skrých vrstev, nazýváme ji hlubokou neuronovou sítí (deep neural network). Schéma této sítě lze vidět na obr. 2.1.

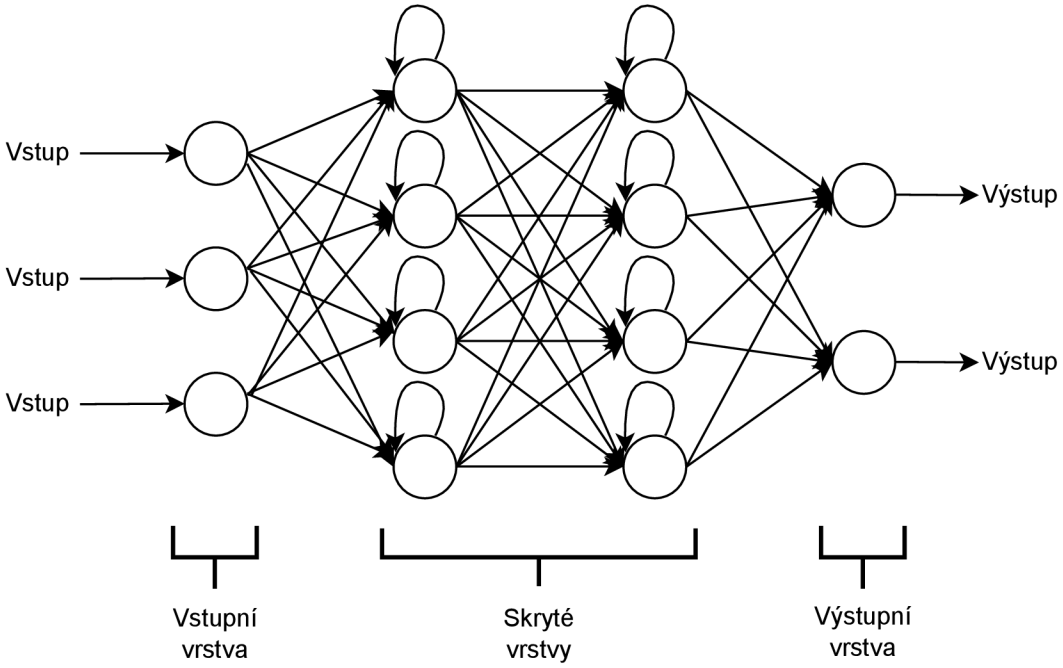


Obrázek 2.1: Neuronová síť

2.4.1 Rekurentní neuronové sítě

Rekurentní neuronové sítě (RNN - z anglického reccurent neural network) jsou typem neuronových sítí, které mají schopnost pracovat se sekvencemi dat. Na rozdíl od běžných feedforward sítí se totiž umí „vracet“. Jsou složeny z neuronů, které mají propojení sami na sebe, což jim umožňuje uchovávat stav nebo paměť v průběhu času. Tato funkcionalita je

vhodná pro zpracování sekvencí, jako jsou texty, časové řady nebo řeč. RNN zpracovávají vstup po jednotlivých krocích a aktualizují svůj vnitřní stav pomocí vstupních dat a svého předchozího stavu. Tento proces jim umožňuje učit se vztahy a vzory v datech závislé na čase. [7] Schéma rekurentní neuronové sítě zobrazuje obr. 2.2.



Obrázek 2.2: Rekurentní neuronová síť

2.4.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN - z anglického convolutional neural network) jsou typem neuronových sítí, které jsou efektivní při zpracování vizuálních dat, jako jsou obrázky (v případě rozpoznávání řeči spektrogramy), ale také se používají v oblastech jako je zpracování přirozeného jazyka (NLP). Jejich architektura je založena na konvolučních vrstvách, které aplikují filtry na vstupní data a tím extrahují lokální vzory. Ty se nazývají convolved feature map (mapa příznaků). Místo toho, aby byly neurony propojeny s každým neuronem ze sousedních vrstev, jsou propojeny pouze s blízkými neurony, navíc s každým stejnou vahou, díky tomu lépe zpracovávají „prostorová“ data - neuronová síť se chová jako oko, které se nesoustředí na celý obrázek (případně text) najednou, ale pouze na nějakou dílčí část. Mimo konvoluční vrstvu obsahují sítě ještě pooling vrstvu, ta podvzorkuje, nebo zmenší velikost vzorků nějaké konkrétní feature map (výstupu konvoluční vrstvy). Toto zrychluje zpracovávání a snižuje množství parametrů, které by musela síť zpracovat. [7]

2.4.3 Enkodér a dekodér

Enkodér a dekodér jsou součástí modelů, které pracují se sekvencemi dat, například s řečí nebo textem. Tyto bloky jsou základem architektury transformer. Enkodér je neuronová síť, která kóduje vstupní sekvencim, tedy mění vstupní sekevenci na vektor příznaků. Tento vektor příznaků je vstup dekodéru, který ho přemění na výslednou sekvenci. Jednodychým

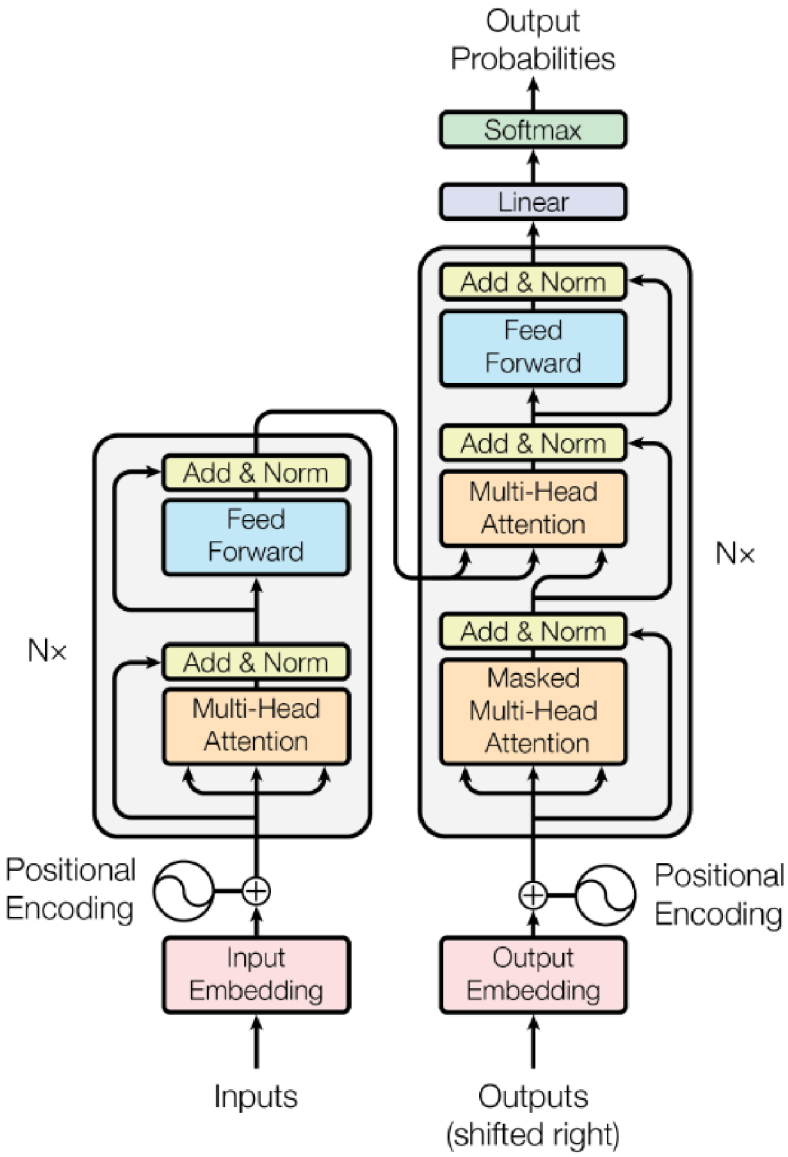
příkladem této funkcionality je například strojový překlad z jednoho jazyka do druhého. Enkodér zakóduje sekvenci v jazyce A, dekodér ji dekóduje do odpovídající sekvence v jazyce B. [20]

2.4.4 Attention

Attention v neurovýpočtových sítích je mechanismus, který umožňuje síti selektivně se zaměřit na určité části vstupních dat během zpracování informací. Tento mechanismus se inspiruje lidskou schopností soustředění se na určité prvky a omezení vnímaní jiných v okolí. V praxi to u neuronových sítí znamená možnost dynamicky přidělovat váhy různým částem vstupních dat během výpočtu. [20]

2.5 Transformer

Neuronová síť s architekturou transformer, poprvé představená v článku „Attention is all you need“[20], se od jiných neuronových sítí odlišuje zejména díky použití tzv. attention vrstev. Ty umožňují síti při výpočtu výstupní sekvence brát v úvahu libovolnou část vstupní i výstupní sekvence. Tato architektura také využívá princip zřetězených enkodérů a dekodérů a pozičního kódování. Schéme této architektury je vyobrazeno na obr. 2.3. Díky novým vlastnostem, které sítě typu transformer přinášejí, je možné dosáhnout zcela nových výsledků v různých oblastech výzkumu, běžně však pro zpracování textu, obrazu či řeči.



Obrázek 2.3: Architektura modelu transformer [20]

2.6 Úlohy pro strojové učení

Programy můžeme naučit mnoha typům úkolů, mezi nejčastější úlohy pro strojové učení patří [7]:

- **Klasifikace**, anglicky classification, je druh úlohy, kdy má počítačový program přiřadit vstup do nějaké kategorie
- **Regrese**, je úkol, kdy má počítačoví program předpovědět pro číselný vstup nějakou hodnotu. Obecně se to dá popsat tak, že zaneseme-li do grafu kazždý vstup na osu x a požadovaný výstup na osu y, algoritmus se snaží najít funkci, která prochází co nejblíže každému z bodů. Díky tomu pak může predikovat výsledky pro vstupy, na kterých netrénoval

- **Transkripce** Jedná se o případ, kdy má program na vstupu data v nějakém relativně nestrukturovaném formátu a přepisuje je do konkrétního textového formátu (např. UTF-8). Transkripce je využívána například při optickém rozpoznávání znaků (OCR), nebo řeči

2.7 Druhy učících algoritmů

Podle způsobu, kterým probíhá učení, můžeme učící algoritmy rozdělit do následujících kategorií [7]:

- **Supervised learning (Učení pod dohledem)** Tento typ algoritmů má k dispozici dataset obsahující jak vstupy, tak požadované výstupy. Na těchto datech, označovaných jako trénovací, se učí jakým způsobem jsou vstupy mapovány na výstupy
- **Unsupervised learning (Učení bez dohledu)** Tento typ algoritmů naopak požadované výsledky při trénování nezná, trénovací data nejsou nijak označena, klasifikována či rozdělena do kategorií. Algoritmus v datech hledá vzory a vztahy, jejichž znalost poté využívá při clusteringu
- **Semi-supervised learning (Učení s polodohledem)** Jedná se o kombinace předchozích způsobů. Trénovací data obsahují jak nepopsaná data, jako pří tréninku bez dohledu, tak data, která obsahují ke vstupům i požadované výstupy

2.8 Dataset

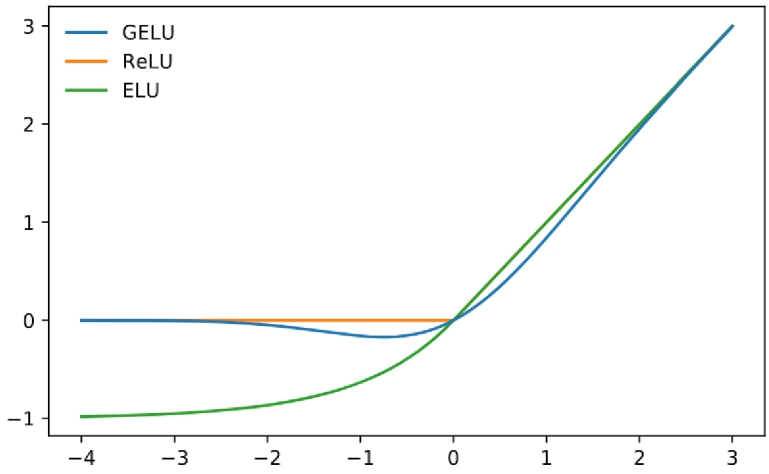
Dataset je soubor dat využívaných pro trénování modelu. Podle druhu modelu a úkolu obsahuje data potřebná pro trénování. Podle druhu modelu a učícího algoritmu obsahuje mimo vstupů i požadované výstupy nebo metadata. Dataset se většinou dělí do několika podmnožin (subset):

- **Train** obsahuje data na kterých se model učí
- **Validation** obsahuje data, na kterých se optimalizuje výběr hyperparametrů a může proběhnout vyhodnocení metrik
- **Test** obsahuje data pro testování výsledků modelu. Na těchto datech je natrénovaný model testovaný, aby mohly být výsledky vyhodnoceny. Data v této podmnožině by nejen neměla být stejná jako data trénovací a validační, ale neměla by mít ani společné důležité vlastnosti. Například pokud by trénovací data pro ASR obsahovala nahrávky od stejného řečníka jako data trénovací, výsledky by byly zkreslené

2.9 Aktivační funkce

Jedná se o funkci [7], která v průběhu učení neuronové sítě převadí vstup z předchozích vrstev (x), na nějaký výstup (y). Pro různé typy úloh se používají různé aktivační funkce, od jednoduchých jako je například funkce identity, kde y = x; funkce ReLU, která pro záporné vstupní hodnoty vrací nulu a pro kladné je stejná jako funkce identy, tedy vrací stejný výsledek, jako byl její vstup. Po složitější, jako je ELU (Exponential Linear Unit) [3], která pro hodnoty x větší, než nula opět vrací stejnou hodnotu, ale pro hodnoty menší, než

nula vrací zápornou hodnotu odvozenou od exponenciály s mocninou x. V modelu Whipser zvoleném pro tuto práci je využívána funkce GELU (Gaussian Error Linear Unit)[9], která tyto dvě funkce (ReLU a ELU) kombinuje. Funcke zobrazuje graf na obr. 2.4.



Obrázek 2.4: Graf zobrazující aktivační funkce ReLU, ELU a GELU [9]

2.10 Hyperparametry

Hyperparametry jsou nastavitelné parametry, které ovlivňují proces učení. Pro úspěšné natrénování modelu, je nutné při tréninku tyto parametry nastavit co nejvhodněji.

- **Learning rate** (rychlost učení) definuje velikost jednoho kroku učícího algoritmu. Tím prakticky určuje rychlost učení
- **Počet kroků** určuje velikost kroku učícího algoritmu
- **Batch size** (velikost dávky) počet tréninkových příkladů použitých při aktualizaci vah modelu během jedné iterace tréninku
- **Epocha** reprezentuje jeden průchod algoritmu trénovací datovou sadou

2.11 Předtrénované modely

Jedná se o modely strojového učení, které byly natrénovány na nějakém zpravidla velkém množství dat, na úlohách, které jsou stejné, nebo podobné těm, na kterých bude výsledný model použit. Trénink těchto modelů zpravidla vyžaduje velký výpočetní výkon a velké trénovací datasety. Používají se jako výchozí body pro další učení a jsou připraveny k jemnému ladění (fine-tuning) na konkrétní úlohy. Různé modely mohou mít těchto předtrénovaných bodů (checkpointů) více a můžou sloužit k různým účelům. Například menší pro použití v jednoduchých aplikacích, kde je důležitá rychlost a nenáročnost na výpočetní výkon, nebo větší model pro dosáhnutí co nejlepších výsledků.[8]

2.12 Overfitting a underfitting

Overfitting a underfitting jsou běžné problémy při trénování modelu. Overfitting nastává, když model příliš přesně odpovídá trénovacím datům a není schopen se dobře generalizovat na nová data. Tento jev je často způsoben příliš složitým modelem nebo nedostatečným množstvím trénovacích dat. Naopak underfitting nastává, když model není dostatečně schopen zachytit strukturu dat a má špatnou výkonnost jak na trénovacích, tak na nových datech. To může být způsobeno příliš jednoduchým modelem nebo nedostatečným trénováním. Mezi důsledky overfittingu patří nízká přesnost na nových datech a nežádoucí citlivost na šum. Underfitting může vést k nedostatečným výsledkům a neschopnosti modelu naučit se na datech efektivně.[1]

2.13 Metrika hodnocení

Pro zhodnocení výsledků rozpoznávání řeči je třeba zvolit hodnotící metriku. Pro tuto práci byla zvolena běžně používáná metrika WER (Word Error Rate) [11]. Je odvozena od Levenshteinovy vzdálenosti, která měří vzdálenost dvou řetězců pomocí počtu změn znaků (náhrada, přídání, nebo odstranění). Tato metrika udává chybovost, je tedy usilováno o co nejnižší hodnotu, ideálně co nejblíže nule. Word Error Rate se počítá následovně:

2.13.1 Výpočet

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C} \tag{2.1}$$

kde

- S je počet náhrad
- D je počet odstranění
- I je počet vložení
- C je počet správných slov
- N je počet slov v referenčním přepisu

2.13.2 Normalizace

Problém pro tuto metriku představuje skutečnost, že za chybu považuje každý rozdíl mezi slovy z původního přepisu, a přepisu vygenerovaného modelem. Například malé a velké písmeno, nebo zápis čísel (1000; 1, 000 a tisíc se vyslovují naprosto stejně, ale zápis je jiný). Aby se zamezilo, zkreslení výsledků způsobené těmito rozdíly, jsou přepisy normalizovány. Speciální znaky, jako uvozovky, otazníky apod. jsou ignorovány, čísla jsou přepsány do stejného formátu, nehledí se na velikost písmen (case sensitivity). Slova která mají různé možnosti zápisu, ale vyslovují se stejně (například color a colour) jsou také sjednoceny. [15]

2.13.3 Alternativní metriky

Místo metriky WER, lze použít například CER (Character Error Rate). CER se počítá stejně jako WER, ale hodnotí se pouze jednotlivé znaky, nikoliv celá slova. Tato metrika je mírnější, a pro běžné užití méně vhodná. Existují ale i případy, kdy je CER vhodnější,

například pří hodnocení jazyků s úplně jiným systémem psaní, například japonštiny. Další alternativou může být SER (Sentence Error Rate), tato metrika je počítána z celých vět a může být použíta při vyhodnocování modelu používaného pro velmi dlouhé texty, kde je vyžadována velká přesnost.

Kapitola 3

Model Whisper

3.1 Úvod

OpenAI Whisper, vydaný v září roku 2022, představuje pokročilý model pro automatické rozpoznávání řeči, který využívá moderní přístup založený na hlubokém učení. Na rozdíl od svých předchůdců (Wav2Vec 2.0), kteří jsou trénovaní na nepřepsaných (unlabeled) datech, Whisper je předtrénován na 680 000 hodinách přepsaných i nepřepsaných zvukových dat. Toto množství je řádově vyšší, než množství nepřepsaných dat použitých k trénování modelu Wav2Vec 2.0 - ten byl trénován na 60 000 hodinách nepřepsaných dat. Navíc 117 000 hodin trénovacích dat je multilinguálních. I díky tomu umí Whisper rozpoznat řeč v 96 jazycích.

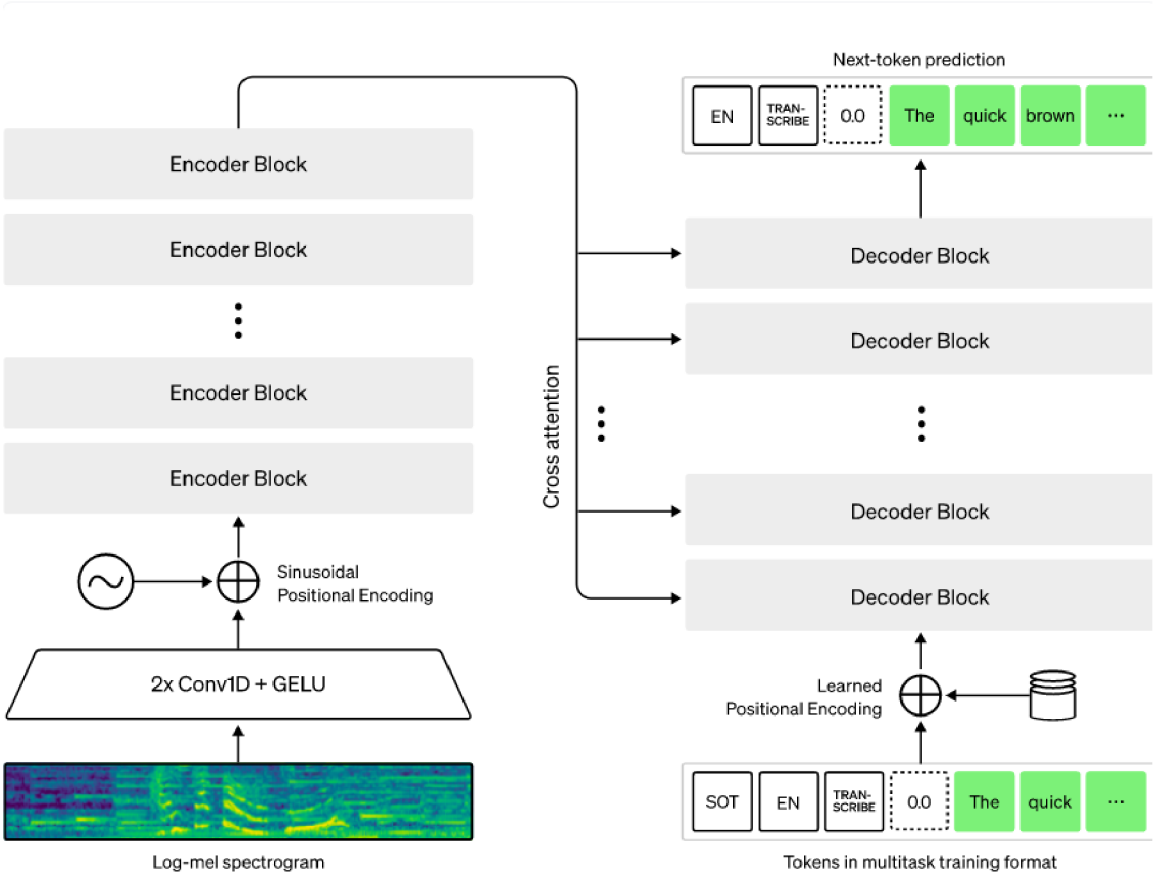
Díky takto velkému množství dat lze Whisper předtrénovat přímo na úkolu rozpoznávání řeči pod dohledem. Učí se mapováním řeči na text z označených dat skládajících se ze zvukové a přepisu. Díky tomu vyžaduje jen málo dodatečného ladění, aby dosáhl kvalitních výsledků při ASR. To je rozdíl od modelu Wav2Vec 2.0, který je předtrénován tak, aby se naučil zprostředkující mapování z řeči do skrytých stavů pouze z nepřepsaných zvykových datech. Zatímco toto předtrénování bez dohledu poskytuje vysoce kvalitní reprezentaci řeči, neučí se mapování řeči na text. Toto mapován se naučí až během jemného ladění (fine-tuningu), proto je pro podání výkonu konkurcenceschopného s modelem Whisper potřeba více dolaďování. Naopak Whisper je díky velikosti trénovací sady velmi robustní a pro realné využití už potřebuje velmi málo, případně žádné ladění.

Díky obrovskému množství trénovacích dat vykazují modely Whisper silnou schopnost zobecnění na mnoho datovách sad. Předtrénované body (checkpointy) dosahuji vynikajicích výsledků. Při testování na testovací sadě LibriSpeech chybovost (WER) dosahovala jen málo přes 3. [18]

3.2 Architektura

Jedná se o end-to-end model implementovaný jako Transformer typu enkodér/dekodér, také označovaný jako model sekvence-sekvence. Mapuje sekvenci zvukových spektorgramů na sekvenci textových tokenů.

Nejprve jsou surové zvukové stopy převedeny na logaritmický spektrogram, o což se stará extraktor příznaků (feature extractor). Enkodér poté zakóduje spektrogram a vytvoří posloupnost skrytých stavů kodéru. Nakonec dekodér předpovídá textové tokeny ovlivněné jak předchozími tokeny, tak skrytými stavy kodéru. Schéma této architektury můžeme vidět na obr. 3.1. [18] [6]



Obrázek 3.1: Schéma architektury modelu Whisper [18]

3.3 Předtrénování původního modelu Whipser

Na modelu Whisper je prováděno předtrénování a fine-tuning pomocí funkce cross enthropy [1]. Jedná se o standardní funkci pro trénování sequnce-to-sequence systémů na klasifikačních úlohách. V případě Whisperu je prováděn trénink na správnou klasifikaci cílového textového tokenu z předem definovaného slovníku textových tokenů.

Whisper má pět originálních konfigurací s různou velikostí modelu. Verze tiny, base, small, medium a large. Tyto modely jsou dostupné na GitHubu [15].

3.4 Vlastní trénovaní modelu Whipser

Pro trénování je nejdříve nutné mít připravený tréninkový dataset. V našem případě různé části datasetu Mozilla Common Voice.

Pipeline automatického rozpoznavače lze rozdělit do několika částí:

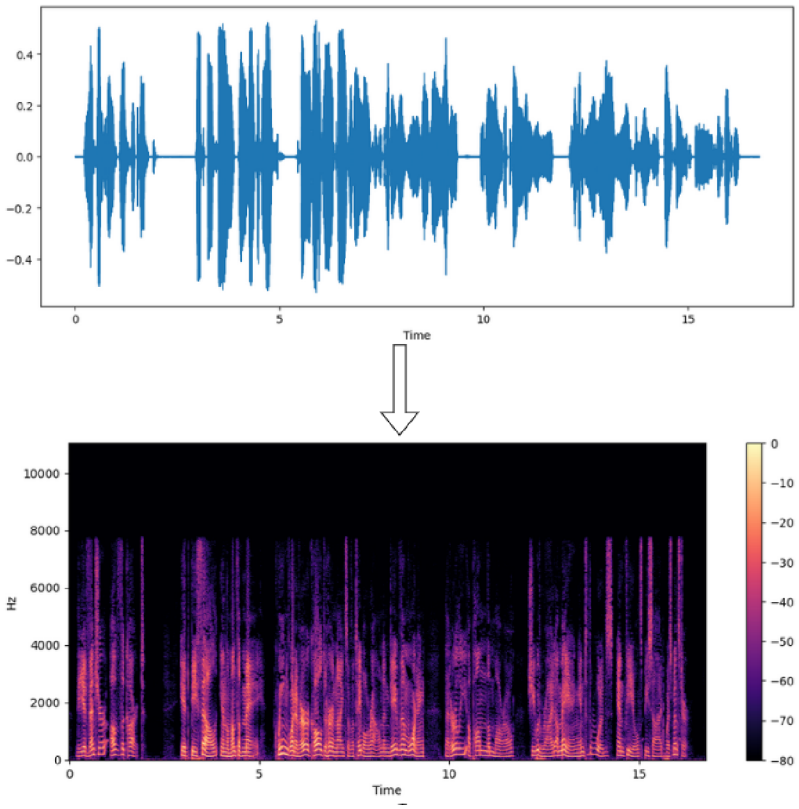
[1] Cross enthropy je míra rozdílu mezi dvěmi distribucemi pravděpodobnosti pro danou náhodnou veličinu nebo množinu událostí. V podstatě reaguje na pravděpodobnost události - méně očekavatelná událost má větší „váhu“, než událost očekávatelná[13].

3.4.1 Feature extractor

Pro správné fungování je nutné nejprve převést audio na správnou vzorkovací frekvenci [18]. V případě použití dat s nesprávným vzorkováním bude docházet k neočekávaným výsledkům a trénování bude praktické zbytečné. Extraktor příznaků modelu Whipser očekává zvukové vstupy o vzorkovací frekvenci 16 kHz, je tedy nutné převést frekvence vstupního audia na tuto hodnotu.

Feature extractor modelu Whisper plní dvě funkce. První z nich je doplňování, případně zkracování zvukových vzorků tak, aby měly všechny přesně 30 sekund. Vzorky které jsou kratší jsou doplněny nulami až do konce sekvence - nuly v audio signalů odpovádají žádnému signálu, tedy tichu. Vzorky delší, než 30 sekund jsou zkráceny na požadovanou délku a zbytek vzorku je zahozen. Díky tomuto mechanismu není nutné používat při předávání zvukových vstupů modelu masku pozornosti. Tato vlastnost je u rozpoznávacích modelů unikátní. U většiny ostatních modelů je nutno maskou pozornosti popisovat, kde byly sekvence doplněny a tím pádem ignorovány. Whisper je navrhnut tak, aby sám poznal, které vstupní signály může ignorovat.

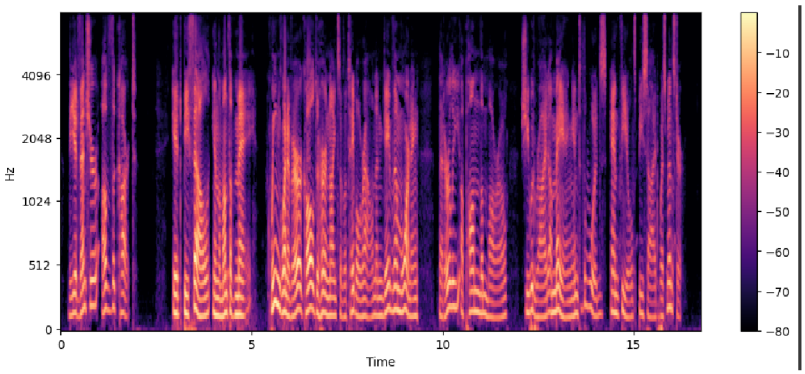
Druhou funkcí Feature extraktoru modelu Whisper je, jak znázorňují obr. 3.2 a 3.3, převod zvukových polí na logaritimické spektrogramy (tzv. Log-Mel Spektrogramy). Tyto spektrogramy jsou vizuální reprezentací frekvencí signálu. [6]



Obrázek 3.2: Převod vzorkovaného zvukového pole na spektrogram.

Logaritimické spektrogramy se pří úlohách na automatické rozpoznávání řeči používají proto, že člověk je různě citlivý na výšky tónu, zatímco obyčejný spektogram zobrazuje

frekvence v lineárních krocích, log-mel spektrogram je zobrazuje v logaritmických krocích. [22]



Obrázek 3.3: Log-Mel Spektrogram.

Jak můžeme vidět na obrázku Log-Mel spektrogram používá melovu škálu, která lépe odpovídá vnímání lidského sluchu. Což vede k většímu prostoru pro nižší frekvence a menšímu pro vyšší.

3.4.2 Tokenizer

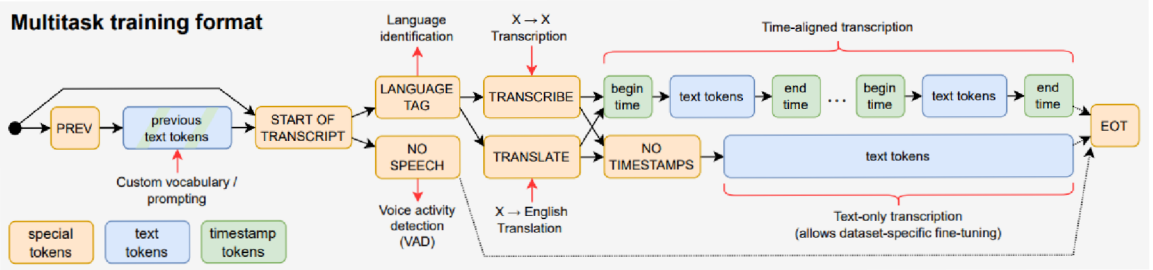
Tokenizer slouží k vypisování textových tokenů, které označují index předpovídaného textu mezi položkami slovníku. Tokenizer mapuje posloupnost textových tokenů na skutečný textový řetězec.

Tokenizer modelu Whisper je předem natrénován na devadesáti šesti jazycích. Díky tomu disponuje rozsáhlým byte-pairem[2] který je vhodný pro většinu multilinguálních aplikací automatického rozpoznávání řeči.

Začátek predikce je označen tokenem `<|startoftranscript|>`. Nejprve se předpoví mluvený jazyk, který je reprezentován jedinečným tokenem pro každý jazyk v trénovací sadě. V případě, že se v audio souboru řeč nevyskyuje, je model trénován k tomu, aby predikoval token `<|nospeech|>`, tedy poznal a zaznamenal, že se nejedná o řeč. Následuje token označující úlohu, kterou má model plnit, tedy transkripci - přepis, nebo translaci - překlad. K tomuto rozlišení používá buď token `<|transcribe|>`, nebo `<|transalte|>`. Následuje určení, zda se mají predikovat časové značky. Implicitně se predikují, pokud není žádoucí značky predikovat, použije se token `<|notimestamps|>`. Schéma můžeme vidět na obr. 3.4.

V tomto okamžiku jsou úloha a požadovaný formát plně specifikovány a začíná výstup. Nakonec je přidán `<|endoftranscript|>` token. [18]

[2]Byte-pair encoding je technika pro tokenizaci textu, která je založena na iterativním slévání nejčastěji se vyskytujících bytů (znaků) v textu dohromady [23]



Obrázek 3.4: Diagram sekvence tokenů [18]

Kapitola 4

Tvorba datasetu

Velkou překážkou pro trénování modelu pro code switching je nedostatek dostupných dat. Existuje jen málo dostupných datasetů, kde by mluvčí střídal jazyky. Pro tuto práci byly využity data z datasetů Mozzila Common Voice [2]. Ty obsahují obrovské množství hlasových dat se správným přepisem od dobrovolníků z celého světa. Díky tomu jsou nahrávky od různých autorů, v různých jazycích, nářečích, s různou kvalitou, rušením. Široké spektrum různých faktorů umožňuje trénováním na těchto datech vytvářet poměrně robustní modely.

Problémem však je, že jsou nahrávky vždy v právě jednom jazyce. Datasety pro finetuning jsou tvořeny spojováním jednotlivých nahrávek a přepisů dohromady, tímto způsobem se dají vícejazyčná data vytvořit poměrně snadno, vznikne ale probém s tím, že se mění mluvčí, což může výsledky značně ovlivnit. Pokud se nahrávka skládá ze dvou jazyků, přičemž se aspoň jeden v rámci nahrávky opakuje, není problém nahrávku tvořit tak, aby v rámci jedné nahrávky byl pro jeden jazyk ten samý řečník. V této práci je snaha i o to, aby byl co nejpodobnější řečník i v druhém jazyce. Toho lze docílit například aspoň tak, že se pohlídají dostupné parametry z datasetu - pohlaví a věk řečníka. Lepším řešením je zjištění podobnosti mluvčích.

4.1 Podobnost mluvčích

K vytvoření datasetu, ve kterém jsou různí mluvčí spojovaní do jedné nahrávky je využívána podobnost mluvčích. V této prácí byla využita knihovna pyannote [4], která umožňuje podle nahrávky vytvoří voice print řečníka. Jedná se o vektor, který popisuje vlastnosti mluvčího. Pro každou dvojici vektorů, je poté vypočítána cosinová vzdálenost, podle které jsou seřazeny a nahrávky jsou spojovany postupně od těch nejpodobnějších mluvčích. Úplně nejpodobnější řečníci jsou použiti na testovací dataset, zbylí, méně podobní jsou využiti na dataset trénovací.

4.2 Míchání dat podle jejich významu

Původně bylo zamýšleno vytváření trénovacích dat takovým způsobem, že budou v přepisech rozlišovány named entities - tedy slova různých kategorií. Například označení osob, lokací organizací a ve výsledných datech budou nahrazováný slovy ze stejné kategorie, akorát v jiném jazyce. O toto rozpoznávání se měla postarat pythonovská NLP knihovna spacy [10], ta fungovala poměrně dobře a skutečně vracela správně označené pojmenované

entity. Problém byl v tom, že Whisper, ani ve verzi timestamped nedokázal přesně časovými značkami označovat jednotlivá slova, což je zásadní pro správné tvoření nahrávek. Největší problémy měl s tím, když byla slova vyslovena krátce po sobě, nebo když bylo slovo těžko rozpoznatelné i bez požadavku na časovou značku. U takto spojených dat by u malé části položek neodpovídal přepis skutečnosti, což by velmi negativně ovlivnilo trénování a žádné testování na významově souhlasících vícejazyčných datech nakonec neproběhlo.

4.3 Vlastní dataset

Pro testování výsledných modelů byl využit dataset, skládající se z nahrávek vlastní řeči, obsahuje 100 nahrávek čtených anglicko-německých vět a jejich přepis. Tento dataset byl vytvořen za účelem zjištění chybovosti na co nejskutešnějších datech. I když jsou testovací datasety tvořeny z nahrávek common voice takovým způsobem, aby se co nejvíce přiblížily reálné vícejazyčné promluvě, je mezi nimi a skutečnou řeší znatelný rozdíl. Zatímco kombinované nahrávky z dat common voice obsahují vždy jedno přepnutí řeči do jiné a zpět, většina nahrávek ve namluveném datasetu těchto přepnutí obsahuje hned několik. Příklad: „I went gestern to the park, and there I met meine Freunde for a picnic.“

Kapitola 5

Použité technologie a postup implementace

Tvorba velkých datasetů je náročná na operační paměť a trénování modelů je velice náročné na paměť grafické karty, z toho důvodu byla většina práce provedena v prostředí Google Colab, to umožňuje pracovat v Jupyter notebooku online s využítím až s 51 GB RAM a 22.5 GB paměti grafické karty. Postup práce sestával z nastudování problematiky, tvorby testovacích a trénovacích datasetů, trénování předtrénovaného modelu a zhodnocení výsledků.

5.1 Hugging face transformers

Hugging face transformers [21] je prostředí pro strojové učení a práci s nástroji PyTorch, Tensorboard, JAX a dalších. Dále nabízí možnost pohodnlě přistupovat k datasetům a cloud pro ukládání vlastních datasetů a modelů - HuggingFace hub.

Poskytuje api a nástroje pro práci s předtrénovanými modely.

5.1.1 PyTorch

PyTorch je jedním ze základních kamenů nástroje Hugging Face. Jedná se o open source framework původně vyvinut společností Meta AI pro hluboké strojové učení. Využívá se pro snadné vytváření a trénování neuronových sítí.

5.1.2 Tensorboard

Tensorboard je nástroj od společnosti TensorFlow, slouží ke sběru a vizualizaci dat potřebnou při trénování. Umožňuje zaznamenávat hodnoty jako je například výsledek ztrátové funkce, nebo přesnost. Tyto a další hodnoty lze snadno vizualizovat do grafické podoby .

5.2 Tvorba datového rámce s voice printy

Pro vytvoření datasetu, kde jsou v jednotlivých nahrávkách spojeny nahrávky podobných mluvčích, bylo potřeba získat voice print každého z řečníků, jak z německého datasetu, tak z anglického. Nejprve byl vytvořen nový dataset, který obsahoval o každém řečníkovi jeho id, voice print a index jeho prvního výskytu v původním datasetu. Znalost tohoto indexu

je zásadní pro rychlé fungování filtrovací funkce pro vytvoření finálních testovacích a trénovacích datových sad. Poté byla spočítána cosinová vzdálenost mezi voice printem každého německého řečníka s voice printem každého anglického. Do výsledného datového rámce byla uložena každá dvojice řečníků, jejich id, index prvního výskytu v původním datasetu a jejich cosinová vzdálenost. Výsledný datagram byl nakonec seřazen od nejpodobnějších dvojic po nejméně podobné.

5.3 Tvorba datasetů

Pro tvorbu datasetů s podobnými mluvčími byl napsán skript, který na základě zmíněného datové rámce postupně spojoval nahrávky a přepisy od nejpodobnějších dvojic mluvčích. Pro rychlý přístup k nahrávkám z původních datasetů Common Voice byla použita funkce, která díky znalosti indexu prvního výskytu řečníka v datasetu vyfiltrovala datasety pouze na položky s dané dvojice podobných výsledků. Tyto položky byly spojeny do podoby: „anglická nahrávka německá nahrávka anglická nahrávka“ a „německá nahrávka anglická nahrávka německá nahrávka“. V jedné výsledné nahrávce jsou tedy vždy právě dva řečníci. Stejným způsobem se spojily i přepisy daných nahrávek. Tímto způsobem skript pokračoval dokud v nově vytvořeném datasetu nebyl požadovaný počeet položek, tedy 4000. Pokaždé když se vyčerpaly všechny nahrávky od dané dvojice, byli oba řečníci odstaněni z datového rámce, tak aby nemohlo dojít k jejich opakovanému přidání do stejného datasetu, nebo se nemohli objevit v datasetu trénovacím, pokud již byly jejich nahrávky v testovací sadě.

Všechny části těchto spojených nahrávek (tedy vždy první anglická, německá i druhá anglická, a obráceně první německá, anglická a druhá německá) se při tvorbě datasetů podobných mluvčích ukládaly. Z těchto uložených nahrávek poté vznikly datasety, jejich položky byly vytvořeny náhodným spojením nahrávek, bez ohledu na mluvčího. Výsledné nahrávky obsahují 4000 nahrávek, které jsou dlouhé průměrně 19 sekund. Celková doba audionahrávek jednoho kombinovaného trénovacího datasetu je tedy přibližně 21 hodin.

5.4 Skript pro fine tuning

Pro fine tuning modelu byl využitý skript z tutoriálu Hugging Face [6] jen s malými úpravami. Funguje tak, že načte původní předtrénovaný model, v případě této práce tedy Whisper na checkpointu medium. Dále načte trénovací a testovací dataset. Trénovací slouží k samotnému trénování, testovací slouží k průběžnému vyhodnocování, jak výsledné úspěšnosti - WER, tak dalších důležitých metrik, například ztrátové funkce (training loss). Dále se načte feature extractor, tokenizer a whisper processor a data se převzorkují na požadovanou vzorkovací frekvenci 16 000. Položky v datasetech se rozdělí do dávek (batch) a definuje se hodnotící metrika (WER). Nakonfiguruje se jazyk učení a hyperparametry. Nakonec proběhne samotný trénink a uložení výsledného modelu na Huggingface hub.

Kapitola 6

Výsledky práce

Každý natrénovaný model byl testován na všech testovacích datasetech, tedy na anglickém (EN), německém (DE), německo-anglickém s podobnými mluvčími(EN-DE-EN-podob), anglicko-německém s podobnými mluvčími (DE-EN-DE-podob), německo-anglickém s různými řečníky (EN-DE-EN-ruzni), anglicko-německém s různými řečníky (DE-EN-DE-ruzni) a namluveném dataset (vlastni). Každé testování proběhlo dvakrát, jednou když byl model při vyhodnocování nastaven na rozpoznávání angličtiny a jednou němčiny.

6.1 Původní stav

Model Whisper v konfiguraci medium dosáhl na testovacích datech očekavatelných výsledků, Při nastavení na stejný jazyk, jako ten na kterém byl testován dosahoval poměrně malé chybovosti - do 10 WER, v případě kombinovaných datasetů, vždy poměrně spolehlivě rozpoznal tu třetinu, nebo dvě třetiny slov, na jejichž jazyk byl nastaven. Na vlastním namluveném datasetu dosahovala chybovost přes 40 WER, konkrétní hodnoty lze vidět v tabulce 6.1:

Tesovací data	WER EN	WER DE
DE	99.54	8.57
EN	9.40	85.99
EN-DE-EN-podob	35.32	63.82
DE-EN-DE-podob	73.10	38.64
EN-DE-EN-ruzni	38.11	62.61
DE-EN-DE-ruzni	74.43	37.38
vlastni	46.30	41.86

Tabulka 6.1: Tabulka výsledků netrénovaného modelu whisper medium při nastavení rozpoznávání na angličtinu a němčinu

6.2 Adaptace na jednojazyčných datech

Výchozím bodem byla adaptace základního Whisperu na čistě anglických a čistě německých datech. Model byl adaptován na každém jazyce dvakrát, jednou při nastavení učení na němčinu a jednou na angličtinu.

6.2.1 Modely adaptované na jednojazyčných datasetech stejného jazyka

Hypotézou bylo, že při trénování modelu v nastavení na jazyk A, na datasetu v jazyce A dojde jen k nepatrným změnám při testování jak německé tak anglické konfigurace - Na výraznější vylepšení výsledků v jazyce A jsou trénovací datasety příliš malé a na vylepšení výsledků v jazyce B, by byly potřeba i nějaká data v jazyce B. Testování tuto hypotézu potvrdilo a oba modely dosahovaly podobných výsledků, jako původní neadaptovaný model, výsledky můžeme vidět v tabulce 6.2.

Tesovací data	WER EN	WER DE
DE	99.54	9.61
EN	9.40	67.53
EN-DE-EN-podob	35.32	33.53
DE-EN-DE-podob	73.10	38.64
EN-DE-EN-ruzni	38.11	62.61
DE-EN-DE-ruzni	74.43	37.38
vlastni	46.30	41.86

Tabulka 6.2: Tabulka výsledků modelu Whisper medium pří trénování a rozpoznání nastaveném na stejný jazyk

6.2.2 Modely adaptované na jednojazyčných datasetech druhého jazyka

V případě trénování při učení nastevenám ja jazyk A a dataset v jazyce B byla hypotéza, že bude výrazné zlepšení na výsledcích na řeči B, a mírné zlepšení na výsledcích kombinované řeči. Výsledeky hypotéze větčinou odpovídaly, v některých případech byly dokonce výrazně lepší než očekávání. Například u modelu adaptovaného na německých datech, který byl při tréninku nastaven na angličtinu, bylo při testování nastaveném na angličtinu dosaženo velmi dobrých výsledků. Jak na kombinovaných, tak na namluveném datasetu byla chybovost WER jen kolem 15. Konkrétní výsledky můžeme vidět v tabulkách 6.3 a 6.4.

Tesovací data	WER
DE	8.82
EN	11.43
EN-DE-EN-podob	13.37
DE-EN-DE-podob	16.83
EN-DE-EN-ruzni	13.97
DE-EN-DE-ruzni	17.46
vlastni	14.72

Tabulka 6.3: Tabulka výsledků modelu Whisper medium adaptovaného na německých datech při učení nastaveném na angličtinu

V případě modelu adaptovaném na anglických datech, který byl při tréninku nastaven na němčinu, byly při testování nastaveném na němčinu (tedy zrcadlový případ předchozího modelu) výsledky o poznání horší. Za to ale byly více odpovídající původní hypotéze.

Tesovací data	WER
DE	20.06
EN	8.67
EN-DE-EN-podob	19.97
DE-EN-DE-podob	18.84
EN-DE-EN-ruzni	22.40
DE-EN-DE-ruzni	17.74
vlastni	30.34

Tabulka 6.4: Tabulka výsledků modelu Whisper medium adaptovaného na anglických datech při učení nastaveném na němčinu

6.3 Adaptace na dvojjazyčných datech

Při tréninku na vícejazyčných datech byly očekávány výsledky s velmi nízkým WER na všech vícejazyčných datasetech. Pozornost byla soustředěna zejména na vliv podobnosti mluvčího u trénovacích dat.

6.3.1 Trénování na datasetech EN-DE-EN podobných mluvčích

Po trénování byly očekávány velmi dobré výsledky testů na kombinovaných datasetech, hlavně toho poskládaného stejným způsobem, jako dataset trénovací. Výsledky na datech Common Voice hypotéze odpovídaly, výsledky testování na vlastních datech dopadly o jednotky WER hůře. Výsledky můžeme vidět v tabulkách 6.5 a 6.6.

Tesovací data	WER EN	WER DE
DE	52.79	9.20
EN	8.81	9.33
EN-DE-EN-podob	11.74	7.41
DE-EN-DE-podob	46.11	8.79
EN-DE-EN-ruzni	12.92	8.36
DE-EN-DE-ruzni	48.25	9.56
vlastni	22.20	17.52

Tabulka 6.5: Tabulka výsledků modelu Whisper medium adaptovaného na datech typu EN-DE-EN s podobnými řečníky při učení nastaveném na němčinu. WER EN a WER DE značí výsledky při testování nastaveném na odpovídající jazyk

Tesovací data	WER EN	WER DE
DE	10.3	9.70
EN	8.94	16.84
EN-DE-EN-podob	7.37	7.87
DE-EN-DE-podob	8.64	8.99
EN-DE-EN-ruzni	8.35	9.25
DE-EN-DE-ruzni	10.62	9.55
vlastni	17.19	18.26

Tabulka 6.6: Tabulka výsledků modelu Whisper medium adaptovaného na datech typu EN-DE-EN s podobnými řečníky při učení nastaveném na angličtinu. WER EN a WER DE značí výsledky při testování nastaveném na odpovídající jazyk

Na těchto výsledcích lze pozorovat výrazné zmenšení závislosti chybovosti na nastavení jazyka při testování.

6.3.2 Trénování na datasetech EN-DE-EN různých mluvčích

Po trénování na datech různých mluvčích byly očekávany podobné výsledky jako u podobných mluvčích, s tím, že měly být o jednotky WER horší v případě vyhodnocování na datasetech podobných mluvčích. Ve výsledku jsou ale rozdíly tréninku na podobných a náhodných mluvčích minimální jak můžeme vidět v tabulkách 6.7 a 6.8.

Tesovací data	WER EN	WER DE
DE	92.29	9.97
EN	8.34	9.08
EN-DE-EN-podob	22.12	7.6
DE-EN-DE-podob	67.07	9.07
EN-DE-EN-ruzni	26.27	7.96
DE-EN-DE-ruzni	67.87	9.76
vlastni	31.0	17.35

Tabulka 6.7: Tabulka výsledků modelu Whisper medium adaptovaného na datech typu EN-DE-EN s různými řečníky při učení nastaveném na němčinu. WER EN a WER DE značí výsledky při testování nastaveném na odpovídající jazyk

Tesovací data	WER EN	WER DE
DE	13.07	9.38
EN	8.73	11.41
EN-DE-EN-podob	7.57	7.82
DE-EN-DE-podob	9.45	8.96
EN-DE-EN-ruzni	8.24	8.72
DE-EN-DE-ruzni	10.49	9.68
vlastni	18.5	17.68

Tabulka 6.8: Tabulka výsledků modelu Whisper medium adaptovaného na datech typu EN-DE-EN s různými řečníky při učení nastaveném na angličtinu. WER EN a WER DE značí výsledky při testování nastaveném na odpovídající jazyk

Modely dosahující nejlepších výsledků byly trénovány opakovaně, aby bylo ověřeno, že dosažené výsledku nejsou jen nějaké výkyvy, ale lze jich dosáhnout opakovaně. V případě modelů EN-DE-EN-podob i EN-DE-EN-ruzni byly výkyvy minimální. Trénování tedy funguje spolehlivě. Rozdíly mezi jednotlivými modely jsou však tak malé, že nelze pozorovat vliv podobnosti mluvčího v tréninkových datasetech na výslednou úspěšnost modelu.

6.3.3 Výsledky rozpoznávání jednotlivých promluv

Metrika WER je vhodná pro objektivní a vyčíslitelné hodnocení úspěšnosti modelu, je ale také důležité vědět jakým způsobem model chybuje. Proto byly na modelech s nejlepšími výsledky testovány jednotlivé položky a byly sledovány chyby jakých se model dopouští. V následující části se budou vyskytovat ukázky výsledných přepisů. Správný přepis z datasetu bude označen jako `S` (Skutečný přepis), Výsledek modelu trénovaném na datasetu EN-DE-EN podobných řečníků, nastavený při tréninku na angličtinu bude označen jako `P` (Podobní řečníci) a stejný dataset, jen s náhodnými mluvčími bude označen jako `R` (Různí řečníci).

Model trénovaný na datech EN-DE-EN-podob i model trénovaný na EN-DE-EN-ruzni většinou chybovali ve stejných případech, kdy se vevětšině případů jednalo o chybu, kterou nezpůsobil code-switching, ale špatné porozumění slovu v jednom jazyce, případně chybějící koncovka. Například:

```
S: Bloomingdale's picked up the brand
P: Bloomingdale picked up the brand
R: Bloomingdale picked up the brand
```

Sice se jedná o chybu, ale nesouvisí s code switchingem a jejím řešením by byl větší trénink na konkrétní jeden jazyk.

Dalším druhem chyby, kde už je na vině code-switching je přepis pojmu, který se vyslovuje v obou jazycích prakticky stejně, ale píše se jinak například Afrika.

```
S: highest mountain in Afrika
P: highest mountain in Africa
R: highest mountain in Africa
```

Tato chyba se sice projeví v hodnotě WER, ale na význam věty nemá vůbec žádný vliv. Není tedy nutné se na zamezení vzniku takových chyb soustředit.

V jednom z mála případů, kdy chyba nebyla stejná u obou testovaných modelů jsou výsledky následující:

```
S: Die Mozart's compositions are celebrated for their beauty und complexity
P: The Mozarts' compositions are celebrated for their beauty and complexity
R: D. Mozart's compositions are celebrated for their beauty and complexity
```

V tomto případě se liší první slovo, V případě modelu trénovaném na podobných řečnících se člen přepsal ve špatném jazyce, jednoznačná chyba při code-switchingu, které by šlo zamezit laděním na větších datech, výslovnost slov „Die“ a „The“ je dostatečně rozdílná na to aby ji řádně natrénovaný model mohl rozeznat. Totéž platí u poslední spojky.

V trénovacích datech se v podstatě nevyskytoval případ, kde by bylo do jiného jazyka změněno jen jedno krátké slovo. Pro lepší zvládání těchto případů by bylo potřeba trénovat code-switching na úrovni slov.

Kapitola 7

Závěr

Cílem této práce bylo adaptovat model Whisper na rozpoznávání řeči obsahující code-switching na různých datasetech a při různých nastaveních trénovaného jazyka. Na začátku proběhlo seznámení se s problematikou strojového učení a automatického rozpoznávání řeči. Dále proběhlo seznámení se s modelem Whisper a daty z datasetů Mozilla Common Voice.

Byly vytvořeny datasety pro trénování a testování modelů. Tyto datasety obsahovaly jak data pouze anglická a pouze německá, tak jejich kombinace. Datasety obsahující dvojjazyčné nahrávky byly ještě rozděleny na datasety obsahující nahrávky sestavené z řeči podobných mluvčích a datasety obsahující nahrávku sestavené z nahrávek náhodných mluvčích.

Kromě těchto datasetů, byl vytvořen ještě jeden testovací dataset z vlastní namluvené řeči. Ten sloužil k ověření úspěšnosti na promluvách skutečné vícejazyčné řeči jednoho řečníka.

Whisper byl nejdříve otestován na všech testovacích sadách. Poté byl trénován na jednojazyčných datasetech. Modely adaptované na jednom jazyce, dosahovaly poměrně slušných výsledků. WER na kombinovaných datech se pohyboval mezi 13.37 s 17.46 na rozdíl od hodnot blížících se 70 u původního neadaptovaného modelu.

Dále proběhlo trénování na vícejazyčných datasetech. Při testování takto trénovaných modelů se výsledky testování na složených nahrávkách z datasetů Common Voice držely na hodnotách menších než 10. To už jsou poměrně slušné výsledky. Testování na vlastním namluveném datasetu dopadlo o poznání hůře, WER se pohyboval kolem 17.

Při testování modelů trénovaných na kombinovaných datasetech nebyl na výsledcích žádný patrný vliv podobnosti mluvčích. Hypotézou bylo, že modely trénované na nahrávkách podobných řečníků budou dosahovat lepších výsledků při testování na datech s podobnými řečníky, či úplně stejným řečníkem. Tato hypotéza se nepotvrdila, rozdíly mezi takto trénovanými modely byly v rámci statistické chyby.

Navázáním na tuto práci by mohlo být zaměření se na vytváření dat, kde ke code-switchingu dochází na úrovni slov. To by pomohlo nejen k tréninku lepších modelů, ale také k možnosti vytvářet datasety, kde na sebe různě jazyky navazují i významově. V této práci byla snaha taková data vytvořit. Zvolený přístup vytváření časových značek model Whisper však nebyl vhodný a vytvořená data obsahovala chyby, které by znehodnotily výsledky.

Literatura

[1] Aliferis, C. a Simon, G. *Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI.* Cham: Springer International Publishing, 2024. 477–524 s. ISBN 978-3-031-39355-6. Dostupné z: https://doi.org/10.1007/978-3-031-39355-6_10.

[2] Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M. et al. Common Voice: A Massively-Multilingual Speech Corpus. In: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020).* 2020, s. 4211–4215.

[3] Clevert, D.-A., Unterthiner, T. a Hochreiter, S. *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs).* 2016.

[4] Coria, J. M., Bredin, H., Ghannay, S. a Rosset, S. A Comparison of Metric Learning Loss Functions for End-To-End Speaker Verification. In: Espinosa Anke, L., Martín Vide, C. a Spasić, I., ed. *Statistical Language and Speech Processing.* Springer International Publishing, 2020, s. 137–148. ISBN 978-3-030-59430-5.

[5] Dong Yu, L. D. *Automatic Speech Recognition.* Springer London, 2014. ISBN 978-1-4471-5779-3.

[6] Gandhi, S. *Fine-Tune Whisper For Multilingual ASR with HuggingFace Transformers* [online]. [cit. 2024-3-1]. Dostupné z: https://huggingface.co/blog/fine-tune-whisper.

[7] Goodfellow, I., Bengio, Y. a Courville, A. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[8] Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X. et al. Pre-trained models: Past, present and future. *AI Open.* 2021, sv. 2, s. 225–250. DOI: https://doi.org/10.1016/j.aiopen.2021.08.002. ISSN 2666-6510. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2666651021000231.

[9] Hendrycks, D. a Gimpel, K. *Gaussian Error Linear Units (GELUs).* 2023.

[10] Honnibal, M. a Montani, I. *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.* 2017.

[11] Klakow, D. a Peters, J. Testing the correlation of word error rate and perplexity. *Speech Communication.* 2002, sv. 38, č. 1, s. 19–28. DOI: https://doi.org/10.1016/S0167-6393(01)00041-3. ISSN 0167-6393. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0167639301000413.

[12] Lowerre, B. T. *The Harpy speech recognition system.* 1976. Disertační práce. Carnegie Mellon University, Pennsylvania.

[13] Mao, A., Mohri, M. a Zhong, Y. *Cross-Entropy Loss Functions: Theoretical Analysis and Applications.* 2023.

[14] Mustafa, M. B., Yusoof, M. A., Khalaf, H. K., Rahman Mahmoud Abushariah, A. A., Kiah, M. L. M. et al. Code-Switching in Automatic Speech Recognition: The Issues and Future Directions. *Applied Sciences.* 2022, sv. 12, č. 19. Dostupné z: https://www.mdpi.com/2076-3417/12/19/9541.

[15] OpenAI. *Whisper: OpenAI's Automatic Speech Recognition System* [https://github.com/openai/whisper]. 2022.

[16] Rabiner, B. J. . L. R. *Automatic Speech Recognition – A Brief History of the Technology Development.* 2004. Dostupné z: https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf.

[17] Rabiner, L. R. *The Acoustics, Speech, and Signal Processing Society. A HistOrical Perspective.* 1984. Dostupné z: https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/216_historical%20perspective.pdf.

[18] Radford, A. *Robust Speech Recognition via Large-Scale Weak Supervision.* 2022. Dostupné z: https://cdn.openai.com/papers/whisper.pdf.

[19] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks.* 2015, sv. 61. DOI: https://doi.org/10.1016/j.neunet.2014.09.003. ISSN 0893-6080. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0893608014002135.

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. *Attention Is All You Need.* 2023.

[21] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C. et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing.* 2020.

[22] Zhou, R., Li, X., Fang, Y. a Li, X. *Mel-FullSubNet: Mel-Spectrogram Enhancement for Improving Both Speech Quality and ASR.* 2024.

[23] Zouhar, V., Meister, C., Gastaldi, J. L., Du, L., Vieira, T. et al. *A Formal Perspective on Byte-Pair Encoding.* 2023.